7-5-2025

INFORME DE AUDITORÍA

SecurityCat

DARÍO CAIANI TARAZONA
PABLO GARCÍA TORRES

EMPRESA: SecurityCat

1. Resumen ejecutivo

En el siguiente informe, se pretende mostrar los resultados obtenidos de la prueba de pentesting realizada al servidor Debian de la empresa 4Geeks. Nuestro principal objetivo ha sido identificar vulnerabilidades de seguridad en el sistema, mediante las cuales se pudiera comprometer la confidencialidad, integridad o disponibilidad de los activos.

Se ha realizado pruebas de penetración de tipo caja negra y se ha utilizado una metodología basada en el estándar OWASP, incluyendo fases de reconocimiento, enumeración, explotación y reporte.

1.1 Resultados destacados.

Los principales datos obtenidos han sido un total de 17 vulnerabilidades con CVEs asociados y diferentes vulnerabilidades por configuraciones de servicios inseguras.

- 4 de severidad crítica.
- 6 de severidad alta.
- 6 de severidad media.
- 1 de severidad baja.

Las vulnerabilidades criticas afectan principalmente al protocolo SSH, pudiendo verse comprometidas las comunicaciones a través de la red.

1.2 Impacto potencial.

La explotación de las vulnerabilidades descubiertas podría resultar en acceso no autorizado, con lo que esto implica, exposición de datos sensibles e interrupciones de servicios críticos, causando un daño reputacional a la empresa.

1.3 Conclusiones.

Puesto que el riesgo de exposición actual es alto, se recomienda actuar de manera inmediata sobre las vulnerabilidades de mayor criticidad para disminuir el riesgo de incidentes de seguridad.

1.4 Recomendaciones generales.

- Actualización del sistema e instalación de parches de seguridad.
- Realizar una gestión de contraseñas robusta.
- Configuración de servicios segura.
- Implementación de firewalls.

2. Información General del Sistema Evaluado

2.1 IPs y/o dominios analizados.

Durante el proceso de pentesting se analizaron los siguientes activos:

- IP Privada:
    - 10.0.2.10

Este activo fue sometido a un análisis de vulnerabilidades, pruebas de intrusión y evaluación de las configuraciones de seguridad de sus servicios.

2.2 Tipo de sistema.

El activo a realizar dichas pruebas es un servidor crítico de la empresa 4GEEKS Academy. Es un servidor Debian, cuya versión es:

Linux debían 6.1.0-33amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.133-1 (2025-04-10) x86_64 GNU/LINUX

2.3 Tecnologías detectadas.

En las pruebas realizadas identificamos las siguientes tecnologías:

- Servidor Web Apache v.2.4.62
- CMS WordPress v.
- Servidor SSH OpenSSH v.9.2p1
- Servidor FTP vsftpd v.3.0.3
- Sistema Operativo Linux 5.6.3 (Debian 6.1.0)

3. Metodología de Evaluación

3.1 Tipo de pentesting.

En este caso, se ha realizado una prueba de penetración de tipo caja negra, puesto la información proporcionada ha sido la IP del activo sin más datos adicionales.

3.2 Fases aplicadas.

La ejecución del proceso de pentesting se ha realizado en las siguientes fases:

- **Reconocimiento:** se realiza una identificación pasiva y activa del objetivo y un escaneo controlado del mismo.
- **Enumeración y fingerprinting**: en esta fase obtenemos toda la información posible sobre versiones, tecnologías y configuraciones presentadas.
- **Análisis de vulnerabilidades**: identificamos, cuantificamos y priorizamos las debilidades encontradas en el sistema. Este proceso se realiza siguiendo buenas prácticas OWASP.
- **Explotación**: se la realiza la explotación de las vulnerabilidades críticas encontradas para verificar su posible impacto.

- **Documentación de hallazgos:** enumeramos las vulnerabilidades encontradas, riesgos asociados e indicamos recomendaciones para su mitigación.

4. Análisis Técnico

Mediante la aplicación de distintas técnicas de pentesting y el uso de herramientas especializadas, se ha conseguido extraer información detallada de la terminal analizada, identificando configuraciones expuestas y realizado a través de estas una intrusión a la terminal, listando directorios sensibles y posibles vulnerabilidades de configuración que podrían ser aprovechadas en un escenario de ataque.

4.1 Explotación del Sistema.

Indicar que se ha podido acceder a la máquina objetivo mediante fuerza bruta. Esto ha sido posible por debilidades en los servicios FTP y SSH. Indicamos el proceso que ha llevado a cabo.

VULNERABILIDAD WEAK PASSWORD POLITICAL.

Mediante la herramienta **Hydra** se ha podido obtener acceso remoto a través del puerto FTP por el uso de fuerza bruta y una lista de usuarios y contraseñas de uso común. Se identificaron los usuarios “root” y “debian”.

- El comando utilizado para el acceso a la máquina objetivo en la herramienta Hydra ha sido.

```
hydra -L /usr/share/wordlists/usuarios.txt -P
/usr/share/wordlists/rockyou.txt -t 16 [IP objetivo] ftp -vV
```

- Obteniendo las credenciales de los usuarios “root” y “debian”. Usuarios con privilegios, teniendo el **control total** de la máquina.

- **root: 123456** - **debian: 123456**

MITIGACIÓN

- ☑ Establecer una política de contraseñas robusta.
- ☑ Crear usuarios específicos para el servicio FTP.
- ☑ Instalar y configurar Fail2ban para limitar intentos de acceso.

```
sudo apt-get update
sudo apt-get install fail2ban -y
cat <<EOF | sudo tee /etc/fail2ban/jail.d/vsftpd.conf
[vsftpd]
enabled  = true
port     = ftp,ftp-data,ftps,ftps-data
filter   = vsftpd
logpath  = /var/log/vsftpd.log
maxretry = 5
bantime  = 3600  # tiempo de baneo en segundos
EOF
sudo systemctl restart fail2ban
```

SSH PASSWORD VULNERABILITY.

El protocolo SSH tiene activo el uso de contraseña para la autenticación al servicio y mediante la misma técnica usada para acceder al servicio FTP, fuerza bruta, podemos acceder a la máquina víctima.

Como se ha indicado, mediante la técnica de pentesting fuerza bruta, se consiguen las credenciales de los usuarios “root” y “debian”. Indicar que son usuarios con privilegios, con lo que tenemos un **control total** de la máquina atacada.

- **root: 123456** - **debian: 123456**

MITIGACIÓN

☑ Deshabilitar autenticación por contraseña y habilitar el uso de autenticación mediante clave cifrada (pública y privada).

Bash.

```
sudo sed -i 's/^#PubkeyAuthentication.*/PubkeyAuthentication yes/' /etc/ssh/sshd_config
sudo sed -i 's/^PasswordAuthentication.*/PasswordAuthentication no/' /etc/ssh/sshd_config
sudo systemctl reload sshd
```

4.2 Configuración de Servicios y Recomendaciones

Se indican una serie de cambios en el sistema y en las configuraciones para mejorar la seguridad del mismo y los diferentes servicios ubicados en él.

---

1. Actualizaciones del sistema

Configuración detectada

- Paquetes desactualizados.
- Sin actualizaciones automáticas.

Impacto: Riesgo elevado por falta de parches de seguridad.

Recomendaciones

- Actualizar el sistema manualmente:

```
sudo apt-get update -y && sudo apt-get upgrade -y
```

- Habilitar actualizaciones automáticas mediante `unattended-upgrades`:

```
sudo apt-get install unattended-upgrades
sudo dpkg-reconfigure unattended-upgrades
```

- Programar actualizaciones regulares con `cron`:

- Editar el crontab del sistema:

```
sudo crontab -e
```

- Añadir la siguiente línea para ejecutar actualizaciones cada lunes a medianoche:

```
0 0 * * 1 apt update && apt upgrade -y
```

2. Servicio vsFTPd (versión 3.0.3-13+b2)

Configuración detectada

```
listen = NO
listen_ipv6 = YES
anonymous_enable = YES
ssl_enable = NO
#chroot_local_user = YES
connect_from_port_20 = YES
```

Recomendaciones

```
listen = YES                # Modo standalone
enable_ipv6 = NO            # Desactivar IPv6 si no se
necesita
anonymous_enable = NO     # Deshabilitar acceso anónimo
ssl_enable = YES          # Habilitar SSL/TLS
chroot_local_user = YES   # Encerrar usuarios en su
directorio home
connect_from_port_20 = NO # Desactivar modo activo si no es
necesario
```

Para editar:

```
sudo nano /etc/vsftpd.conf
```

Reiniciar servicio:

```
sudo systemctl restart vsftpd
```

3. Servicio SSH (SSHD) (versión 1:9.2p1-2+deb12u3)

Configuración detectada

```
PermitRootLogin = YES
PasswordAuthentication = YES
#PubkeyAuthentication = YES
X11Forwarding = YES
#MaxAuthTries = 6
#PermitEmptyPasswords = NO
```

Recomendaciones

```
PermitRootLogin no          # Deshabilitar inicio de sesión
como root
PasswordAuthentication no   # Deshabilitar autenticación por
contraseña
PubkeyAuthentication yes    # Habilitar autenticación por
clave pública
X11Forwarding no            # Deshabilitar reenvío X11
MaxAuthTries 3              # Limitar intentos de
autenticación
PermitEmptyPasswords no     # No permitir contraseñas vacías
```

Para editar:

```
sudo nano /etc/ssh/sshd_config
```

Reiniciar servicio:

```
sudo systemctl restart sshd
```

Configuración detectada

Principal (**/etc/apache2/apache2.conf**)

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

<Directory /usr/share/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
<Directory />
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>
```

VirtualHost HTTP (**/etc/apache2/sites-enabled/000-default.conf**)

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    <Directory /var/www/html>
        AllowOverride All
```

```
    </Directory>
</VirtualHost>
```

Recomendaciones

Desactivar sitio HTTP:

```
sudo a2dissite 000-default.conf
```

Configurar y habilitar sitio SSL:

sudo nano /etc/apache2/sites-available/default-ssl.conf:

```
    <VirtualHost *:443>
        ServerName www.2x2geeks.com
        DocumentRoot /var/www/html
        SSLEngine on
        SSLCertificateFile /etc/ssl/certs/2x2geeks.com.crt
        SSLCertificateKeyFile
    /etc/ssl/private/2x2geeks.com.key
        <Directory /var/www/html>
            AllowOverride None
            Require all granted
        </Directory>
    </VirtualHost>
```

Activar HTTPS y recargar Apache:

```
sudo a2ensite default-ssl.conf
sudo systemctl reload apache2
```

Cambiar permisos en `/var/www/html`

```
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/
```

Configuración cliente (`/etc/mysql/debian.cnf`)

Configuración detectada

```
User = root
```

Recomendación

```
User = debian-sys-maint
Password = <clave segura>
```

Configuración SSL (`/etc/mysql/mariadb.conf.d/50-client.cnf`)

Configuración detectada

```
ssl-cert = /etc/mysql/client-cert.pem
ssl-key = /etc/mysql/client-key.pem
ssl-verify-server-cert = on
```

Recomendación

- Revisar privilegios de usuarios y aplicar principio de menor privilegio.
- Aplicar una política de gestión de roles y privilegios.

6. Firewall

Detección: No se han implementado reglas de firewall.

Recomendación:

sudo apt-get install ufw

sudo ufw default deny incoming

sudo ufw default allow outgoing

sudo ufw allow ssh

sudo ufw enable

Configurar reglas adicionales según servicios activos.

5. Análisis de CVEs y Recomendaciones

En el presente punto se van a indicar las distintas vulnerabilidades detectadas en el sistema. Se adjunta tabla de valores de criticidad.

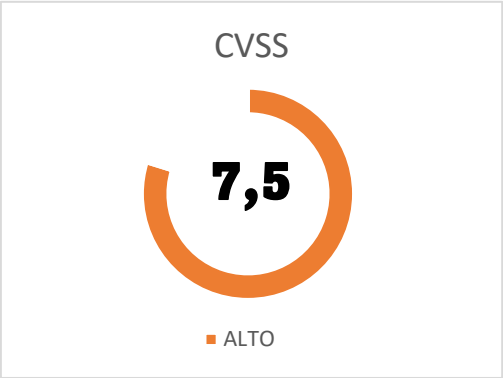
PUNTUACIÓN DE CVSS	GRAVEDAD EN ASESORAMIENTO
9.0 – 10.0	Crítico
7.0 – 8.9	Alto
4.0 – 6.9	Medio
0.1 – 3.9	Bajo

5.1 Vulnerabilidades detectadas en FTP (vsFTPd. Versión 3.0.3-13+b2)

Se identifican los CVEs del servicio FTP detectados en el sistema, se indica su id de CVE, riesgo, alcance y acción recomendada para corregirlo.

❖ **CVE-2021-30047**

- **ESTADO:** ABIERTO
- **RIESGO:** ALTO



ALCANCE:

VSFTPD 3.0.3) Permite a un atacante remoto saturar el servidor FTP hasta impedir nuevas conexiones, ocasionando denegación de servicio (disponibilidad afectada)

ACCIÓN RECOMENDADA:

- Actualizar VSFTPD a la última versión parcheada.

  bash

  sudo apt update && sudo apt upgrade -y

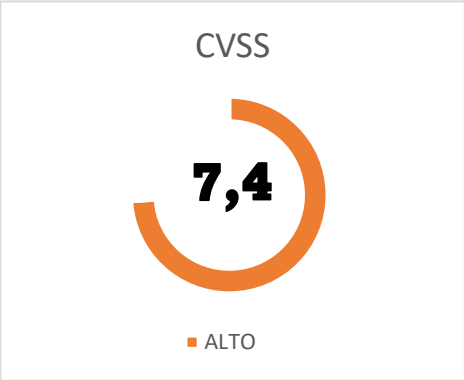
- Deshabilitar el servicio FTP si no es imprescindible

  bash

  sudo systemctl stop vsftpd

  systemctl disable vsftpd

❖ **CVE-2021-3618**



- **ESTADO:** ABIERTO
- **RIESGO:** ALTO

ALCANCE:

VSFTPD 3.0.3) Es un ataque de confusión de contenido TLS (“ALPACA”): un MitM que controle tráfico TCP/IP puede redirigirlo entre subdominios con certificados compatibles (multi-dominio/wildcard), rompiendo la autenticación TLS de VSFTPD y posibilitando ataques cruzados entre servicios. Esto podría comprometer la confidencialidad e integridad de sesiones FTPS.

ACCIÓN RECOMENDADA:

- Actualizar VSFTPD a la última versión parcheada.

  bash

  sudo apt update && sudo apt upgrade -y

- Deshabilitar el servicio FTP si no es imprescindible

  bash

  sudo systemctl stop vsftpd

  systemctl disable vsftpd

- Evitar certificados multi-dominio en VSFTPD o usar certificados dedicados por servicio.
- Reforzar las suites TLS y deshabilitar versiones obsoletas (SSLv3, TLS 1.0/1.1).

sudo nano /etc/vsftpd.conf

ssl_tlsv1_2=YES

ssl_tlsv1_3=YES

ssl_sslv2=NO

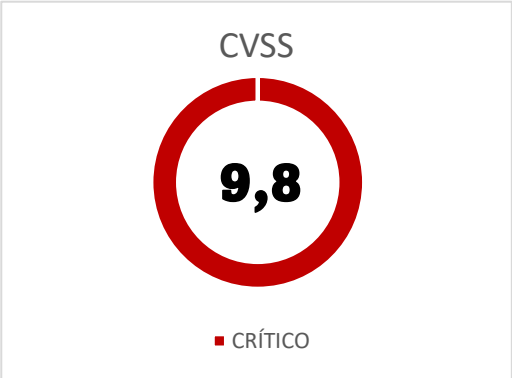
ssl_sslv3=NO

5.2 Vulnerabilidades detectadas en SSH (OpenSSH v.9.2p1)

A continuación, indicamos los CVEs del servicio SSH, identificando su ID de CVE, riesgo, alcance y acción recomendada para mitigar la vulnerabilidad.

❖ **CVE-2023-38408**

- **ESTADO:** ABIERTO
- **RIESGO:** CRÍTICO



ALCANCE:

Afecta al ssh-agent; un atacante remoto en un servidor al que se haya reenviado el agente puede forzar que éste cargue librerías arbitrarias desde /usr/lib en la máquina cliente con muy alto impacto; el atacante lograría ejecución de código remoto con privilegios del usuario (posible escalada total del sistema cliente) si el agente SSH del usuario está abierto y reenviado.

ACCIÓN RECOMENDADA:

- Actualizar OpenSSH a ≥9.3p2.

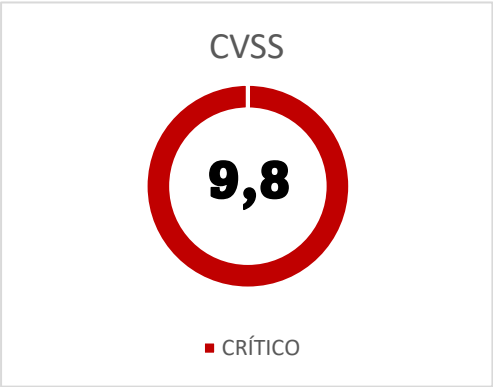
  sudo apt update && sudo apt upgrade -y

- Deshabilitar el reenviado de agente si no es imprescindible, esto evitará que un servidor remoto malicioso pueda acceder al ssh-agent del usuario.

  sudo nano /etc/ssh/sshd_config
  AllowAgentForwarding no

❖ **CVE-2023-28531**

- **ESTADO:** ABIERTO
- **RIESGO:** CRÍTICO



ALCANCE:

En versiones <9.3 afecta el comando ssh-add: llaves de smartcard pueden agregarse al ssh-agent sin las restricciones de destino esperadas. Esto implica que un atacante que influya en el agente (por ejemplo, al reutilizar agentes en hops múltiples) podría usar la clave de smartcard más allá del alcance previsto, potencialmente accediendo a sistemas intermedios sin autorización. Aunque requiere entorno especial (smartcard), el impacto puede ser grave al comprometer la seguridad de claves de autenticación.

ACCIÓN RECOMENDADA:

- Actualizar OpenSSH a la versión parcheada.

  bash

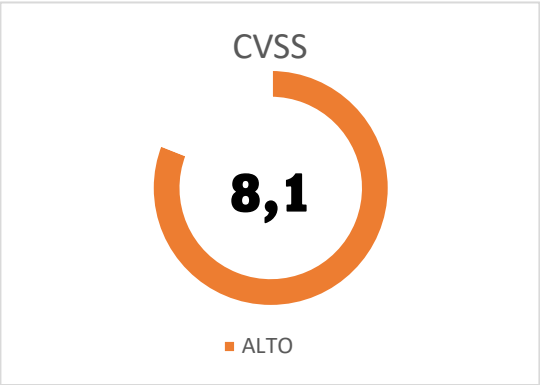
  sudo apt update && sudo apt upgrade -y

- Limpiar el agente tras uso de smartcards.

  bash

  ssh-add -D

❖ **CVE-2024-6387**



- **ESTADO:** ABIERTO
- **RIESGO:** ALTO

ALCANCE:

En versiones <9.3 un atacante remoto sin credenciales puede enviar múltiples peticiones de conexión fallidas hasta que venza el tiempo de gracia.

Entonces sshd maneja una señal (SIGALRM) de forma insegura, permitiendo ejecución de código arbitrario como root.

ACCIÓN RECOMENDADA:

- Actualizar OpenSSH a la versión parcheada, este ataque afecta la configuración por defecto, por lo que la actualización es la medida principal

  bash

  sudo apt update && sudo apt upgrade -y

- Reducir LoginGraceTime y restringir usuarios.

  bash

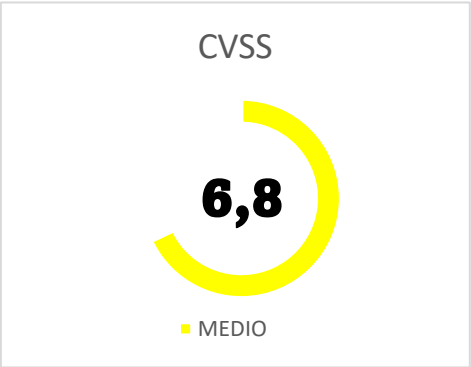
  sudo nano /etc/ssh/sshd_config

  LoginGraceTime 30

  AllowUsers usuario1,usuario2

❖ **CVE-2025-26465**

- **ESTADO:** ABIERTO
- **RIESGO:** MEDIO



ALCANCE:

En versiones <9.3 Se trata de un ataque MITM contra el cliente SSH cuando VerifyHostKeyDNS está habilitado (opción por defecto desactivada en Debian). Si un atacante logra forzar al cliente a consumir toda su memoria, la verificación de la clave del host puede fallar lo siguiente y el atacante se hace pasar por servidor legítimo. Esto permite interceptar o modificar la sesión SSH (confidencialidad/integridad comprometida). La complejidad es alta (agotar memoria), pero el riesgo existe si se habilita esta opción.

ACCIÓN RECOMENDADA:

- Actualizar OpenSSH a la versión parcheada, este ataque afecta la configuración por defecto, por lo que la actualización es la medida principal

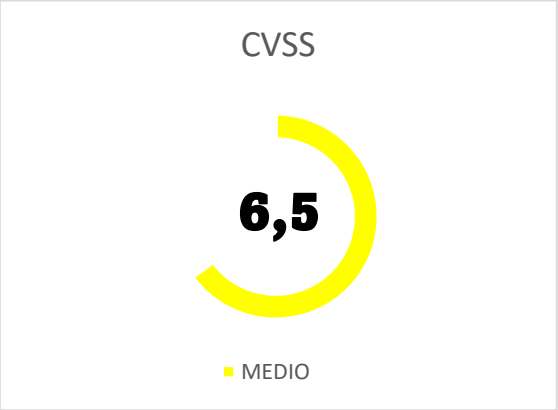
  bash
  sudo apt update && sudo apt upgrade -y

- Asegurarse de que VerifyHostKeyDNS no. si se usa, considerar UseDNS no

  bash
  sudo nano /etc/ssh/sshd_config
  VerifyHostKeyDNS no
  StrictHostKeyChecking yes

❖ **CVE-2023-51385**

- **ESTADO:** ABIERTO
- **RIESGO:** MEDIO



ALCANCE:

En versiones <9.3 Permite que un repositorio Git malicioso con nombre de usuario o host que incluya caracteres especiales (;, &&, etc.) ejecute comandos OS en el cliente al procesar estas cadenas en tokens de expansión (p. ej. %(user)s). El adversario podría así ejecutar código con los permisos del usuario que realiza la conexión SSH o la operación Git, obtener información sensible o dañar archivos.

ACCIÓN RECOMENDADA:

- Actualizar OpenSSH a la versión parcheada, este ataque afecta la configuración por defecto, por lo que la actualización es la medida principal

  bash
  sudo apt update && sudo apt upgrade -y

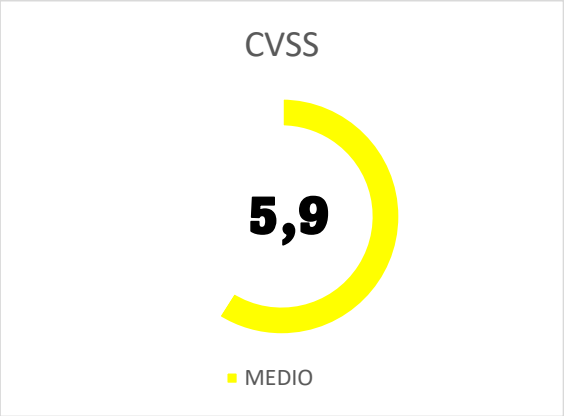
- Limpiar el agente tras el uso.

  bash
  ssh-add -D

- Verificar nombres de host/usuario en entorno donde se haga Git clone.

❖ **CVE-2023-48795**

- **ESTADO:** ABIERTO
- **RIESGO:** MEDIO



ALCANCE:

En versiones <9.3) Facilita un ataque MITM: manipulando los números de secuencia durante el handshake SSH, el atacante puede eliminar mensajes de negociación de extensiones, provocando que el cliente y servidor queden en un canal de comunicación con seguridad debilitada (algunas características de seguridad desactivadas). Ej: Un atacante podría degradar el cifrado al usar chacha20-poly1305@openssh.com o CBC-EtM con integridad comprometida. Esto compromete confidencialidad e integridad de datos transmitidos (secuencias de paquetes pueden ser alteradas).

ACCIÓN RECOMENDADA:

- Este fallo exige OpenSSH ≥9.6 para corregirlo. Como Debian 12 incluye 9.2 y no hay parche oficial, se recomienda planificar la actualización a Debian 13 o superior cuando sea posible, además se recomiendan las siguientes mitigaciones en el S.O Debian 12

- Actualizar OpenSSH para evitar otros problemas

  bash

  sudo apt update && sudo apt upgrade -y

- Modificar la configuración SSH para deshabilitar los algoritmos afectados y excluir ciphers inseguros.

bash

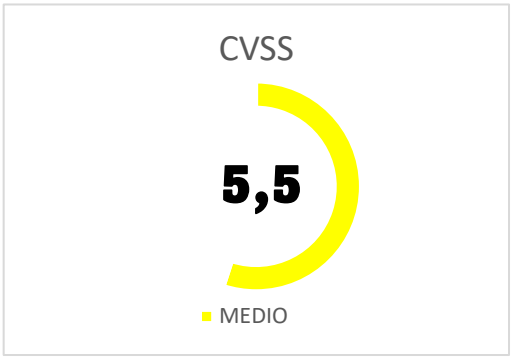
sudo nano /etc/ssh/sshd_config/ssh_config

Ciphers aes128-ctr,aes192-ctr,aes256-ctr

MACs hmac-sha2-256,hmac-sha2-512

KexAlgorithms ***curve25519-sha256@libssh.org,diffie-hellman-group14-sha256***

❖ **CVE-2023-51384**



- **ESTADO:** ABIERTO
- **RIESGO:** MEDIO

ALCANCE:

En versiones <9.3, en contexto de PKCS#11, implica que, si una tarjeta inteligente devuelve múltiples llaves, sólo la primera llevará las restricciones esperadas; las llaves adicionales podrían usarse sin control. Un atacante local con acceso al token podría, en teoría, añadir llaves adicionales al agente y usarlas de forma no autorizada.

ACCIÓN RECOMENDADA:

- Actualizar OpenSSH a la versión parcheada.

  bash

  sudo apt update && sudo apt upgrade -y

- Evitar usar múltiples llaves PKCS#11 en una sola operación de ssh-add.

- Agregar llaves PKCS#11 de una en una con restricciones explícitas.

  bash

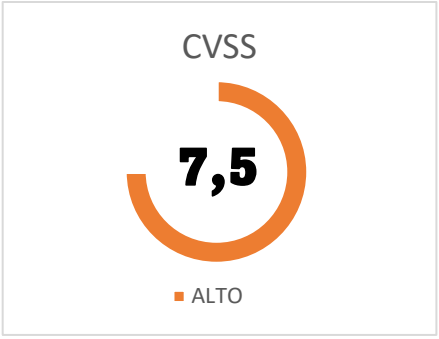
  ssh-add -c /usr/share/tulibreria/libpkcs11.so

- Evitar agentes en entornos no confiables.

5.3 Vulnerabilidades detectadas sin CVEs asociados.

Se describen las vulnerabilidades detectadas sin CVE asociado. Al igual que anteriormente, se indica a que servicio pertenecen, el riesgo estimado de la vulnerabilidad, alcance, como se podría explotar y la acción recomendada para mitigar su explotación.

❖ **FTP sin cifrado SSL/TLS en VSFTPD**



- **ESTADO:** ABIERTO
- **RIESGO:** ALTO

ALCANCE:

VSFTPD 3.0.3) El servicio FTP opera en texto plano (ssl_enable=NO), por lo que credenciales y datos viajan sin cifrado. Esto expone la información a intercepción o manipulación en la red.

MECANISMO DE EXPLOTACIÓN:

- **Sniffing:** Un atacante en la misma red puede usar herramientas como Wireshark o tcpdump para capturar y leer paquetes no cifrados.
- **Man-in-the-Middle (MitM):** Un actor malicioso puede redirigir o alterar el tráfico FTP para robar credenciales, modificar archivos durante la transferencia o inyectar malware en descargas legítimas.
- **Ataques de repetición (Replay Attacks):** Un atacante puede capturar comandos de sesión válidos (ej. autenticación) y reenviarlos para suplantar al usuario legítimo.

ACCIÓN RECOMENDADA:

- Habilitar SSL/TLS en VSFTPD

  bash

  sudo nano /etc/vsftpd.conf

  ssl_enable=YES

  ```
  Forzar uso de TLS (versión moderna y segura)
  ```

  ssl_tlsv1=YES

  ssl_sslv2=NO

  ssl_sslv3=NO

  ```
  Ruta del certificado y clave SSL
  ```

  rsa_cert_file=/etc/ssl/certs/vsftpd.pem

  rsa_private_key_file=/etc/ssl/private/vsftpd.key

  ```
  Obligar a los clientes a usar cifrado
  ```

  require_ssl_reuse=NO

  force_local_data_ssl=YES

  force_local_logins_ssl=YES

- Generar un certificado SSL

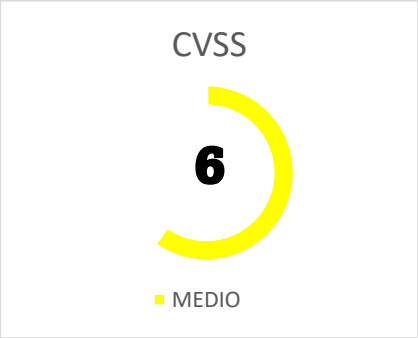
  bash

  sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout

  /etc/ssl/private/vsftpd.key -out /etc/ssl/certs/vsftpd.pem

❖ **Conexiones FTP anónimas habilitadas en VSFTPD**

- **ESTADO:** ABIERTO
- **RIESGO:** MEDIO



ALCANCE:

VSFTPD (v.3.0.3). La directiva anonymous_enable = YES en el archivo /etc/vsftpd.conf habilita conexiones FTP sin autenticación, permitiendo que cualquier usuario acceda al servidor con el nombre de usuario anonymous o ftp, sin necesidad de contraseña válida.

- **Encapsulamiento en directorio**: Por defecto, los usuarios anónimos son restringidos al directorio /var/ftp mediante el encapsulamiento (*chroot*), lo que limita su acceso al resto del sistema 611.
- **Permisos predeterminados**: Inicialmente, los usuarios anónimos solo tienen permisos de lectura (anon_world_readable_only = YES), pero configuraciones incorrectas pueden permitir escritura, creación de directorios o incluso eliminación de archivos 128.

MECANISMO DE EXPLOTACIÓN:

- **Exposición de información sensible**: Si el directorio "/var/" contiene información sensible marcada como "pública", cualquier usuario podría descargarla sin restricciones.
- **Consumo de recursos**: Conexiones masivas anónimas podrían saturar el ancho de banda o espacio en disco, afectando a otros servicios.
- **Fuga de metadatos**: Aunque los usuarios están encapsulados, errores en permisos o enlaces simbólicos podrían exponer rutas del sistema real.
- **Ataques de fuerza bruta**: Si se combina con otras vulnerabilidades, el atacante podría ganar acceso a usuarios legítimos.

ACCIÓN RECOMENDADA:

- Desactivar el acceso anónimo

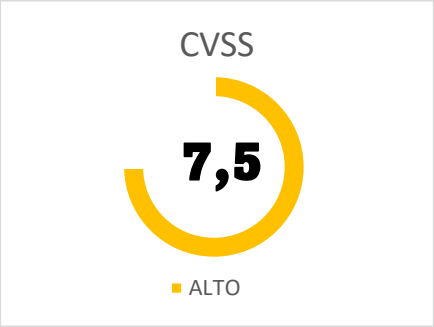
bash

sudo nano /etc/vsftpd.conf

sudo sed -i 's/anonymous_enable=YES/anonymous_enable=NO/' /etc/vsftpd.conf

sudo systemctl restart

❖ **Autenticación root por servicio SSH**



- **ESTADO:** ABIERTO
- **RIESGO:** ALTO

ALCANCE:

OpenSSH (v.9.2p1). La directiva PermitRootLogin = YES en el archivo /etc/ssh/sshd_config habilita el acceso remoto a la cuenta **root** mediante SSH, la cual tiene privilegios administrativos absolutos en el sistema. Esto crea un vector de ataque crítico ya que el atacante podría:

- Instalar malware o ransomware, modificar registros para ocultar actividades maliciosas o acceder a bases de datos, claves cifradas y archivos sensibles.
- Desde el servidor comprometido, el atacante podría pivotar a otros sistemas en la red interna.
- **PCI-DSS**: Prohíbe explícitamente el acceso remoto como root (Req. 2.2.4).
- **HIPAA/GDPR**: Un compromiso de root violaría los principios de *mínimo privilegio* y *protección de datos*.

MECANISMO DE EXPLOTACIÓN:

- **Ataques de fuerza bruta:** Herramientas como **Hydra** automatizan intentos masivos de contraseñas para acceder como root.

ACCIÓN RECOMENDADA:

- Deshabilitar el acceso root por SSH

bash

sudo nano /etc/ssh/sshd_config:

PermitRootLogin

PasswordAuthentication no

- Crear un usuario no privilegiado con sudo

bash

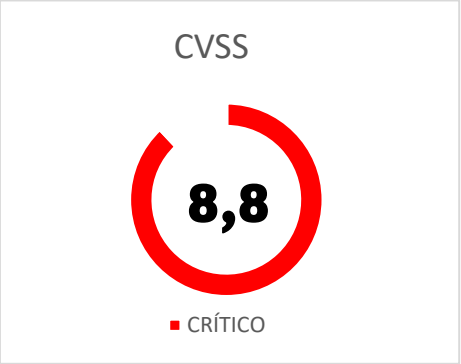
adduser admin_user

usermod -aG sudo admin_user

systemctl restart sshd

❖ **Autenticación por contraseña habilitada en SSH**

- **ESTADO:** ABIERTO
- **RIESGO:** CRÍTICO



ALCANCE:

OpenSSH (v.9.2p1). La autenticación por contraseña en SSH (habilitada mediante PasswordAuthentication = yes en /etc/ssh/sshd_config) permite que los usuarios accedan al sistema introduciendo una contraseña en lugar de usar claves criptográficas. Las implicaciones de esta vulnerabilidad son:

- **Acceso no autorizado al sistema:** Esto implica ejecución de código malicioso, escalada de privilegios o robo de datos.
- **Impacto en la infraestructura:** Ataques de denegación de servicio (DoS) y pivoteo en la red interna.
- **PCI-DSS**: Prohíbe explícitamente el acceso remoto como root (Req. 2.2.4).
- **HIPAA/GDPR**: Un compromiso de root violaría los principios de *mínimo privilegio* y *protección de datos*.

MECANISMO DE EXPLOTACIÓN:

- **Realizar ataques de fuerza bruta** y obtener las credenciales de acceso en poco tiempo, ya que no existe encriptación.
- **Sniffing de redes** lo cual le permitiría obtener las credenciales cuando un usuario se autenticase.

ACCIÓN RECOMENDADA:

- Deshabilitar autenticación por contraseña y utilizar claves SSH, así como 2FA

bash

Generar clave en el cliente (Ej. ed25519):

ssh-keygen -t ed25519 -a 100 -f ~/.ssh/id_ed25519

Copiar clave pública al servidor:

ssh-copy-id -i ~/.ssh/id_ed25519.pub usuario@servidor

sudo nano /etc/ssh/sshd_config: eliminar auth. contraseña

PasswordAuthentication no

ChallengeResponseAuthentication no

sudo nano /etc/ssh/sshd_config:

PermitRootLogin

PasswordAuthentication no

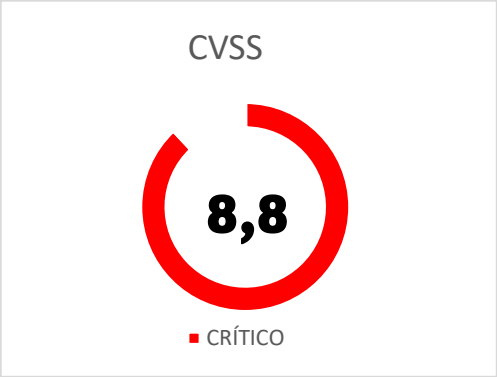
Implementar autenticación multifactor (2FA)

sudo apt install libpam-google-authenticator

auth required pam_google_authenticator.so

❖ **Credenciales débiles en SSH**

- **ESTADO:** ABIERTO
- **RIESGO:** CRÍTICO



ALCANCE:

OpenSSH (v.9.2p1) El uso de **credenciales débiles** para autenticación en SSH permite a los atacantes realizar **ataques de fuerza bruta** o **diccionario** para comprometer cuentas. Esto se agrava si el servidor permite autenticación por contraseña (PasswordAuthentication = YES) en lugar de claves SSH. Las implicaciones más importantes son:

- **Ejecución de código arbitrario**: Instalación de malware, minería de criptomonedas…
- **Escalada de privilegios**: Si el usuario comprometido tiene permisos sudo, el atacante puede obtener acceso root.
- **Exfiltración de datos**: Robo de bases de datos, claves SSH, certificados SSL o información sensible.
- **Ataques de ransomware**: Cifrado de archivos críticos para exigir rescates.
- **PCI-DSS**: Prohíbe explícitamente el acceso remoto como root (Req. 2.2.4).
- **HIPAA/GDPR**: Un compromiso de root violaría los principios de *mínimo privilegio* y *protección de datos*.

MECANISMO DE EXPLOTACIÓN:

- **Ataques de fuerza bruta:** Intentos masivos de contraseñas para acceder como root.
- **Credential stuffing**: Reutilización de credenciales filtradas.

ACCIÓN RECOMENDADA:

- Eliminar la autenticación por contraseña y habilitar acceso por certificado.

bash

sudo nano /etc/ssh/sshd_config:

ssh-keygen -t ed25519 -a 100

Copiar clave pública al servidor:

ssh-copy-id -i ~/.ssh/id_ed25519.pub usuario@servidor

Deshabilitar contraseñas en **/etc/ssh/sshd_config**:

PasswordAuthentication no

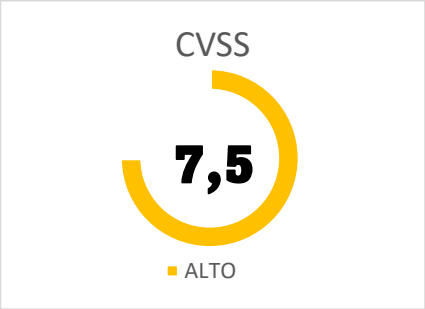
ChallengeResponseAuthentication no

PermitRootLogin

PasswordAuthentication no

❖ **Credenciales débiles en FTP**

- **ESTADO:** ABIERTO
- **RIESGO:** ALTO



ALCANCE:

VSFTPD (v.3.0.3) El uso de **credenciales débiles** para autenticación en SSH permite a los atacantes realizar **ataques de fuerza bruta** o **diccionario** para comprometer cuentas. Esto se agrava si el servidor permite autenticación por contraseña en lugar de claves SSH. Las implicaciones más importantes son:

- **Ejecución de código arbitrario**: Instalación de malware, minería de criptomonedas…
- **Escalada de privilegios**: Obtener acceso root si el usuario es sudoer.
- **Exfiltración de datos**: Robo de bases de datos, claves SSH…
- **Ataques de ransomware**: Cifrado de archivos críticos para exigir rescates.
- **PCI-DSS**: Prohíbe explícitamente el acceso remoto como root (Req. 2.2.4).
- **HIPAA/GDPR**: Un compromiso de root violaría los principios de mínimo privilegio y protección *de* datos.

MECANISMO DE EXPLOTACIÓN:

- **Ataques de fuerza bruta:** Intentos masivos de contraseñas para acceder como root.
- **Credential stuffing**: Reutilización de credenciales filtradas.

ACCIÓN RECOMENDADA:

- Usar claves RSA de 4096 bits

  bash
  openssl genrsa -out /etc/ssl/private/vsftpd.key 4096

- Deshabilitar protocolos y cifrados inseguros

  bash
  sudo nano /etc/ssl/private/vsftpd.conf
  ssl_tlsv1=NO
  ssl_sslv2=NO
  ssl_sslv3=NO
  allow_anon_ssl=NO
  force_local_logins_ssl=YES

- Permisos restrictivos:

  bash

  chmod 600 /etc/ssl/private/vsftpd.key

  chown root:root /etc/ssl/private/vsftpd.key

❖ **Comunicación HTTP sin cifrar en Apache**

- **ESTADO:** ABIERTO
- **RIESGO:** ALTO

ALCANCE:

El servicio **HTTP sin cifrar** en Apache implica que toda la comunicación entre el cliente (navegador) y el servidor se realiza en **texto plano**, sin usar protocolos seguros como **SSL/TLS** (HTTPS). Esto implica:

- **Exposición de información crítica:** Credenciales en login de usuario, robo de la cookie de sesión o inyección de código malicioso.
- **Riesgos reputacionales:** Pérdida de confianza (marcados como no seguros), y Blacklisting por motores de búsqueda reconocidos.
- **PCI-DSS (Req. 8.2.3):** Exige autenticación multifactor (MFA) para acceso remoto.
- **GDPR/HIPAA:** Una brecha por contraseñas débiles puede considerarse negligencia**.**

MECANISMOS DE EXPLOTACIÓN:

- **Sniffing**: captura de paquetes de datos en redes públicas (Wi-Fi, ISPs).
- **Man-in-the-Middle**: Interceptación activa mediante **ARP spoofing** o DNS hijacking.
- **Session Hijacking**: Robo de cookies no cifradas para suplantar usuarios.

ACCIÓN RECOMENDADA:

- Obtener un certificado SSL/TLS

bash

sudo apt install certbot python3-certbot-apache

sudo certbot --apache -d tudominio.com

- Configurar Apache para usar HTTPS

bash

sudo nano /etc/apache2/sites-available/tudominio-ssl.conf

*<VirtualHost *:443>*

ServerName tudominio.com

SSLEngine on

SSLCertificateFile /etc/ssl/certs/tudominio.pem

SSLCertificateKeyFile /etc/ssl/private/tudominio.key

Redirect permanent / https://tudominio.com/

</VirtualHost>

Redireccionar obligatoriamente a HTTPS

sudo nano/etc/apache2/sites-available/000-default.conf

*<VirtualHost *:80>*

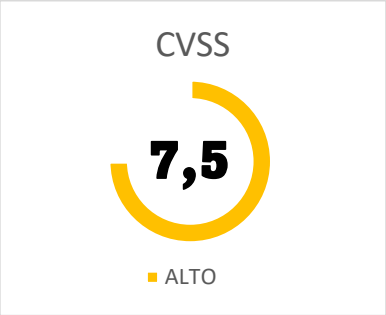
ServerName tudominio.com

Redirect permanent / https://tudominio.com/

</VirtualHost>

❖ **Directiva AllowOverride All en Apache**

- **ESTADO:** ABIERTO
- **RIESGO:** ALTO



ALCANCE:

En el VirtualHost principal (/var/www/html) se permite AllowOverride All, lo que autoriza a archivos .htaccess sobrescribir configuraciones globales. Un atacante que logre escribir en el directorio web podría crear un .htaccess malicioso (cambiar reglas de acceso, habilitar ejecución de scripts) y eludir controles globales. Esto impacta en las siguientes áreas:

- **Confidencialidad:** Permite exponer archivos sensibles (.env, config.php) mediante directivas como Indexes o FollowSymLinks.
- **Integridad:** Posibilita la ejecución de código arbitrario (ej. usando AddHandler para tratar archivos como scripts PHP).
- **Disponibilidad:** Podría causar DoS mediante redirecciones masivas o reglas maliciosas en .htaccess.
- **PCI-DSS (Req. 6.5):** Configuraciones inseguras como AllowOverride All violan el principio de mínimo privilegio.
- **GDPR (Art. 32):** Exposición de datos personales por archivos expuestos podría generar multas de hasta €20 millones.

MECANISMOS DE EXPLOTACIÓN:

- **Expopsición**: Habilitar listado de directorios o seguir enlaces simbólicos.
- **Ejecución de Código Arbitrario**: Usar AddHandler para tratar archivos estáticos (.txt, .log) como scripts ejecutables (PHP, Python, etc.).
- **Redirección Maliciosa**: Usar Redirect o para robar credenciales o distribuir malware.

ACCIÓN RECOMENDADA:

- Configurar AllowOverride None:

  bash

  sudo nano /etc/apache2/apache2.conf

  AllowOverride None

- Restringir permisos de escritura:

  bash

  chown -R root:www-data /var/www/html

  chmod -R 750 /var/www/html

- Auditar archivos .htaccess existentes:

  bash

  find /var/www/ -name .htaccess -exec grep -HnE

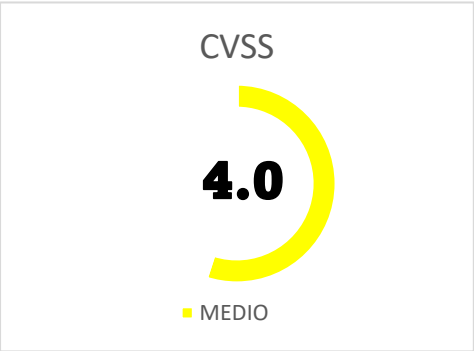
  "AddHandler|Redirect|Options" {} \;

- Habilitar mod_security:

  bash

  sudo a2enmod security2

❖ **Password field with autocomplete enabled**



- **ESTADO:** ABIERTO
- **RIESGO:** MEDIO

ALCANCE:

La mayoría de los navegadores tienen la posibilidad de recordar las credenciales de usuario que se ingresan en formularios HTML. Esta función puede ser configurada por el usuario y también por aplicaciones que emplean credenciales de usuario. Si la función está habilitada, las credenciales ingresadas por el usuario se almacenan en su computadora local y el navegador las recupera en futuras visitas a la misma aplicación. Las credenciales almacenadas pueden ser capturadas por un atacante que obtiene el control sobre la computadora del usuario. Además, un atacante que encuentre una vulnerabilidad de aplicación separada, como secuencias de comandos en sitios cruzados, puede explotar esto para recuperar las credenciales almacenadas en el navegador de un usuario.

ACCIÓN RECOMENDADA:

Para evitar que los navegadores almacenen las credenciales ingresadas en formularios HTML, incluya el **atributo autocompletar="off"** dentro de la etiqueta FORM (para proteger todos los campos de formulario) o dentro de las etiquetas INPUT relevantes (para proteger campos individuales específicos).
Tenga en cuenta que los navegadores web modernos pueden ignorar esta directiva. A pesar de esto, existe la posibilidad de que no deshabilitar el autocompletado pueda causar problemas para obtener el cumplimiento de PCI.

❖ **Robots.txt file**

- **ESTADO:** ABIERTO
- **RIESGO:** INFORMACIÓN

ALCANCE:

El archivo robots.txt se utiliza para dar instrucciones a los robots web, como los rastreadores de motores de búsqueda, sobre las ubicaciones dentro del sitio web que los robots pueden, o no, rastrear e indexar.
La presencia de robots.txt no presenta en sí misma ningún tipo de vulnerabilidad de seguridad. Sin embargo, a menudo se usa para identificar áreas restringidas o privadas de los contenidos de un sitio. Por lo tanto, la información en el archivo puede ayudar a un atacante a mapear el contenido del sitio, especialmente si algunas de las ubicaciones identificadas no están vinculadas desde otra parte del sitio. Si la aplicación se basa en robots.txt para proteger el acceso a estas áreas y no impone un control de acceso adecuado sobre ellas, esto presenta una vulnerabilidad grave.

ACCIÓN RECOMENDADA:

El archivo robots.txt no es en sí mismo una amenaza para la seguridad, y su uso correcto puede representar una buena práctica por razones que no son de seguridad. No debe asumir que todos los robots web cumplirán con las instrucciones del archivo. Más bien, suponga que los atacantes prestarán mucha atención a cualquier ubicación identificada en el archivo. No confíe en robots.txt para proporcionar ningún tipo de protección sobre el acceso no autorizado.

❖ **Directory listing**

- **ESTADO:** ABIERTO
- **RIESGO:** INFORMACIÓN

ALCANCE:

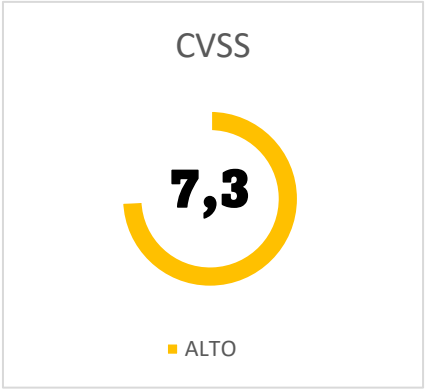
Los servidores web se pueden configurar para enumerar automáticamente el contenido de los directorios que no tienen una página de índice presente. Esto puede ayudar a un atacante al permitirles identificar rápidamente los recursos en una ruta determinada y proceder directamente a analizar y atacar esos recursos. En particular, aumenta la exposición de archivos confidenciales dentro del directorio que no están destinados a ser accesibles para los usuarios, como archivos temporales y volcados.
Los listados de directorios en sí mismos no constituyen necesariamente una vulnerabilidad de seguridad. Cualquier recurso sensible dentro de la raíz web debe, en cualquier caso, estar adecuadamente controlado por el acceso, y no debe ser accesible por una parte no autorizada que conozca o adivine la URL. Incluso cuando los listados de directorios están deshabilitados, un atacante puede adivinar la ubicación de los archivos confidenciales utilizando herramientas automatizadas.

ACCIÓN RECOMENDADA:

Por lo general, no hay una buena razón para proporcionar listados de directorios, y deshabilitarlos puede colocar obstáculos adicionales en la ruta de un atacante. Esto normalmente se puede lograr de dos maneras:

- ☑ Configure su servidor web para evitar listados de directorios para todas las rutas debajo de la raíz web;
- ☑ Coloque en cada directorio un archivo predeterminado (como index.htm) que mostrará el servidor web en lugar de devolver una lista de directorios.

❖ **Vulnerabilidades en Archivos de Configuración**



- **ESTADO:** ABIERTO
- **RIESGO:** ALTA

ALCANCE:

La vulnerabilidad relacionada con el archivo wp-config.php de WordPress es especialmente crítica cuando se utiliza PHP mediante CGI, ya que permite a los atacantes acceder a información sensible como el código fuente de la plataforma y los datos de conexión a la base de datos. Esta no es una vulnerabilidad de WordPress en sí, sino que se produce debido a la falta de controles en el pasaje de parámetros en la configuración CGI de PHP

ACCIÓN RECOMENDADA:

Para mitigar esta vulnerabilidad, es importante actualizar y mantener seguras las configuraciones de PHP y CGI, así como mantener actualizado el software de WordPress y sus plugins. Además, es recomendable seguir las mejores prácticas de seguridad de WordPress, como ocultar la versión de WordPress, desactivar los errores de PHP y restringir el acceso a archivos y directorios sensibles.

6. Conclusiones Finales

Tras las pruebas de seguridad realizas, se han detectado un total de 17 vulnerabilidades, que afectan a los protocolos FTP y SSH, clasificadas en 4 críticas, 6 altas, 6 medias y 1 de categoría baja. Las vulnerabilidades críticas afectan principalmente al servicio OpenSSH, el cual permite realizar comunicaciones cifradas a través de una red, usando el protocolo SSH.

El impacto potencial de estás vulnerabilidades detectadas, podrían permitir a un intruso sin acceso comprometer la integridad y disponibilidad del sistema, mediante la ejecución de comandos o interrupción de servicios críticos para la empresa. Esto supondría un peligro para la confidencialidad de la información sensible del que pudiéramos almacenar en el sistema.

Actualmente existe una exposición elevada, pero si se prioriza adecuadamente, podría solucionarse en un corto plazo, de 1 a 3 semanas, dedicando el personal y recursos necesarios.

Se recomienda enfocarse de manera prioritaria:

- Actualización inmediata del sistema operativo, instalando los parches de seguridad necesarios.
- Configuración de los servicios SSH y FTP, priorizando en la seguridad de los mismos.
- Establecer una política de contraseñas robusta.

De manera menos urgente, se sugiere establecer procesos internos para la gestión de vulnerabilidades, formar en la medida que se posible al personal interno en Ciberseguridad y realizar auditorías de manera interna de forma regular.

Para finalizar, indicar que, siguiendo las medidas indicadas, la empresa tiene una gran oportunidad para mejorar su postura de seguridad de manera significativa.

7. Anexos

7.1 Tabla explicativa de vulnerabilidades detectadas.

Se adjunta tabla de vulnerabilidades encontradas para una mejor visualización de las mismas.

Servicio	Puerto	ID de Vulnerabilidad	Criticidad (CVSS)	Clasificación de Riesgo	Estado	Descripción Breve
FTP	21	CVE-2021-30047	7.5	Alta	Abierta	Ejecución remota de comandos a través de desbordamiento.
FTP	21	CVE-2021-3618	7.4	Alta	Abierta	Vulnerabilidad que permite bypass de autenticación.
SSH	22	CVE-2023-38408	9.8	Crítica	Abierta	Ejecución remota de código en OpenSSH por manejo incorrecto de objetos.
SSH	22	CVE-2023-28531	9.8	Crítica	Abierta	Escalación de privilegios en servidores OpenSSH vulnerables.
SSH	22	CVE-2024-6387	8.1	Alta	Abierta	Condición de carrera permitiendo posibles ataques DoS o RCE.
SSH	22	CVE-2025-26465	6.8	Media	Abierta	Posibilidad de acceso no autorizado debido a falta de validación de sesión.
SSH	22	CVE-2023-51385	6.5	Media	Abierta	Riesgo de fuga de información sensible a través de canal SSH.
SSH	22	CVE-2023-48795	5.9	Media	Abierta	Vulnerabilidad menor de exposición de metadatos.
SSH	22	CVE-2023-51384	5.5	Media	Abierta	Vulnerabilidad que podría permitir ataque de fuerza bruta eficiente.
SSH	22	2C119FFA-ECE0-5E14-A4A4-354A2C38071A	10	Crítica	Abierta	Vulnerabilidad crítica detectada en servicio SSH (requiere validación).

SSH	22	B8190CDB-3EB9-5631-9828-8064A1575B23	9.8	Crítica	Abierta	Riesgo de ejecución de código no autenticado en SSH.
SSH	22	F8981437-1287-5B69-93F1-657DFB1DCE59	8.1	Alta	Abierta	Error de validación de certificados en servidor SSH.
SSH	22	1337DAY-ID-39674	8.1	Alta	Abierta	Vulnerabilidad de escalada de privilegios usando exploits conocidos.
SSH	22	PACKETSTORM:173661	7.5	Alta	Abierta	Exposición de configuraciones inseguras en demonio SSH.
SSH	22	F79E574D-30C8-5C52-A801-66FFA0610BAA	6.8	Media	Abierta	Riesgo de acceso indebido por mala gestión de llaves SSH.
SSH	22	54E1BB01-2C69-5AFD-A23D-9783C9D9FC4C	5.9	Media	Abierta	Vulnerabilidad en proceso de autenticación secundaria.
SSH	22	PACKETSTORM:140261	0	Informativa	Abierta	Vulnerabilidad sin criticidad asignada; se requiere evaluación manual.