

---

# Informe de Pentesting - Máquina

## Debian (10.0.2.11)

- Fecha del Informe: 3 de Abril de 2025
  - Fecha de las Pruebas: Aproximadamente 2-3 de Abril de 2025
  - IP Objetivo: 10.0.2.11
  - Sistema Operativo Identificado: Debian Linux
  - Realizado por: Rodrigo Martinez Cardozo
  - Alcance: Evaluación de seguridad de la máquina Debian en la IP 10.0.2.11, enfocándose en servicios expuestos (FTP, SSH, HTTP/WordPress, MySQL) y configuraciones del sistema, basado en la evidencia proporcionada.
- 

### Índice / Tabla de Contenidos

(Nota: Los números de página se omiten en este formato)

1. Resumen Ejecutivo
2. Metodología Detallada
3. Definición de Niveles de Riesgo
4. Hallazgos Detallados
  - 4.1. Vulnerabilidades Críticas
    - 4.1.1. V-01: Acceso Root Obtenido vía Fuerza Bruta SSH
    - 4.1.2. V-02: Permisos Inseguros en wp-config.php (777)
    - 4.1.3. V-03: Permisos Inseguros en Directorio Web y Archivos WP (777)
  - 4.2. Vulnerabilidades Altas
    - 4.2.1. V-04: Contraseña Débil y Reutilizada (root SSH y wordpressuser DB)
    - 4.2.2. V-05: WordPress Core Desactualizado (6.6.2)
    - 4.2.3. V-06: Claves Secretas y Sales Débiles/Por Defecto en WordPress
    - 4.2.4. V-07: Servidor FTP con Acceso Anónimo y Vulnerabilidades Conocidas
    - 4.2.5. V-08: Ausencia de Firewall de Host Activo
    - 4.3. Vulnerabilidades Medias
      - 4.3.1. V-09: Listado de Directorios Habilitado en Servidor Web
      - 4.3.2. V-10: Servicios Potencialmente Innecesarios Escuchando Localmente (CUPS)
      - 4.3.3. V-11: Versiones de Software (Servicios) Desactualizadas/Vulnerables (vsftpd, OpenSSH)
      - 4.4. Vulnerabilidades Bajas / Informativas
        - 4.4.1. V-12: XML-RPC Habilitado en WordPress
  5. Conclusiones y Próximos Pasos

---

## 1. Resumen Ejecutivo

La evaluación de seguridad realizada sobre el sistema Debian en la IP 10.0.2.11 ha revelado **múltiples vulnerabilidades críticas y altas**, culminando en la **obtención de acceso administrativo (root) al servidor** mediante un ataque de fuerza bruta contra el servicio SSH.

Los hallazgos más significativos incluyen:

- **Compromiso Total del Servidor:** Se obtuvo la contraseña del usuario root (123456) mediante fuerza bruta SSH, aprovechando la habilitación del login de root y la autenticación por contraseña.
- **Permisos de archivo extremadamente inseguros (777)** en el directorio web y archivos críticos como wp-config.php.
- **Credenciales débiles y reutilizadas:** La contraseña 123456 es utilizada tanto para root en SSH como para el usuario wordpressuser de la base de datos.
- **Configuraciones inseguras en SSH y FTP** (PermitRootLogin, PasswordAuthentication, Anonymous FTP enabled).
- **Versiones de software vulnerables** (OpenSSH 9.2p1, vsftpd 3.0.3, WordPress 6.6.2).
- **Ausencia total de un firewall de host activo.**
- Otros problemas como listado de directorios, claves secretas por defecto en WordPress y XML-RPC habilitado.

El sistema se encuentra en un estado de **riesgo crítico extremo**. Un atacante puede obtener control total del servidor con relativa facilidad. Se requiere la **remediación inmediata y urgente** de todas las vulnerabilidades críticas y altas.

## 2. Metodología Detallada

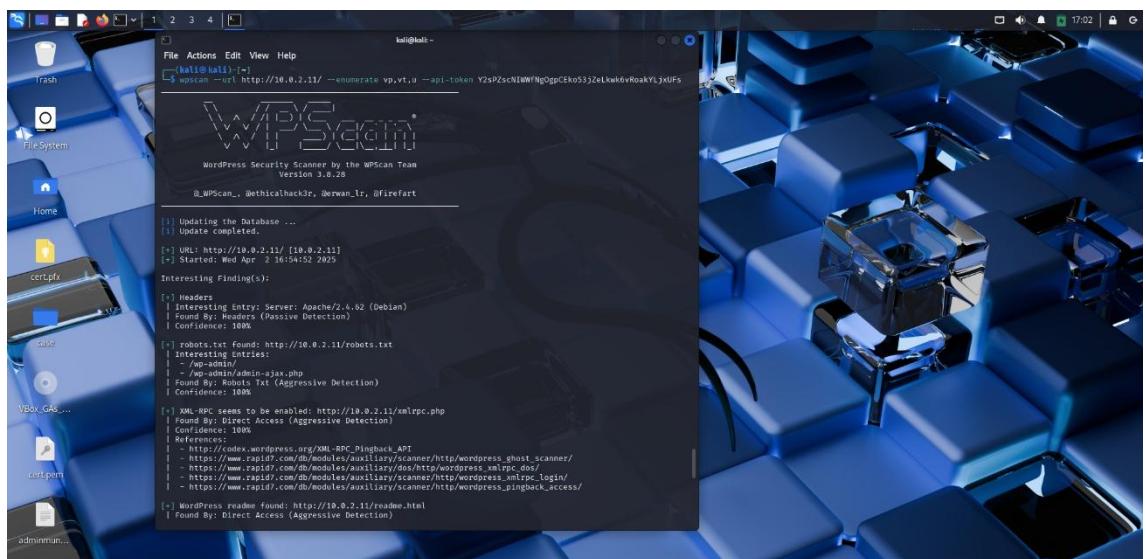
La evaluación se realizó combinando técnicas de escaneo remoto (Kali) y análisis/pruebas en el sistema objetivo (Debian), documentado en los archivos y salidas proporcionados:

- **Descubrimiento y Escaneo de Vulnerabilidades (Desde Kali):**

- `sudo nmap -sV --script=vuln 10.0.2.11`: Identificó puertos abiertos (21, 22, 80), servicios/versiones (vsftpd 3.0.3, OpenSSH 9.2p1, Apache 2.4.62), listó CVEs potenciales (vsftpd, OpenSSH) y enumeró información básica de WordPress.



- `wpscan --url http://10.0.2.11/ ...`: Identificó WordPress 6.6.2 (desactualizado), confirmó listado en /wp-content/uploads/, XML-RPC y WP-Cron habilitados. No encontró vulnerabilidades específicas en plugins/temas con este escaneo.



## **Pruebas Interactivas (Desde Kali):**

- **FTP Anónimo:** Se ejecutó ftp 10.0.2.11, se inició sesión como anonymous. Las pruebas (ls, cd, get) mostraron que el login fue exitoso pero el acceso estaba restringido a un directorio raíz aparentemente vacío y sin permisos de lectura/escritura.

```
File Actions Edit View Help
└$ mysql -h 10.0.2.11 -u wordpressuser -p
Enter password:
ERROR 2002 (HY000): Can't connect to server on '10.0.2.11' (115)
[ kali㉿kali: ~ ]
└$ ./vftpd-exploit.py 10.0.2.11
Connected to 10.0.2.11.
220 (vsFTPd 3.0.0)
Name (10.0.2.11:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote socket type is UNIX.
Using binary mode to transfer files.
Ftp> ls
221 Entering Extended Passive Mode (|||44673|)
15 Here comes the directory listing.
drwxr-xr-x 2 0 122 4096 Oct 08 16:09 ..
drwxr-xr-x 2 0 122 4096 Oct 08 16:09 ..
220 Directory send OK.
Ftp> pwd
Remote directory: /
ftp> cd /home
550 Failed to change directory.
ftp> cd www
550 Failed to change directory.
Ftp> cd /www
550 Failed to change directory.
Ftp> ls -la
229 Entering Extended Passive Mode (|||45902|)
330 Here comes the directory listing.
drwxr-xr-x 2 0 122 4096 Oct 08 16:09 ..
drwxr-xr-x 2 0 122 4096 Oct 08 16:09 ..
220 Directory successfully changed.
Ftp> rm -rf *
usage: cd remote-directory
Ftp> cd ..
220 Directory successfully changed.
Ftp> rm -rf *
usage: cd remote-directory
Ftp> exit
221 Goodbye.
[ kali㉿kali: ~ ]
└$ npmcn --url http://10.0.2.11/ --enumerate vp,vt,u --api-token Y2sPzscNTAwNgOgpCEko53jZeLkuKdvRoakYLxj0fFa
```

- **Búsqueda de Exploits CVE:** Se utilizó msfconsole para buscar módulos de exploit para las CVEs listadas por Nmap (search CVE-..., search vsftpd 3.0.3, search OpenSSH 9.2p1). **No se encontraron módulos de exploit directos** en Metasploit para las versiones/CVEs específicas con las búsquedas realizadas.
  - **Fuerza Bruta SSH:** Se utilizó hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://10.0.2.11. Esta prueba **tuvo éxito**, obteniendo la contraseña 123456 para el usuario root.

A screenshot of a Kali Linux desktop environment. The desktop background features a blue, metallic, geometric pattern. A terminal window titled 'kali@kali: ~' is open in the foreground, showing the output of a Hydra password cracking session. The command run was 'hydra -v -t 10 -l user -P xato-net-1-million-passwords.txt ssh://19.0.2.15'. The terminal shows progress, including attack statistics like '10 tasks per 3 servers, overall 16 tasks, 5189454 login tries (1:1/5189454), -32x41 tries per task [DATA] attacking ssh://19.0.2.15:22'. It also indicates that 1 target was successfully completed and 1 valid password was found. A warning message about writing restore files is shown. The session ends with a success message: 'Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-19 16:22:44'. The desktop interface includes a dock with icons for various tools like FileZilla, Firefox, and terminal, and a menu bar at the top.

- **Análisis de Configuración y Sistema (En Debian):**

- Se verificaron configuraciones (grep, cat) de vsftpd (anonymous\_enable=YES), sshd (PermitRootLogin yes, PasswordAuthentication yes), Apache (Options Indexes), WordPress (wp-config.php con DB pass '123456', sales por defecto; .htaccess).
  - Se verificaron permisos (ls -la) en /var/www/html y subdirectorios (encontrando 777).
  - Se accedió a MySQL (sudo mysql -u root) para listar usuarios (SELECT...) y bases de datos (SHOW DATABASES;).
  - Se verificó la red (netstat) y el estado del firewall (iptables fallido, ufw fallido, nft list ruleset sin reglas).
  - Se listaron usuarios del sistema (cat /etc/passwd).
-

### **3. Definición de Niveles de Riesgo**

Los hallazgos en este informe se clasifican según el siguiente esquema de riesgo:

- **CRÍTICA:** Vulnerabilidades que permiten un compromiso total del sistema, acceso a datos altamente sensibles, o interrupción significativa del servicio con un esfuerzo relativamente bajo. Requieren acción inmediata.
  - **ALTA:** Vulnerabilidades que podrían permitir a un atacante obtener acceso no autorizado significativo, escalar privilegios, o acceder a datos sensibles. Requieren acción prioritaria.
  - **MEDIA:** Vulnerabilidades que exponen información útil para ataques posteriores, afectan la seguridad bajo condiciones específicas, o representan violaciones de las mejores prácticas con un impacto potencial moderado. Deben ser corregidas.
  - **BAJA:** Vulnerabilidades con impacto limitado o que requieren condiciones muy complejas para ser explotadas. Se recomienda corregir como parte del mantenimiento regular.
-

## 4. Hallazgos Detallados

### 4.1. Vulnerabilidades Críticas

#### 4.1.1. V-01: Acceso Root Obtenido vía Fuerza Bruta SSH

- **Riesgo:**  CRÍTICA (**¡COMPROBAMOS TOTAL!**)
- **Descripción:** Se obtuvo la contraseña (123456) del usuario root mediante un ataque de fuerza bruta contra el servicio SSH, aprovechando que permite el login de root (PermitRootLogin yes) y la autenticación por contraseña (PasswordAuthentication yes) junto con una contraseña extremadamente débil.
- **Evidencia:** La ejecución de hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://10.0.2.11 reportó: 1 valid password found [...] login: root password: 123456. Configuraciones de /etc/ssh/sshd\_config confirman PermitRootLogin yes, PasswordAuthentication yes.
- **Impacto:** Control total y absoluto del servidor Debian por parte del atacante. Acceso a todos los datos, capacidad para instalar malware, usar el servidor para otros ataques, etc.
- **Remediación:** **¡URGENTE!**
  1. **Cambiar INMEDIATAMENTE la contraseña del usuario** root por una contraseña extremadamente fuerte y única.
  2. Editar /etc/ssh/sshd\_config: Cambiar PermitRootLogin no y PasswordAuthentication no.
  3. Implementar y requerir autenticación por clave pública SSH.
  4. Actualizar OpenSSH (sudo apt update && sudo apt upgrade openssh-server).
  5. Reiniciar servicio (sudo systemctl restart sshd).
  6. **Revisar TODO el sistema** en busca de posibles backdoors o modificaciones realizadas por un atacante que ya pudo haber entrado. Considerar una reinstalación limpia si no se puede garantizar la limpieza.

#### 4.1.2. V-02: Permisos Inseguros en wp-config.php (777)

- **Riesgo:**  CRÍTICA
- **Descripción:** Archivo /var/www/html/wp-config.php con permisos -rwxrwxrwx.
- **Evidencia:** ls -la /var/www/html/wp-config.php. cat /var/www/html/wp-config.php muestra define( 'DB\_PASSWORD', '123456' );.
- **Impacto:** Exposición de credenciales de BD, modificación no autorizada. Agravado por V-01.
- **Remediación:** chmod 600 /var/www/html/wp-config.php; chown www-data:www-data /var/www/html/wp-config.php (ajustar usuario/grupo).

#### 4.1.3. V-03: Permisos Inseguros en Directorio Web y Archivos WP (777)

- **Riesgo:**  CRÍTICA
- **Descripción:** Múltiples directorios/archivos en /var/www/html con permisos 777.
- **Evidencia:** ls -la en /var/www/html y subdirs (wp-content, plugins).
- **Impacto:** Modificación no autorizada; subida de web shells. Agravado por V-01.
- **Remediación:** Aplicar permisos estándar WP (dirs 755, files 644). chown -R www-data:www-data /var/www/html.

### 4.2. Vulnerabilidades Altas

#### 4.2.1. V-04: Contraseña Débil y Reutilizada (root SSH y wordpressuser DB)

- **Riesgo:**  ALTA (La parte SSH ya es Crítica V-01)
- **Descripción:** La contraseña trivial '123456' es utilizada por root (comprometido) y por wordpressuser de la BD.
- **Evidencia:** Hydra encontró root:123456. wp-config.php contiene define( 'DB\_PASSWORD', '123456' );.
- **Impacto:** Mala práctica severa. Facilita acceso a la BD. Demuestra falta de políticas de contraseñas.
- **Remediación:** ¡URGENTE! Cambiar contraseña de wordpressuser en MySQL por una robusta y diferente a la nueva de root. Actualizar DB\_PASSWORD en wp-config.php. Implementar política de contraseñas seguras.

#### 4.2.2. V-05: WordPress Core Desactualizado (6.6.2)

- **Riesgo:**  ALTA
- **Descripción:** Versión de WordPress 6.6.2 (2024-09-10) está desactualizada.
- **Evidencia:** Salida de wpscan identifica la versión 6.6.2 como "Outdated".
- **Impacto:** Exposición a vulnerabilidades conocidas del núcleo de WordPress.
- **Remediación:** Actualizar WordPress a la última versión estable INMEDIATAMENTE (después de backup).

#### **4.2.3. V-06: Claves Secretas y Sales Débiles/Por Defecto en WordPress**

- **Riesgo:**  ALTA
- **Descripción:** Claves/sales criptográficas en wp-config.php usan valores placeholder.
- **Evidencia:** wp-config.php muestra 'put your unique phrase here' para AUTH\_KEY, etc.
- **Impacto:** Debilita seguridad de cookies de sesión y nonces.
- **Remediación:** Generar nuevas claves/sales únicas y reemplazarlas en wp-config.php.

#### **4.2.4. V-07: Servidor FTP con Acceso Anónimo y Vulnerabilidades Conocidas**

- **Riesgo:**  ALTA
- **Descripción:** FTP (vsftpd 3.0.3) permite login anónimo (con permisos limitados según prueba ftp) y posee CVEs conocidas.
- **Evidencia:** /etc/vsftpd.conf muestra anonymous\_enable=YES. Prueba ftp 10.0.2.11 confirma login anónimo exitoso pero restringido. Nmap identifica vsftpd 3.0.3 y lista CVEs. msfconsole search no encontró exploits directos.
- **Impacto:** Aumenta superficie de ataque, riesgo de explotación de CVEs, permite enumeración.
- **Remediación:** Deshabilitar acceso anónimo (anonymous\_enable=NO). Actualizar vsftpd. Considerar SFTP.

#### **4.2.5. V-08: Ausencia de Firewall de Host Activo**

- **Riesgo:**  ALTA
- **Descripción:** No hay reglas de firewall activas (nftables o ufw).
- **Evidencia:** Comandos ufw status y nft list ruleset sin reglas activas. Nmap confirma puertos 21, 22, 80 abiertos.
- **Impacto:** Servicios expuestos directamente a la red.
- **Remediación:** Instalar y configurar ufw. Política DENEGAR entrante por defecto. Permitir solo servicios necesarios (HTTP/80, HTTPS/443 si aplica, SSH/22 desde IPs de confianza).

#### 4.3. Vulnerabilidades Medias

##### 4.3.1. V-09: Listado de Directorios Habilitado en Servidor Web

- **Riesgo:**  MEDIA
- **Descripción:** Apache permite listado de directorios. WPScan confirma en /wp-content/uploads/.
- **Evidencia:** Config Apache (apache2.conf) con Options Indexes. WPScan: Upload directory has listing enabled....
- **Impacto:** Fuga de información.
- **Remediación:** Quitar Indexes de Options en config Apache o añadir Options -Indexes en .htaccess. Reiniciar Apache.

##### 4.3.2. V-10: Servicios Potencialmente Innecesarios Escuchando Localmente (CUPS)

- **Riesgo:**  MEDIA
- **Descripción:** Servicio de impresión CUPS (cupsd) escucha en puerto 631 local.
- **Evidencia:** netstat -tulnp.
- **Impacto:** Superficie de ataque local innecesaria.
- **Remediación:** Si no se necesita: sudo systemctl stop cups y sudo systemctl disable cups.

##### 4.3.3. V-11: Versiones de Software (Servicios) Desactualizadas/Vulnerables (vsftpd, OpenSSH)

- **Riesgo:**  MEDIA (El riesgo de OpenSSH se eleva a Crítico por V-01)
- **Descripción:** vsftpd 3.0.3 y OpenSSH 9.2p1 tienen CVEs conocidas listadas por Nmap.
- **Evidencia:** Resultados Nmap (-sV --script=vuln). msfconsole search no encontró exploits directos.
- **Impacto:** Riesgo de explotación. La vulnerabilidad de OpenSSH es prácticamente explotable vía fuerza bruta (V-01).
- **Remediación:** Actualizar paquetes: sudo apt update && sudo apt dist-upgrade. (Actualizar OpenSSH y vsftpd ya está en otras recomendaciones).

#### 4.4. Vulnerabilidades Bajas / Informativas

##### 4.4.1. V-12: XML-RPC Habilitado en WordPress

- **Riesgo:**  **BAJA**
  - **Descripción:** Interfaz XML-RPC habilitada.
  - **Evidencia:** WPScan detecta: XML-RPC seems to be enabled....
  - **Impacto:** Puede usarse para fuerza bruta o amplificación DDoS.
  - **Remediación:** Deshabilitar XML-RPC si no es necesario.
- 

#### 5. Conclusiones y Próximos Pasos

El servidor Debian 10.0.2.11 está **críticamente comprometido**. La capacidad de obtener acceso root mediante fuerza bruta SSH con una contraseña trivial (123456), reutilizada en la base de datos, junto con permisos de archivo peligrosamente laxos, software desactualizado y la ausencia de un firewall, crean un escenario de riesgo máximo.

La **remediación debe ser inmediata y exhaustiva**:

1. **CONTENCIÓN Y ERRADICACIÓN (¡YA!):**

- **Cambiar contraseña de root INMEDIATAMENTE.**
- **Asegurar SSH:** Deshabilitar login root/password auth, usar claves, actualizar.
- **Revisar TODO el sistema** en busca de actividad maliciosa/backdoors.  
Considerar **reinstalación limpia**.

2. **REMEDIACIÓN URGENTE (Post-contención):**

- **Corregir TODOS los permisos** en /var/www/html.
- **Asegurar** wp-config.php (permisos, cambiar pass DB, generar sales).
- **Actualizar WordPress Core, temas y plugins.**
- **Configurar y activar un firewall (ufw).**
- **Asegurar/Deshabilitar FTP y actualizar vsftpd.**
- **Cambiar contraseña de wordpressuser** en la BD (diferente a root).
- Deshabilitar CUPS y XML-RPC si no son necesarios.
- Realizar apt dist-upgrade.

**DADA LA SEVERIDAD DEL COMPROMISO DEMOSTRADO, ES CRUCIAL ACTUAR CON RAPIDEZ. SE RECOMIENDA ENCARECIDAMENTE UNA AUDITORÍA FORENSE SI ESTE SISTEMA HA ESTADO EXPUESTO O HA MANEJADO DATOS SENSIBLES.**

#### CAPTURAS DE PANTALLA ADICIONALES.

