	Informe de análisis de vulnerabilidades, explotación y resultados del reto Robots.				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	27/11/2024	27/11/2024	1.0	MQ-HM-Robots	RESTRINGIDO

Informe de análisis de vulnerabilidades,

explotación y resultados del reto robots.

N.- MQ-Robots

Generado por:

GhoxPwn

Fecha de creación:

27.11.2024

Contenido

Tabla de Ilustraciones

***** SOLO PARA USO EDUCATIVO*****

Contenido de Tablas

***** SOLO PARA USO EDUCATIVO*****

1. Reconocimiento

Para iniciar el análisis de Pentesting, es necesario analizar las direcciones IP objetivo y los puertos abiertos de las máquinas a vulnerar. Estas acciones se realizarán a continuación:

Escaneo de dirección IP

Primero, debemos saber nuestra dirección IP, como se señala en la siguiente imagen:

Figura 1. Dirección IP de maquina Kali

Para la detección de la dirección IP de la maquina objetivo se hará un escaneo de arp dentro de la red como se muestra en la siguiente imagen:

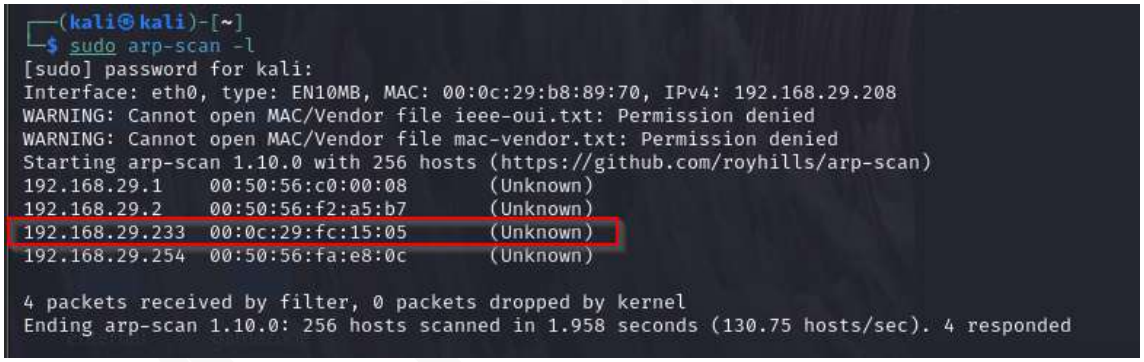


Figura 2. Dirección IP de la maquina Robots

Realizamos un ping a la dirección para saber su TTL, como se muestra a continuación:

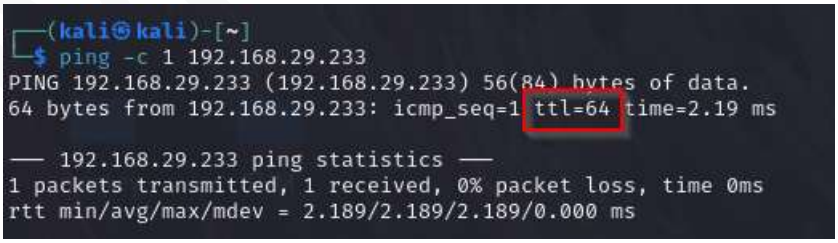


Figura 3. Testeo de paquetes máquina Robots

Tabla 1. Arquitectura de la maquina Robots

Arquitectura	Dirección
Linux	192.168.29.233

***** SOLO PARA USO EDUCATIVO*****

Como podemos ver, aún no sabemos qué máquina es, solo que tipo de arquitectura tiene. Más adelante, en el escaneo de puertos, podremos obtener mayor información de la máquina.

Escaneo de puertos

En esta fase, debemos escanear los puertos abiertos de la máquina descubierta en la Tabla 1. Para ello, utilizamos un escaneo de dos vías para las dos direcciones y todos sus puertos abiertos.

A continuación, se muestra los puertos abiertos de la máquina:

```
PORT    STATE SERVICE REASON
80/tcp  open  http    syn-ack ttl 64
443/tcp open  https   syn-ack ttl 64
MAC Address: 00:0C:29:FC:15:05 (VMware)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
           Raw packets sent: 3 (116B) | Rcvd: 3 (116B)
```

Figura 4. Escaneo de Puertos con Rustscan

Una vez detectados los puertos de la dirección, se realiza un análisis profundo de los puertos, como se muestra en la siguiente imagen:

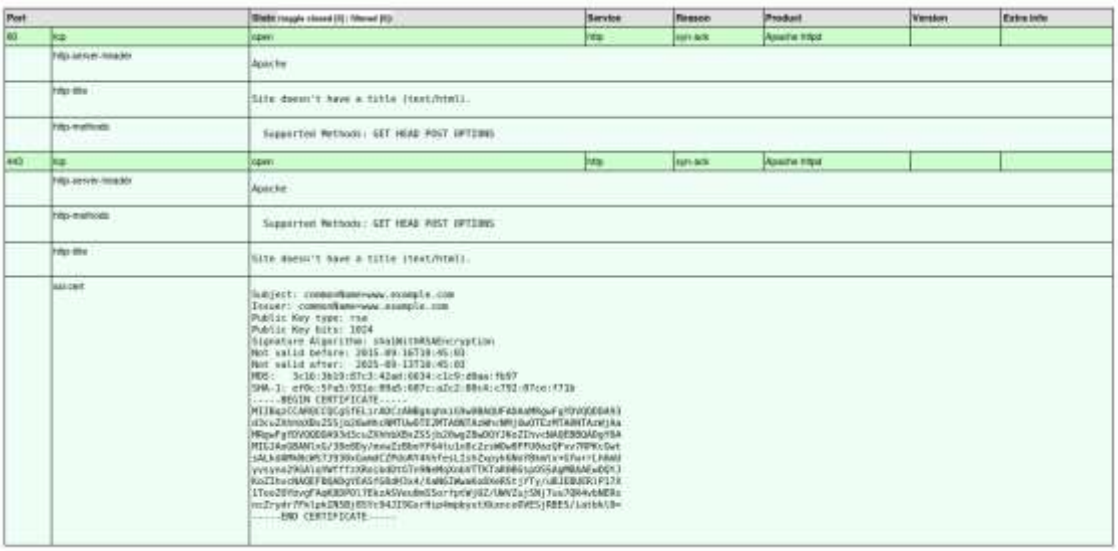


Figura 5. Escaneo de servicios y versiones

***** SOLO PARA USO EDUCATIVO*****

De la imagen, obtenemos las versiones de los puertos abiertos:

Tabla 2. Puertos abiertos de la maquina Robots

Puerto	Versión
80	Apache
443	Apache

2. Análisis de vulnerabilidades (Puertos abiertos)

Debido a que la máquina tiene puertos de servicio web, se hace una inspección del puerto 80 como primera prioridad.

Análisis de puerto HTTP (puerto 80)

Como primer paso, analizamos la portada de la página web y los servicios que tiene activados con la herramienta Wappalyzer:

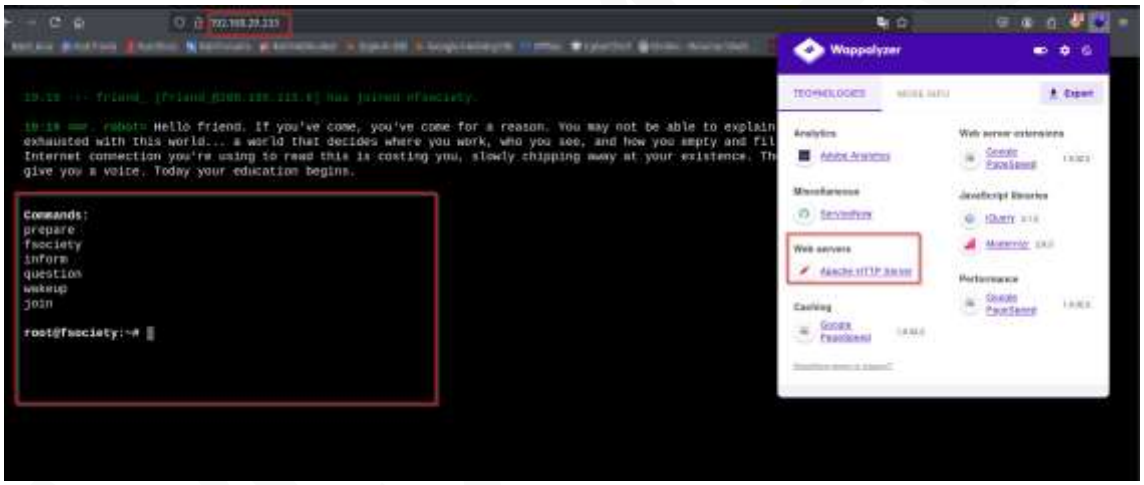


Figura 6. Evaluación inicial del HTTP (puerto 80)

Se examina si se tiene un directorio robots.txt, como se muestra a continuación:

```
User-agent: *
fsocity.dic
bandera1.txt
```

Figura 7. Contenido de directorio robots.txt

Del directorio robots podemos ver 2 direcciones interesantes: El primero es la bandera y el segundo es un diccionario, como se muestra a continuación:

***** SOLO PARA USO EDUCATIVO*****

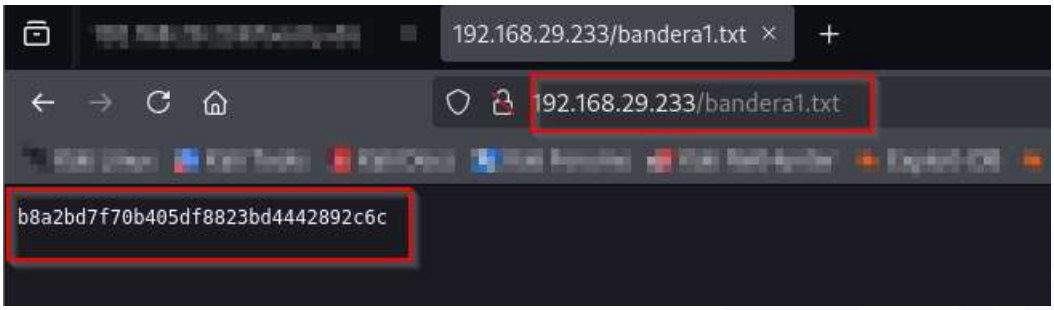


Figura 8. Contenido de la bandera1.txt

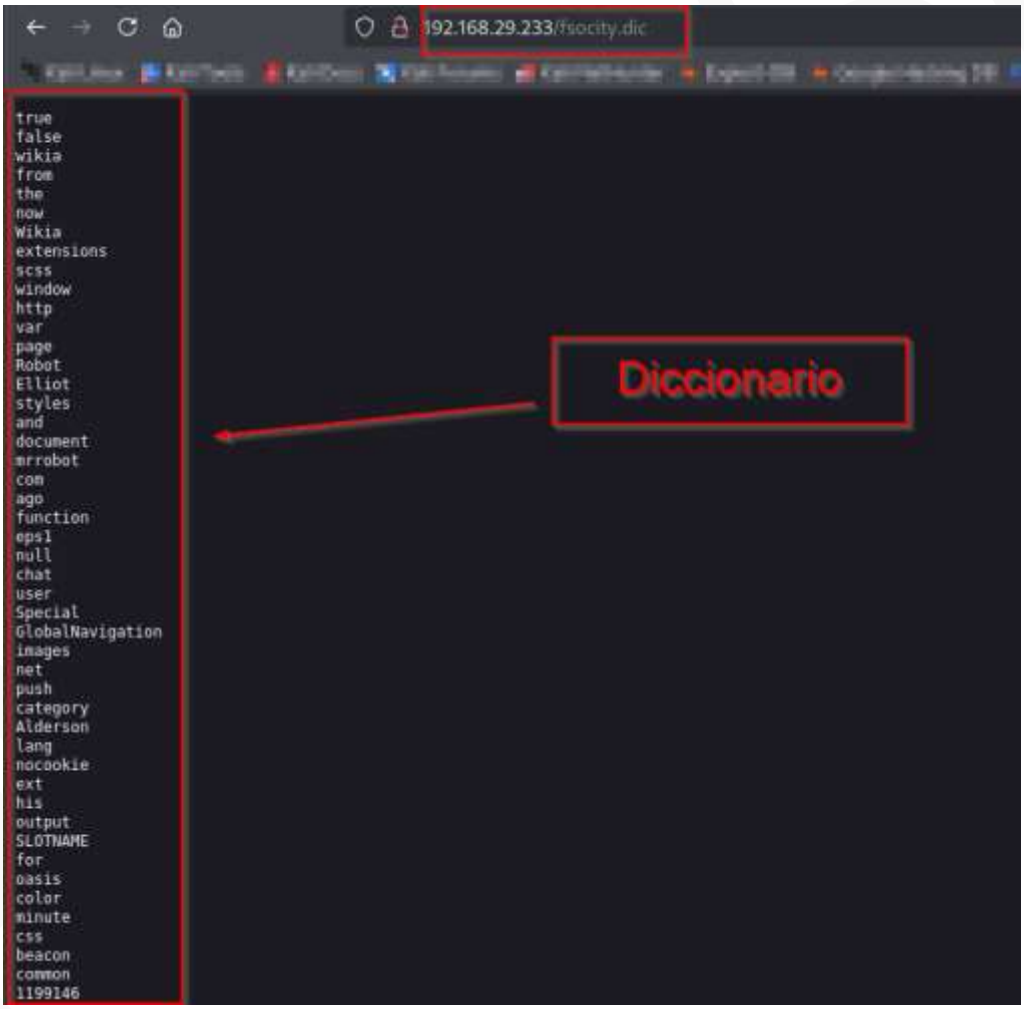


Figura 9. Contenido del directorio fsocity.dic

El contenido del último directorio se usará para hacer pruebas de credenciales por fuerza bruta.

***** SOLO PARA USO EDUCATIVO*****

Guardar directorio como diccionario

A continuación, se describen los pasos para usar como diccionario el directorio encontrado:

1. Se toma la dirección del diccionario y se guarda en nuestra maquina Kali.

Figura 10. Descarga del diccionario

2. Verificamos el contenido del diccionario. Como vemos que tiene muchas palabras, eliminamos las palabras repetidas.

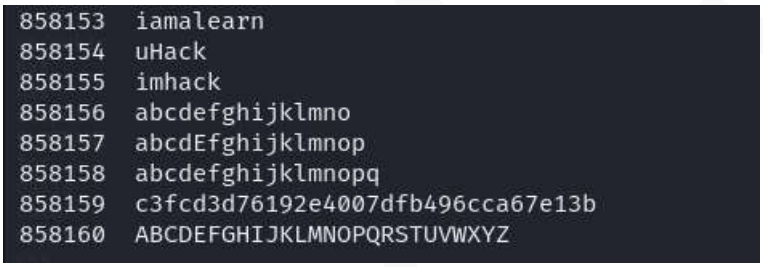


Figura 11. Cantidad de palabras del diccionario sin filtrar

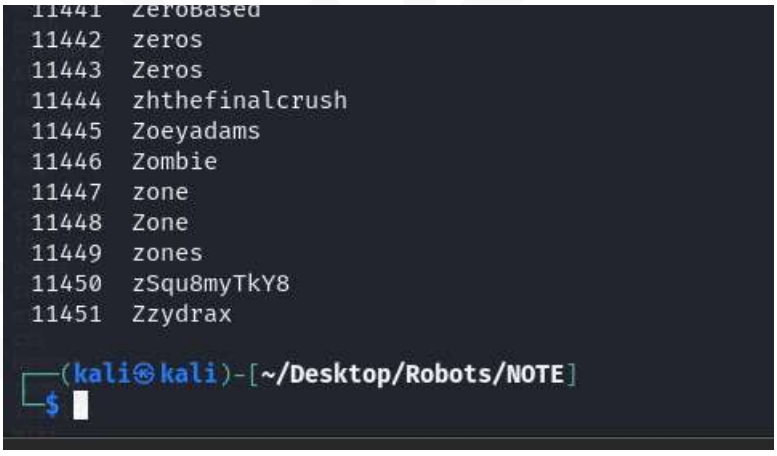


Figura 12. Cantidad de palabras del diccionario filtrados

El contenido del diccionario encontrado se guardó como diccionario.txt dentro de nuestra máquina. Debido a que no se encuentran más directorios dentro de robots.txt, se procederá a usar el método de fuzzing.

***** SOLO PARA USO EDUCATIVO*****

Fuzzing en dirección IP de la máquina

Realizamos una búsqueda de directorios por el método de fuzzing usando el comando gobuster.

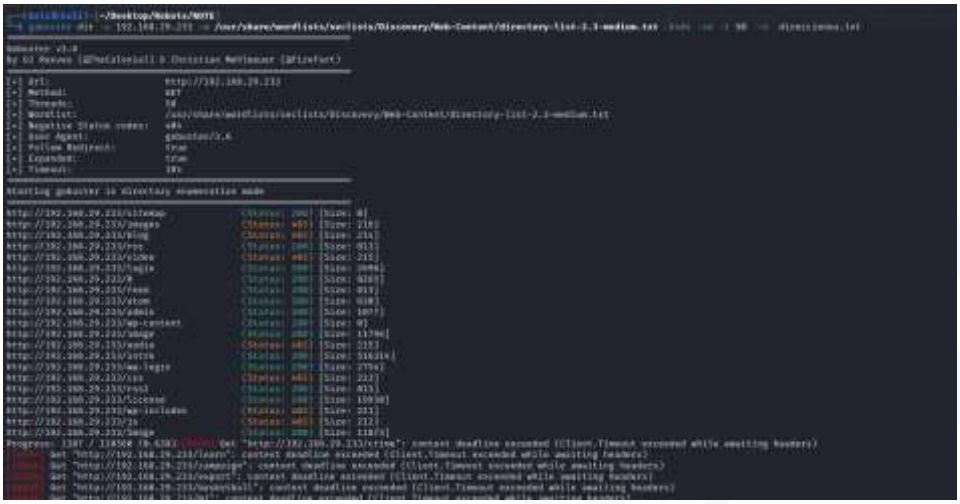
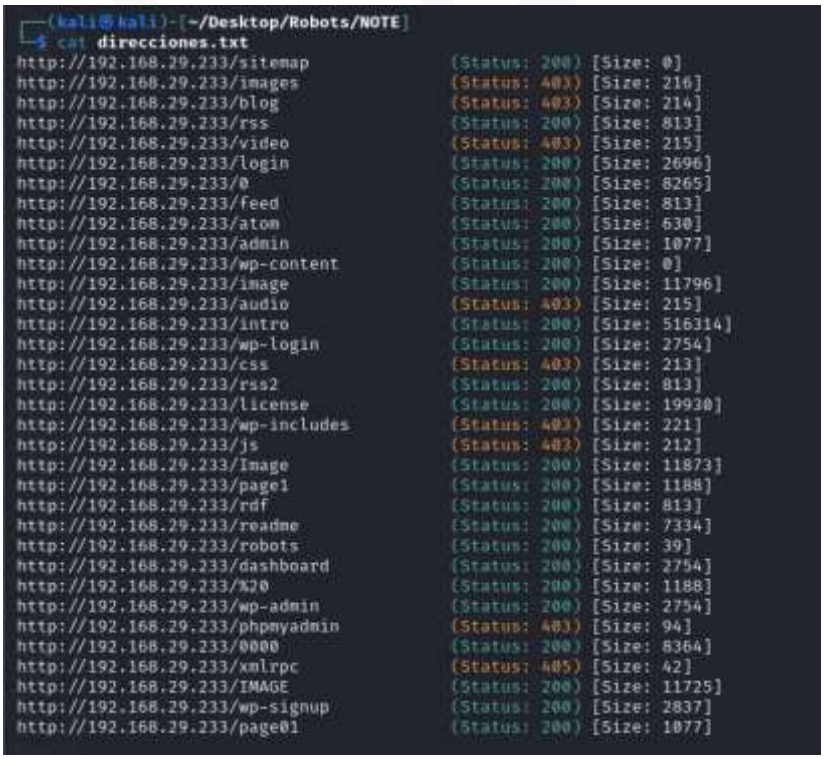


Figura 13. Fuzzing con la herramienta gobuster.

Se debe tener en cuenta que, por medidas de seguridad y para filtrar los errores que pueden surgir en las peticiones, se guardaron las direcciones encontradas en el archivo direcciones.txt.



```
┌──(kali㉿kali)-[~/Desktop/Robots/NOTE]
└─$ cat direcciones.txt
http://192.168.29.233/sitemap            (Status: 200) [Size: 0]
http://192.168.29.233/images             (Status: 403) [Size: 216]
http://192.168.29.233/blog               (Status: 403) [Size: 214]
http://192.168.29.233/rss                (Status: 200) [Size: 813]
http://192.168.29.233/video              (Status: 403) [Size: 215]
http://192.168.29.233/login              (Status: 200) [Size: 2696]
http://192.168.29.233/0                  (Status: 200) [Size: 8265]
http://192.168.29.233/feed               (Status: 200) [Size: 813]
http://192.168.29.233/atom               (Status: 200) [Size: 630]
http://192.168.29.233/admin              (Status: 200) [Size: 1077]
http://192.168.29.233/wp-content         (Status: 200) [Size: 0]
http://192.168.29.233/image              (Status: 200) [Size: 11796]
http://192.168.29.233/audio              (Status: 403) [Size: 215]
http://192.168.29.233/intro              (Status: 200) [Size: 516314]
http://192.168.29.233/wp-login           (Status: 200) [Size: 2754]
http://192.168.29.233/css                (Status: 403) [Size: 213]
http://192.168.29.233/rss2               (Status: 200) [Size: 813]
http://192.168.29.233/license            (Status: 200) [Size: 19930]
http://192.168.29.233/wp-includes        (Status: 403) [Size: 221]
http://192.168.29.233/js                 (Status: 403) [Size: 212]
http://192.168.29.233/Image              (Status: 200) [Size: 11873]
http://192.168.29.233/page1              (Status: 200) [Size: 1188]
http://192.168.29.233/rdf                (Status: 200) [Size: 813]
http://192.168.29.233/readme             (Status: 200) [Size: 7334]
http://192.168.29.233/robots             (Status: 200) [Size: 39]
http://192.168.29.233/dashboard          (Status: 200) [Size: 2754]
http://192.168.29.233/%20                (Status: 200) [Size: 1188]
http://192.168.29.233/wp-admin           (Status: 200) [Size: 2754]
http://192.168.29.233/phpmyadmin         (Status: 403) [Size: 94]
http://192.168.29.233/0000               (Status: 200) [Size: 8364]
http://192.168.29.233/xmlrpc             (Status: 405) [Size: 42]
http://192.168.29.233/IMAGE              (Status: 200) [Size: 11725]
http://192.168.29.233/wp-signup          (Status: 200) [Size: 2837]
http://192.168.29.233/page01             (Status: 200) [Size: 1077]
```

Figura 14. Direcciones encontradas con gobuster

***** SOLO PARA USO EDUCATIVO*****

De las direcciones encontradas, las más interesantes son images, wp_login y wp_admin. Sin embargo, el único directorio en el que pude ingresar es wp_login, como se muestra a continuación:

Figura 15. Contenido del directorio images

Figura 16. Contenido de wp_admin

Se puede deducir que el contenido de wp_admin es una carpeta por lo cual se necesitaría otro fuzzing solo a esa carpeta

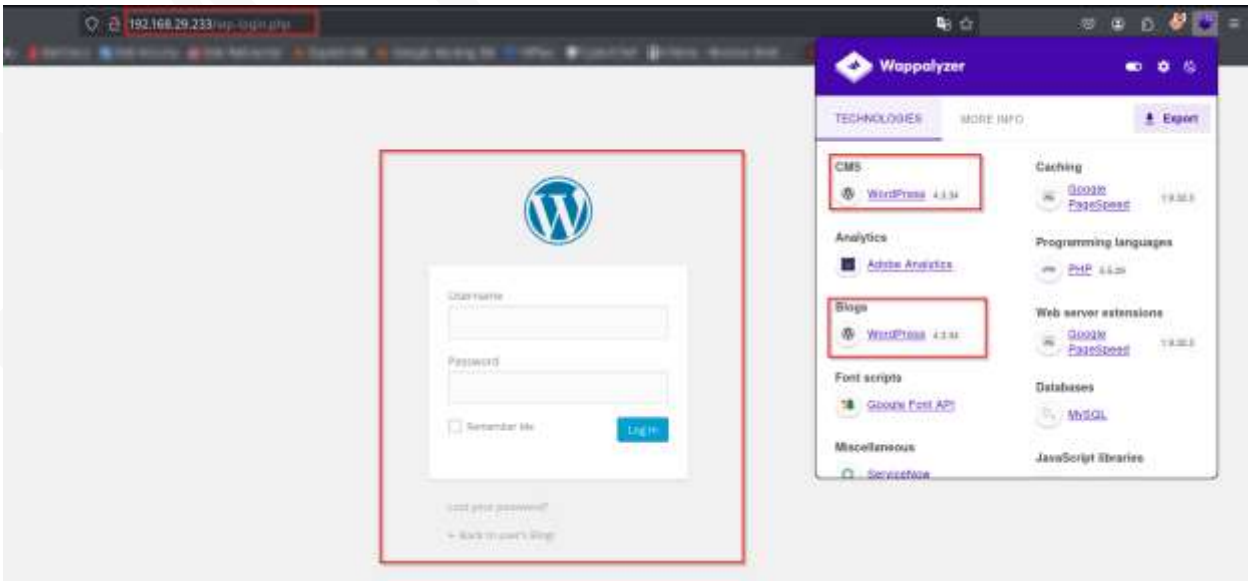


Figura 17. Contenido del directorio wp_login

***** SOLO PARA USO EDUCATIVO*****

Analizando si se encuentra un exploit solo encontramos de la versión 4.3.3 y no la requerida

Figura 18. Buscando exploit en wordpress

Figura 19. Verificando contenido del exploit encontrado versión diferente

Como podemos ver, el único posible exploit encontrado no cubre otras versiones posteriores, por lo cual se descarta este exploit.

3. Explotación de vulnerabilidades

En esta fase, se utilizará el método de fuerza bruta para obtener las credenciales necesarias y acceder al servicio web a través del directorio wp_login, utilizando las herramientas Hydra y Burp Suite. El primer paso será capturar la solicitud de inicio de sesión con Burp Suite para analizar el formato de login y los mensajes de error que devuelve el servidor, como se muestra en las siguientes imágenes:

Figura 20. Captura del POST (/wp-login.php)

***** SOLO PARA USO EDUCATIVO*****

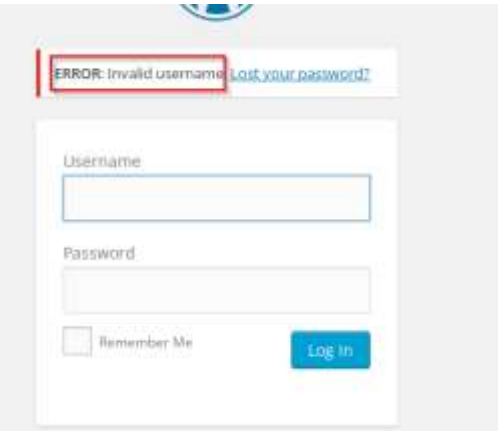


Figura 21. Log del error de wp-login

Con el formato de inicio de sesión y el log de error obtenidos, se utilizará Hydra para intentar descubrir el nombre de usuario de la siguiente manera:

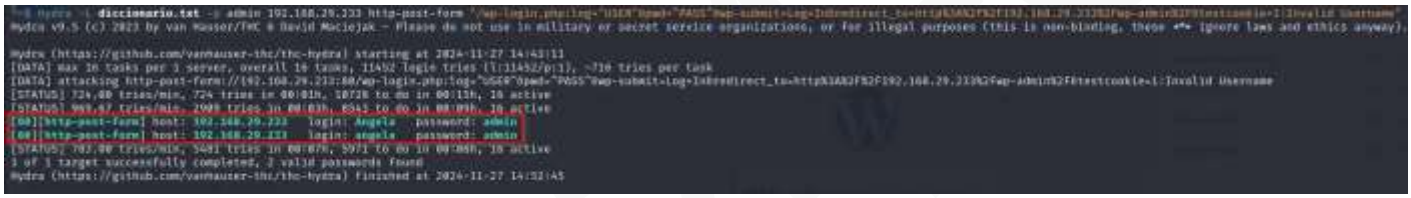


Figura 22. Detección de usuarios validos por fuerza bruta

Una vez identificado un nombre de usuario válido, se procederá a buscar el mensaje de error relacionado con contraseñas incorrectas, como se muestra a continuación:

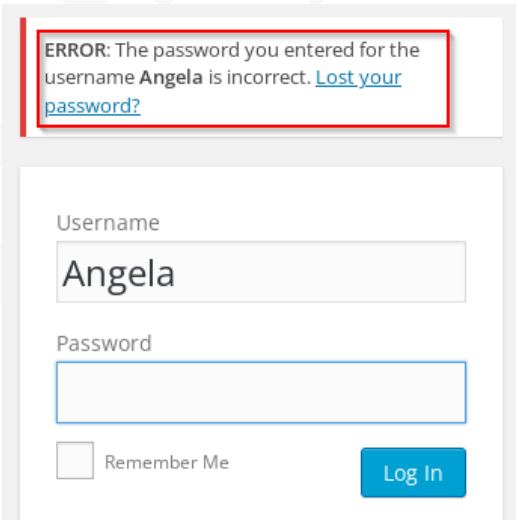


Figura 23. Log para error de contraseña

***** SOLO PARA USO EDUCATIVO*****

A continuación, se utilizará este nuevo mensaje de error para realizar un ataque de fuerza bruta con Hydra, de la siguiente forma:

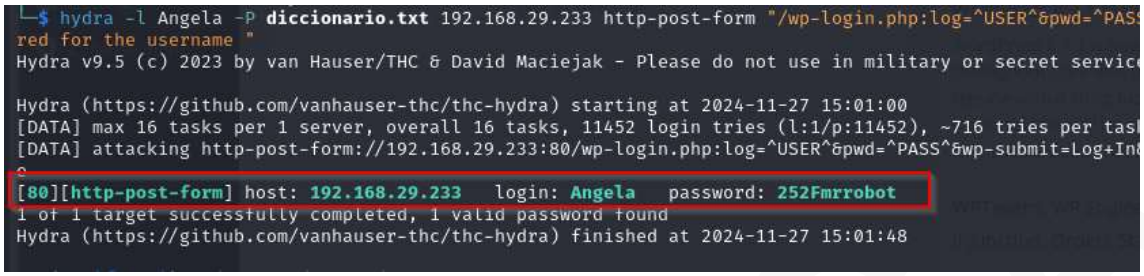


Figura 24. Obteniendo contraseña con HYDRA

Con las credenciales correctas obtenidas, se procederá a iniciar sesión en la página web, lo cual nos dará acceso a la siguiente información:

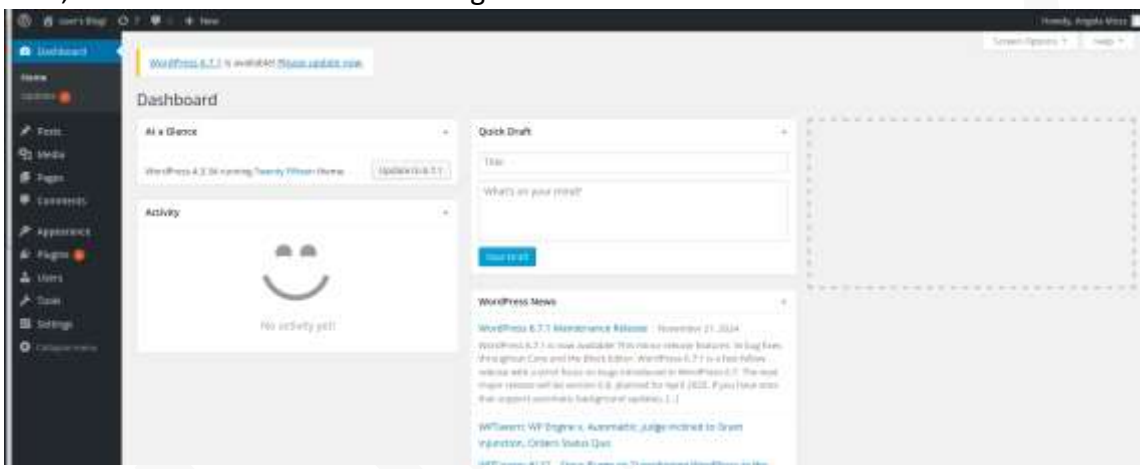


Figura 25. Ingreso con las credenciales

Al navegar por las opciones de la página, se obtiene el nombre de usuario y otra información adicional, como se muestra en la siguiente imagen:

Figura 26. Información encontrada

Reverse Shell

Para obtener acceso a la máquina, se buscarán vectores de ataque donde se pueda utilizar un reverse shell. Se identificó la siguiente vulnerabilidad:

Figura 27. Agregar plugins como reverse Shell

Se observó que es posible cargar archivos .zip dentro del directorio de plugins. Se creará un archivo PHP con la cabecera adecuada para WordPress y se comprimirá en formato .zip, como se muestra en las siguientes imágenes:

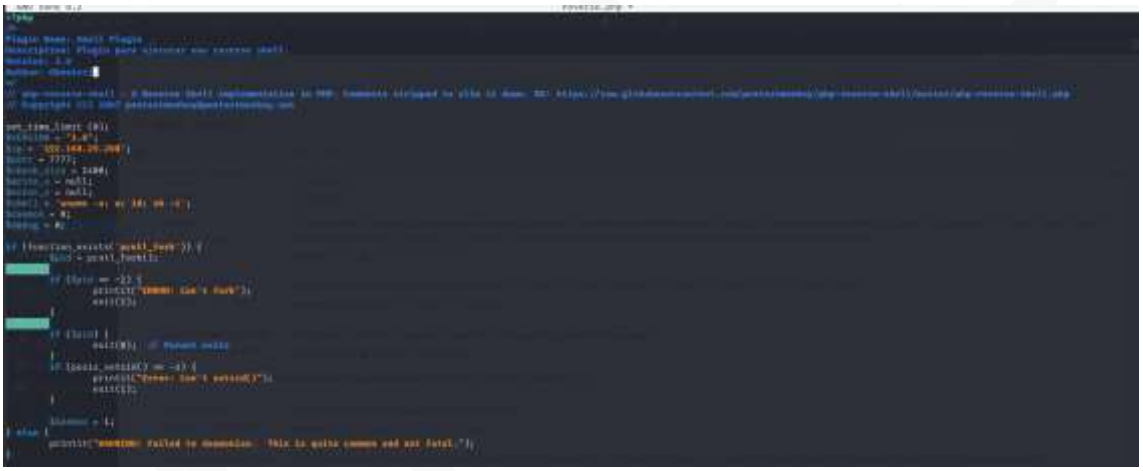


Figura 28. Reverse Shell en formato PHP

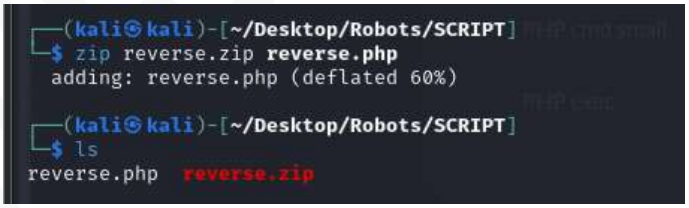


Figura 29. Comprimiendo el archivo

***** SOLO PARA USO EDUCATIVO*****

Si la cabecera del archivo PHP es correcta, el siguiente mensaje confirmará que el plugin ha sido activado, como se muestra a continuación:

Figura 30. Subiendo el plugin

Mientras tanto desde la maquina Kali se debió de estar en modo escucha con netcat en el puerto correspondiente:

Figura 31. Netcat modo escucha puerto 7777

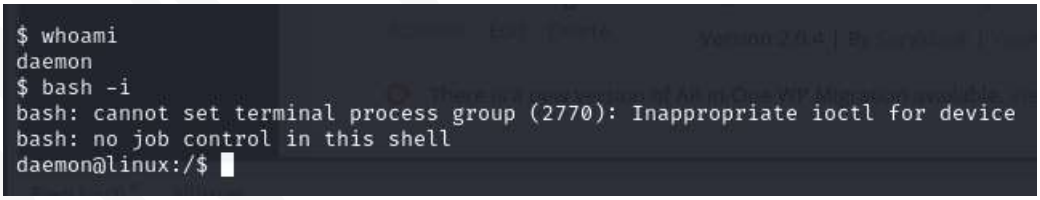


Figura 32. Verificamos que quienes somos

4. Escala de privilegios

En esta fase, el objetivo principal es buscar un usuario dentro de la máquina y explorar vulnerabilidades que permitan escalar privilegios a nivel de administrador. Para ello, utilizaremos la herramienta LinPEAS para analizar posibles vulnerabilidades relacionadas con la escalada de privilegios.

***** SOLO PARA USO EDUCATIVO*****

Obtención de credenciales

Realizamos una búsqueda global de documentos que puedan contener contraseñas dentro de la máquina y encontramos lo siguiente:

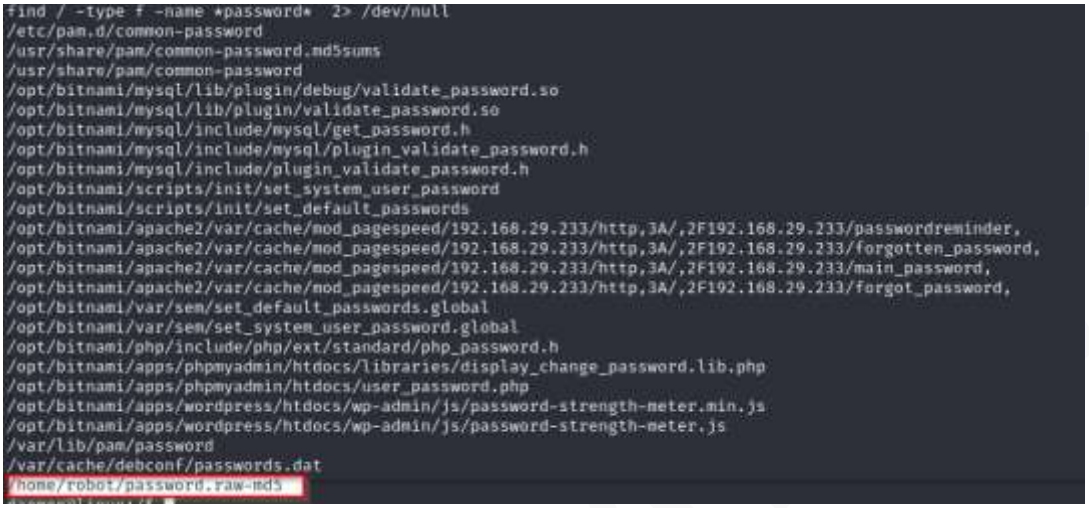


Figura 33. Buscando archivos con nombre password

Podemos deducir que probablemente se trate de un hash MD5 para el usuario "robot", como se observa en la dirección:

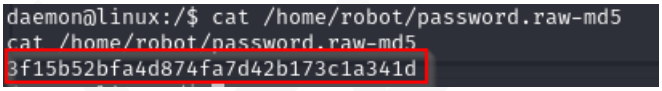


Figura 34. Hash del usuario robots

Verificamos que el hash es un MD5 utilizando un detector de hashes, como se muestra en la siguiente imagen:

Figura 35. Detección de formato de hash

***** SOLO PARA USO EDUCATIVO*****

A continuación, desciframos el hash utilizando la herramienta John the Ripper:

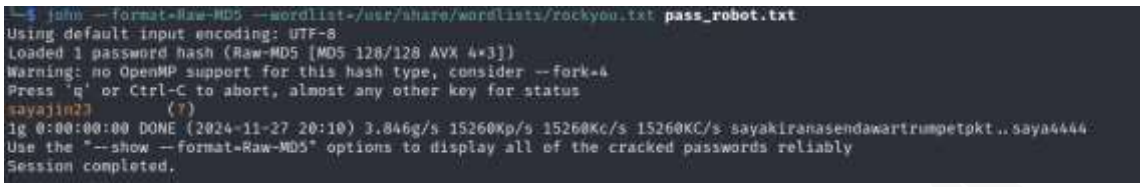


Figura 36. Usando JOHN para descifrar el hash

Una vez obtenemos la contraseña, iniciamos sesión con el usuario "robot", como se muestra a continuación:

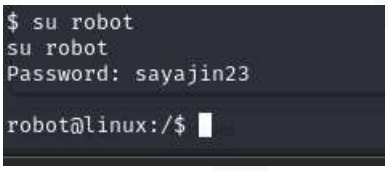


Figura 37. Logeo del usuario robot

Para buscar vulnerabilidades que nos permitan escalar privilegios, utilizamos LinPEAS. Los pasos para ello son los siguientes:

- Abrimos un servidor local para descargar el linpeas.sh

Figura 38. Server local kali

- Descargamos el archivo en el directorio shm de la maquina atacada

Figura 39. Descarga del linpeas

- Ejecutamos el linpeas.sh

***** SOLO PARA USO EDUCATIVO*****

Resultados de linpeas

El script LinPEAS nos proporciona una lista de vulnerabilidades y binarios que podrían permitirnos escalar privilegios.

Resultados de vulnerabilidades

A continuación, se muestran los resultados del análisis realizado por LinPEAS con respecto a las vulnerabilidades encontradas:

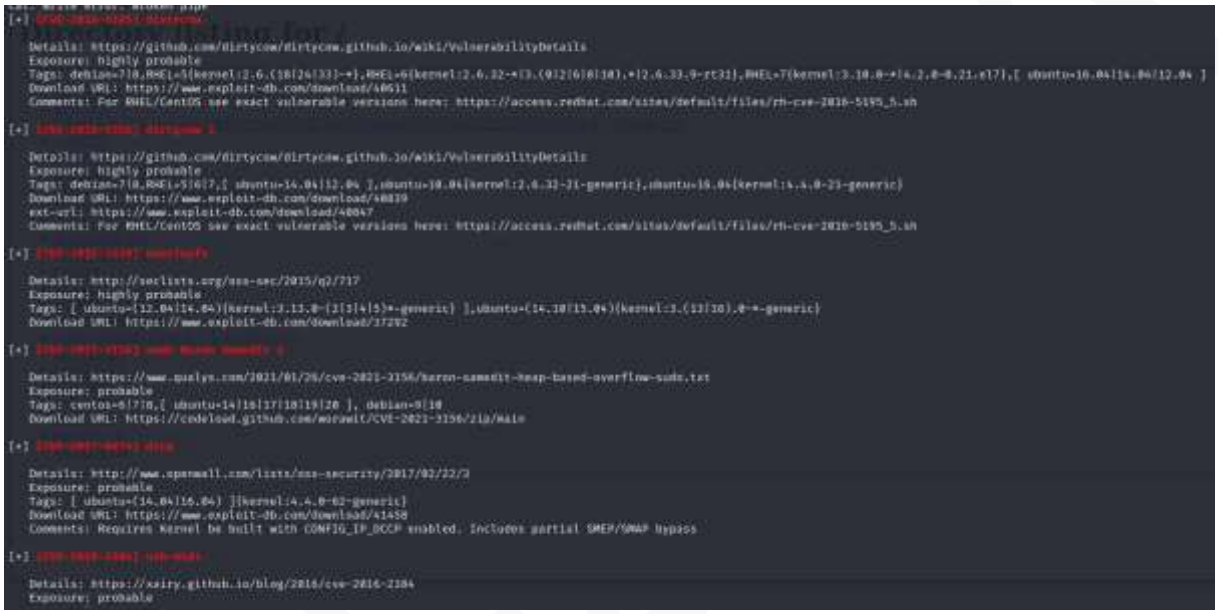


Figura 40. Resultados de CVE con linpeas

Asumiendo que contamos con un compilador, las siguientes vulnerabilidades podrían ser explotadas:

Posibles exploits que podrías usar:

- CVE-2016-5195 (dirtycow)
- CVE-2016-5195 (dirtycow 2)
- CVE-2021-3156 (sudo Baron Samedit 2)
- CVE-2015-1328 (overlayfs)
- CVE-2021-3156 (sudo Baron Samedit)
- CVE-2017-6074 (dccp)

Exploits que deberías descartar:

- CVE-2016-2384 (usb-midi)
- CVE-2015-8660 (overlayfs - ovl_setattr)
- CVE-2022-32250 (nft_object UAF)

***** SOLO PARA USO EDUCATIVO*****

N.- MQ-HM-Robots

- CVE-2019-18634 (sudo pwfeedback)
- CVE-2019-15666 (XFRM_UAF)
- CVE-2018-1000001 (RationalLove)
- CVE-2017-7308 (af_packet)
- CVE-2016-0728 (keyring)
- CVE-2014-5207 (fuse_suid)
- CVE-2017-1000253 (PIE_stack_corruption)

Resultados de SUID

De los resultados obtenidos con LinPEAS, encontramos los siguientes archivos SUID con privilegios de root, lo que nos podría permitir escalar privilegios:

Figura 41. Resultados de SUID con linpeas

Probaremos el SUID asociado a nmap, ya que LinPEAS lo marcó como una vulnerabilidad con alta probabilidad de éxito. El siguiente paso será verificar qué información podemos obtener para escalar privilegios con nmap:

Figura 42. Información para escalar privilegios con nmap

Para ello, necesitamos determinar la versión de nmap en la máquina. Esto se logra mediante el siguiente comando:

Figura 43. Versión del nmap

***** SOLO PARA USO EDUCATIVO*****

Vemos que la versión es vulnerable, por lo que procederemos a realizar la escalada de privilegios de la siguiente manera:

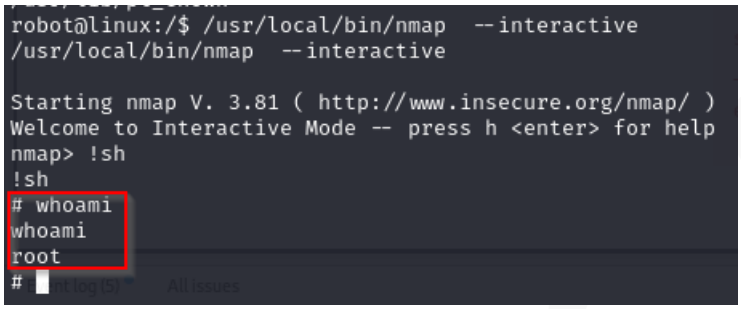


Figura 44. Escalando con NMAP

5. Banderas

Para este paso buscaremos los archivos llamados bandera1.txt, bandera2.txt y bandera3.txt. Se buscarán los archivos usando el comando find de la siguiente manera:

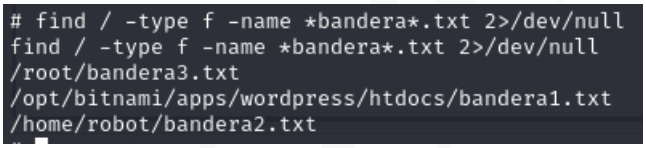


Figura 45. Buscando las banderas

Una vez tenemos las direcciones leemos el contenido de los archivos con cat

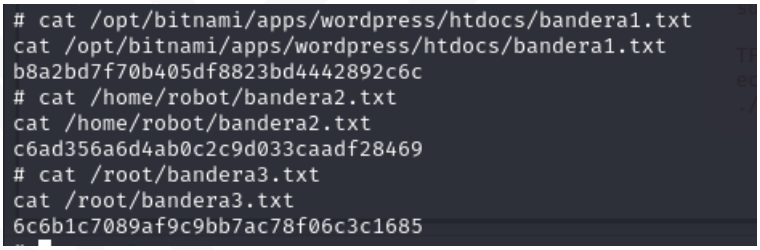


Figura 46. Contenido de las banderas

A continuación, pondrá en una tabla el contenido de los archivos requeridos

Tabla 3. Banderas maquina Robot

Bandera	Contenido
Bandera1.txt	b8a2bd7f70b405df8823bd4442892c6c
Bandera2.txt	c6ad356a6d4ab0c2c9d033caadf28469
Bandera3.txt	6c6b1c7089af9c9bb7ac78f06c3c1685

***** SOLO PARA USO EDUCATIVO*****

6. Conclusiones y Recomendaciones

- Revisión de binarios SUID: Realicé una revisión exhaustiva de todos los binarios SUID en el sistema. Se recomienda eliminar aquellos que no sean estrictamente necesarios para las operaciones diarias, ya que los binarios con privilegios elevados pueden ser explotados para escalar privilegios de manera maliciosa.

- Seguridad en rutas o directorios accesibles: Identifiqué posibles rutas o directorios accesibles a través de ataques de fuerza bruta. Como medida de seguridad, se recomienda limitar la cantidad de intentos de acceso por usuario y aplicar mecanismos de bloqueo temporal tras varios intentos fallidos, lo que dificultaría los intentos de acceso no autorizados.

- Protección de credenciales: Es fundamental evitar dejar credenciales almacenadas en archivos con permisos de lectura abiertos para cualquier usuario del sistema. Se debe garantizar que los archivos con información sensible tengan permisos estrictos para reducir el riesgo de exposición.

- Uso de algoritmos de hash robustos: Se debe evitar el uso de algoritmos de hash débiles como MD5 para las contraseñas. En su lugar, se recomienda emplear algoritmos de hash más seguros, como SHA-256 o bcrypt, que ofrecen mayor resistencia a ataques de colisión y de fuerza bruta.

***** SOLO PARA USO EDUCATIVO*****