2025

Informe Forense

ANALISTAS DE CIBERSEGURIDAD

DARÍO CAIANI TARAZONA

PABLO GARCÍA TORRES

EMPRESA: **SecurityCat**

SecurityCat

Tabla de contenido

1. Objetivo y Alcance

1.1 *Objetivo*

El objetivo del presente análisis forense es la investigación de un posible compromiso de la seguridad del servidor, bloquear posibles exploit, identificar vulnerabilidades en los diferentes servicios, recopilar evidencias, en el caso de que las hubiera y proponer medidas correctivas y preventivas para la seguridad del sistema.

1.2 *Alcance*

El análisis que se va a realizar al servidor Debian va a ser un análisis completo y exhaustivo, abarcando desde el registro de logs, servicios activos, configuraciones inseguras, archivos maliciosos y usuarios del sistema.
Para enfrentarse al incidente de seguridad, como analista tengo claro las acciones que se deben realizar, siendo meticuloso y detallado, procurando ser lo menos intrusivo posible con el fin de preservar el sistema en su estado original.
Finalmente tengo presente los requisitos y pautas a la hora de realizar el presente informe forense a la hora de no incumplir ningún proceso legal ni quebrantar la legislación actual.

2. Entorno Analizado

El sistema afectado es un servidor crítico que ha sido comprometido en la empresa **4GEEKS**. Versión del sistema:

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

3. Metodología y Herramientas Utilizadas

3.1 *Adquisición de evidencias*

Se realiza una imagen forense, mediante la clonación de la máquina virtual a realizar el análisis para la preservación de evidencias digitales en su estado original sin alteraciones. Dicho proceso es esencial para la recuperación de datos, la identificación de pruebas para posibles certificaciones de información en procedimientos judiciales.
Se realiza extracción de logs para su análisis, ya que estos registros nos van a proporcionar información valiosa sobre todos los eventos ocurridos en el sistema.

3.2 *Análisis de Logs*

Se procedió a la extracción y revisión completa de los registros del sistema y del servicio Apache2. Tras un análisis detallado, no se identificaron indicios de ataque ni evidencias de compromiso en el sistema. No obstante, la información recopilada revela la existencia de múltiples configuraciones deficientes que representan un riesgo significativo para la seguridad del entorno.
Con el objetivo de compilar la información de forma más precisa, se llevó a cabo una extracción segmentada de los registros por servicio. Esta metodología permitió realizar una comparación cronológica de los eventos, facilitando una identificación más clara y exacta de las acciones ejecutadas desde la terminal.
Para realizar dicha segmentación de los registros de logs por servicio se utilizó la herramienta **Notepad++,** y para obtener los extractos de líneas se utilizó la herramienta **Windsurf**.

3.3 *Análisis de servicios y configuraciones*

Se verifican los servicios activos en el sistema y los archivos que ejecutan las tareas programadas.

- ☑ Servicio FTP activo y corriendo por el puerto 21.
- ☑ Servicio SSH activo y corriendo por el puerto 22.
- ☑ Servicio WEB (http) activo y corriendo por el puerto 80.
- ☑ Servicio de Impresión (IPP) activo y corriendo por el puerto 631.

- ☑ Base de datos instalada y activa, escuchando por el puerto 3306.
- ☑ Verificación de servicios Cron y Anacron.

3.4 *Búsqueda de archivos sospechosos*

Se realiza una exhaustiva búsqueda en ubicaciones comunes utilizadas por malware.

- ☑ Búsqueda de binarios ocultos en **/usr/bin/, /usr/local/bin/, /opt/, /home/[usuario]/.config/**.

- ☑ Revisión de directorios temporales en busca de scripts o payloads, **/tmp/, /var/tmp/, /dev/shm/**.

- ☑ Archivos o carpetas con nombres que comienzan por punto **(.script, .hidden, .update, etc.)**.

3.5 *Detección de malware*

Para la detección de posibles malware, como pueden ser rootkits, troyanos, etc., se ha utilizado herramientas especializadas.

- ☑ **Rkhunter**: es una herramienta que detecta los rootkits, puertas traseras y los exploits locales mediante la comparación de los resúmenes MD5 de los ficheros importantes con su firma correcta en una base de datos en línea.
- ☑ **Lynis:** herramienta de auditoría de seguridad, realizando escaneos profundos y detectando fallos de seguridad.
- ☑ **ClamAV:** es un software antivirus de código abierto, identifica y bloquea posibles virus del sistema.

3.6 *Análisis de procesos activos*

Se realiza administración y monitoreo de procesos tanto de forma estática como en tiempo real de los procesos en ejecución del sistema. Se utilizan herramientas de línea de comandos.

- ☑ Ps
- ☑ top
- ☑ Htop

3.7 *Análisis de usuarios y privilegios*

Se realiza revisión de los diferentes ficheros con información relevante sobre los usuarios y sus privilegios.

- ☑ **/etc/passwd** → información de usuarios.
- ☑ **/etc/group** → información de grupos de usuarios.
- ☑ **/etc/shadow** → información de contraseñas cifradas.
- ☑ **/etc/sudoers** → información de privilegios de usuarios y grupos para ejecutar comandos de superusuario.
- ☑ También se realiza un análisis al archivo **.bash_history**, el cual almacena el historial de comandos ejecutados en la terminal Bash.

3.8 *Detección de persistencia maliciosa*

Se realiza revisión del archivo **crontab**, el cual contiene las tareas programadas que se ejecutan automáticamente en el sistema.

- ☑ crontab -l
- ☑ cat /etc/crontab
- ☑ ls -la /etc/cron.*/*

También se realiza búsqueda de scripts de inicio, los cuales son una parte crítica para los atacantes, pues mediante estos suelen configurar persistencia. Se realiza revisión de diferentes archivos de configuración, entre los que cabe destacar.

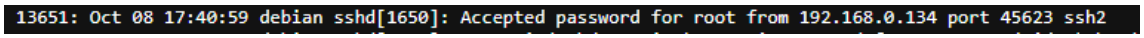
- ☑ ~/bashrc
- ☑ ~/bash_profile / ~/.profile
- ☑ ~/.xinitrc
- ☑ Revisión de ficheros con las extensiones, .sh, .bash, .py, .php, .js, .pl.

4. Evidencias y Hallazgos

Tras un exhaustivo estudio del servidor no se puede llegar a la conclusión de un acceso no autorizado al sistema. No se identifican evidencias claras de intrusión, pero si se recomiendan implementar configuraciones seguras en diferentes servicios clave para el buen funcionamiento del sistema.

4.1 *Indicadores de Compromiso (IOCs)*

Cabe destacar como información relevante el acceso mediante el protocolo **sshd**, el día 8 de octubre a las 17:40:59 horas, de la IP 192.168.0.134 por el puerto 45623 con usuario “root”. Se muestra imagen con el log, recogido en el fichero “journal” en la ruta /var/log/journal/.



5. Fallos de seguridad identificados

Se genera una tabla con los fallos de seguridad encontrados, indicando la categoría del fallo y una breve descripción del mismo.

Categoría	Descripción
Base de Datos: mysql	Usuarios con exceso de privilegios Uso de SSL desactivado Contraseña de acceso débil
Servicio SSH	Configuración insegura
Servicio FTP	Configuración insegura
Servicio Web (HTTP)	Configurar el servicio para el uso de HTTPS.
Usuario Debian	Posee privilegios completos de “sudo”
Usuarios Debian / root	Contraseñas débiles
Sistema NO actualizado	Falta de actualizaciones del sistema
Sistema sin Firewalls	Instalar Firewall
Permisos elevados	Directorios con permisos elevados

6. Recomendaciones y Mitigaciones

En el siguiente apartado vamos a indicar una serie de recomendaciones en las configuraciones de servicios y del sistema, pues al realizar el análisis hemos detectado diferentes debilidades en las configuraciones que pueden representar un peligro para la integridad del sistema.

6.1 *Actualizaciones del sistema*

Configuración errónea detectada:

El sistema presenta paquetes desactualizados y no cuenta con actualizaciones automáticas disponibles. No tener el sistema correctamente actualizado puede ser un gran peligro para la seguridad del sistema.

Configuración recomendada:

Actualizar el sistema ejecutando el comando:

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

Habilitar actualizaciones de seguridad automáticas, mediante **unattended-upgrades.** Para habilitarlo ejecutamos el comando:

sudo dpkg-reconfigure unattended-upgrade

También se puede utilizar la herramienta “cron”, la cual nos permite programar tareas para que se ejecuten automáticamente en horarios específicos, sin necesidad de intervención manual. Por ejemplo, podemos añadir en el archivo “crontab”, el cual contiene las tareas programadas la siguiente línea:

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

Añadiendo esta línea, actualizaremos paquetes cada lunes a medianoche, este proceso lo realizará automáticamente sin intervención manual

6.2 *Servicio vsFTPd. Versión 3.0.3-13+b2*

Para modificar el fichero de configuración del servicio vsFTPd ejecuten el comando:

sudo nano /etc/vsftpd.conf

Configuración errónea detectada:

- Listen = NO → vsftpd escucha en modo standalone (no mediante inetd/xinetd)
- listen_ipv6 = YES → Desactiva IPv6 (si no lo necesitas)
- Anonymous_enable = YES → Deshabilita el acceso anónimo
- ssl_enable = NO → Habilita SSL/TLS
- #chroot_local_user = YES → Encierra a usuarios en su directorio home (jail)
- Connect_from_port_20 = YES → Uso de conexiones por el puerto 20 para el modo activo.

Configuración recomendada:

- Listen = YES → vsftpd escucha en modo standalone (no mediante inetd/xinetd)
- listen_ipv6 = NO → Desactiva IPv6 (si no se necesita)
- Anonymous_enable = NO → Deshabilita el acceso anónimo
- ssl_enable = YES → Habilita SSL/TLS
- chroot_local_user = YES → Encierra a usuarios en su directorio home (jail). Habilitar esta opción quitando (#)
- Connect_from_port_20 = NO → Uso de conexiones por el puerto 20 para el modo activo. Desactivar modo activo si no es necesario.

6.3 *Servicio SSHD (servidor SSH). Versión 1:9.2p1-2+deb12u3*

Para llegar al fichero de configuración del servidor SSH, ejecutamos el comando:

sudo nano /etc/ssh/sshd_config

Configuración errónea detectada:

- PermitRootLogin = YES → No permitir inicio de sesión como usuario “Root”.
- PasswordAuthentication = YES → Autenticación por contraseña (Si se usa claves SSH desactivar).
- #PubkeyAuthentication = YES → Habilitar uso de claves SSH
- X11Forwarding = YES → Evitar riesgos deshabilitando reenvío X11
- #MaxAuthTries = 6 → Habilitar número máximo de intentos, por ejemplo, a 3.
- #PermitEmptyPasswords = NO → Habilitar no permitir usuarios vacíos.

Configuración recomendada:

- PermitRootLogin = NO → No permitir inicio de sesión como usuario “Root”.
- PasswordAuthentication = NO → Autenticación por contraseña (Si se usa claves SSH desactivar).
- PubkeyAuthentication = YES → Habilitar uso de claves SSH. Habilitar quitando (#).
- X11Forwarding = NO → Evitar riesgos deshabilitando reenvío X11
- MaxAuthTries = 3 → Habilitar número máximo de intentos. Habilitar quitando (#).
- PermitEmptyPasswords = NO → Habilitar no permitir usuarios vacíos. Habilitar quitando (#).

Una ver que se realizan los cambios, realizar un reinicio del servicio, ejecutando el comando:

sudo systemctl restart sshd

6.4 *Servicio Web (Apache)*

Para llegar al fichero de configuración del servidor Apache, ejecutamos el comando:

sudo nano /etc/apache2/apache2.conf

Configuración errónea detectada:

☑ Fichero apache2.conf ubicado en la ruta /etc/apache2

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

  ➢ Require all granted → Acceso total al directorio raíz, criticidad máxima

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

  ➢ Require all granted → Acceso total al directorio share, criticidad elevada

☑ Archivo 000-default.conf en /etc/apache2/sites-enabled

Para llegar al fichero de configuración de puertos de Apache, ejecutamos el comando:

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

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

  ➢ HTTP → El tráfico de datos no está cifrado, de ser interceptado la información está desprotegida.
  ➢ ServerName → No existe serverName definido, apache recibirá peticiones de dominios no contemplados y podrá mostrar información no deseada.

- AllowOverride All → Permite que cualquier ".htacces" pueda modificar la configuración, no es seguro en un entorno de http.

☑ Permisos elevados en el directorio /var/www/html

- permisos 777 en /var/www/html/wp-config.php → permisos de root para cualquier usuario en archivo sensible.
- permisos 777 en /var/www/html → permisos de root para cualquier usuario en todo el directorio.

Configuración recomendada:

sudo nano /etc/apache2/apache2.conf

☑ Fichero apache2.conf ubicado en la ruta /etc/apache2

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

  - Require all denied → Evita que usuarios del servicio apache puedan acceder a la carpeta raíz.

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

  - Require all denied → Evita que otros usuarios del servicio tengan acceso a la carpeta.

☑ Crear claves privadas SSH con OpenSSL

- **sudo mkdir -p /etc/ssl/private/2x2geeks.com**
- **sudo chmod 700 /etc/ssl/private/2x2geeks.com**

☑ Crear certificados públicos SSH con OpenSSL

- **sudo mkdir -p /etc/ssl/certs/2x2geeks.com**
- **sudo chmod 755 /etc/ssl/certs/2x2geeks.com**

☑ Generar una clave privada

- **cd /etc/ssl/private/2x2geeks.com**
- **sudo openssl genrsa -out 2x2geeks.com.key 4096**

☑ Ajustar los permisos del certificado público con permisos de lectura para root y www-data

- **sudo chown root:www-data /etc/ssl/certs/2x2geeks.com/*.crt **
- **sudo chmod 640 /etc/ssl/certs/2x2geeks.com/*.crt **

☑ configurar el archivo default-ssl.conf en /etc/apache2/sites-available/default-ssl.conf añadiendo las lineas que corresponden al ServerName y las rutas de los certificados y llaves SSL

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

- <VirtualHost *:443>
  ServerName www. 2x2geeks.com
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/html
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
  SSLEngine on
  SSLCertificateFile /etc/ssl/certs/2x2geeks.com.crt
  SSLCertificateKeyFile /etc/ssl/private/2x2geeks.com.key
  <FilesMatch "\.(?:cgi|shtml|phtml|php)$">

```
        SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>
</VirtualHost>
```

- HTTPS → El tráfico de datos está cifrado, de ser interceptado la información está protegida.
- ServerName (www.2x2geeks.com)→ Colocar un ServerName evitará fallos en el servicio y permite la ampliación a más dominios en el futuro
- SSL access → Las contraseñas seguras y encriptadas son clave para la seguridad de los activos.

☑ Inactivar protocolo HTTP en el archivo 000-default.conf en /etc/apache2/sites-enabled/000-default.conf

- **sudo a2dissite 000-default.conf**

☑ Activar protocolo HTTPS en el archivo default-ssl.conf en /etc/apache2/sites-available/default-ssl.conf

- **sudo a2ensite default-ssl.conf**

☑ Recargar los cambios en Apache

- **sudo systemctl reload apache2**

☑ Cambios de permisos en directorios /var/www/html para usuario www-data

- Registra como dueño del directorio html al usuario y grupo www-data
  - **sudo chown -R www-data:www-data /var/www/html/**

- Permisos completos a www-data y permisos de solo lectura y ejecución a resto de usuarios
  - **sudo chmod -R 755 /var/www/html/**

☑ Cambios de permisos en archivo /var/www/html/wp-config.php

- Permisos de solo lectura y escritura a www-data y permisos nulos al resto de usuarios

  ➢ **sudo chown -R 600 /var/www/html/wp-config.php**

6.5 *Base de Datos (MySQL). Versión 15.1 Distrib 10.11.6-MariaDB*

Existen diferentes debilidades en la configuración de la base de datos, se propone estos cambios.

Configuración errónea detectada:

☑ Fichero debian.cnf ubicado en la ruta /etc/mysql
- User = root → No utilizar el usuario root (client).
- User = root → No utilizar el usuario root (mysql_upgrade)
- Password → Habilitar contraseñas.

☑ Fichero 50-client.cnf ubicado en la ruta /etc/mysql/mariadb.conf.d
- #ssl-cert = /etc/mysql/client-cert.pem → Certificado para conexiones encriptadas.
- #ssl-key = /etc/mysql/client-key.pem → Certificado para conexiones encriptadas.
- #ssl-verify-server-cert = on → Activación verificación del certificado

☑ Usuarios de la base de datos.
- Usuario “wordpressuser” → exceso de privilegios (GRANT ALL PRIVILEGES), revocar privilegios para una mayor seguridad.
- Usuario “user” → exceso de privilegios (ALL PRIVILEGES ON y WITH GRANT OPTION), tiene privilegios equivalentes a root, riesgo muy alto, para mayor seguridad revocar privilegios.
- Los usuarios → ”root”, ”mysql” y “mariadb.sys” también cuenta con privilegios elevados, aunque pudiera ser normal para estos usuarios, se recomienda que no todos ejerzan de superusuarios.

Configuración recomendada:

☑ Fichero debian.cnf ubicado en la ruta /etc/mysql

- User = debian-sys-maint → No utilizar el usuario root (client).
- User = debian-sys-maint → No utilizar el usuario root (mysql_upgrade)
- Password → Habilitar contraseñas.

☑ Fichero 50-client.cnf ubicado en la ruta /etc/mysql/mariadb.conf.d

- ssl-cert = /etc/mysql/client-cert.pem → Certificado para conexiones encriptadas. Habilitar quitando (#).
- ssl-key = /etc/mysql/client-key.pem → Certificado para conexiones encriptadas. Habilitar quitando (#).
- ssl-verify-server-cert = on → Activación verificación del certificado. Habilitar quitando (#).

☑ Usuarios de la base de datos.

- Se recomienda una revisión de privilegios por parte del DBA (administrador de la base de datos), otorgando el principio de menor privilegio.

6.6 *Firewalls*

Se recomienda el uso de firewalls para controlar y proteger el tráfico de la red. Ya que un firewall es una barrera de seguridad esencial.

Configuración errónea detectada:

☑ Se verifica la falta de uso de firewalls.

Configuración recomendada:

☑ Se recomienda el uso de firewalls, ya se implementando reglas “iptables” o mediante la instalación de “UFW”. Para la instalación de “UFW”, ejecute el siguiente comando.

```
sudo apt install ufw
```

Una vez instalado, asignar las reglas necesarias para mantener una red segura y el correcto funcionamiento de la misma.

6.7 *Acciones adicionales.*

- Forzar al uso de contraseñas complejas para los usuarios.
- Realizar auditorías aleatorias de permisos y servicios.
- Realizar backups cifrados de la información.
- Realizar periódicamente escaneo y análisis de seguridad.
- Desactivar puertos y servicios que no se estén utilizando.

7. Conclusión

Tras el análisis forense realizado, podemos confirmar que no se han encontrado evidencias concluyentes de acceso no autorizado al servidor, intrusión o actividad maliciosa en el sistema. Después de una revisión exhaustiva a los diferentes vectores de ataque, como puede ser, logs de autenticación, scripts de inicio, procesos en ejecución, etc., se puede verificar que no existen indicios de compromiso ni presencia de malware.

Es cierto, que durante el proceso de análisis se identificaron diferentes debilidades en la configuración del sistema y servicios, aunque en este caso no han sido explotadas, representan un alto peligro en términos de seguridad y podrían facilitar intrusiones en el sistema en un futuro. Como se indica en el presente informe, estas debilidades incluyen servicios expuestos, cuentas de usuario con contraseñas débiles, ausencia de mecanismos de monitoreo o detección de intrusos entre otras.

En el apartado de “Mitigaciones”, se detallan una serie de recomendaciones correctivas y buenas prácticas enfocadas en fortalecer la seguridad del sistema, garantizar un entorno más seguro y robusto ante posibles amenazas y minimizar la superficie de exposición.

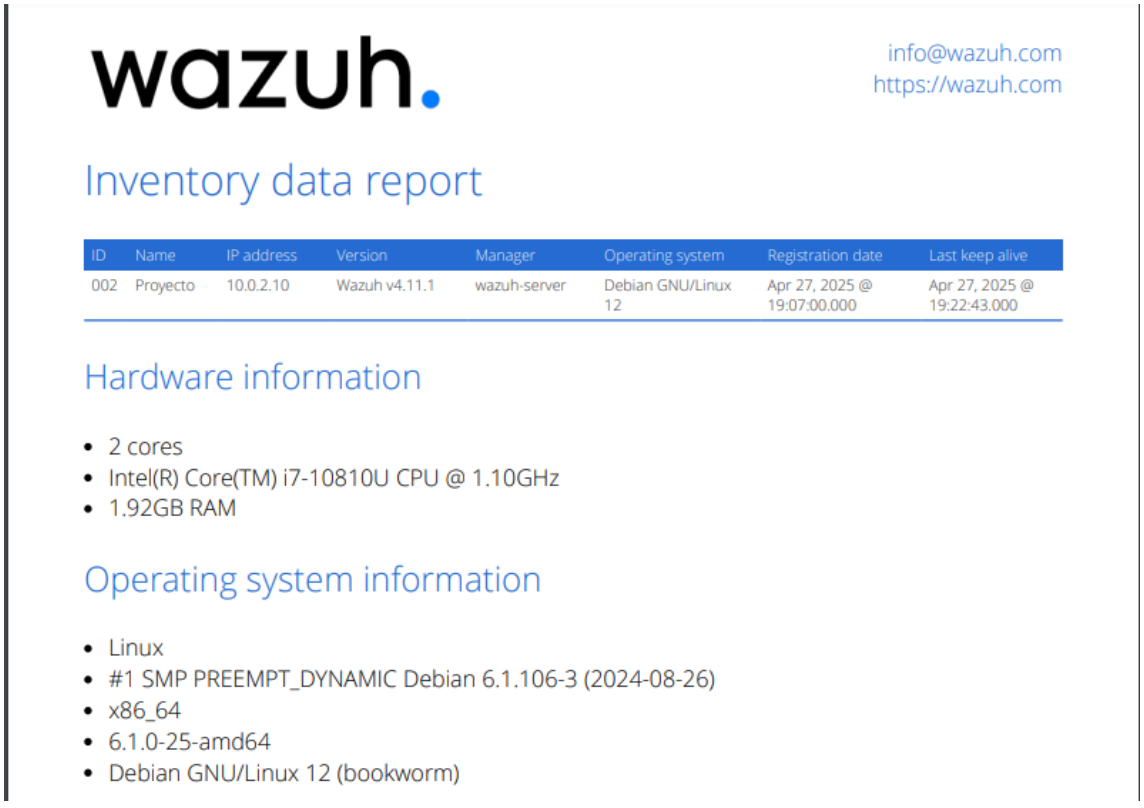
Para finalizar, aunque el servidor no ha sido comprometido, es fundamental abordar las configuraciones inseguras detectadas para evitar futuros incidentes de seguridad.

8. Anexo

8.1 *Wazuh*

Se utiliza la plataforma Wazuh para generar un informe acerca del estado del servidor Debian. Mediante esta plataforma se realiza monitoreo del servidor y en una interfaz gráfica se visualizan alertas, se generan informes y se realizan auditorias de cumplimiento.
Se adjunta enlace para la visualización de dicho informe.

Informe Wazuh



wazuh.

info@wazuh.com
https://wazuh.com

Inventory data report

ID	Name	IP address	Version	Manager	Operating system	Registration date	Last keep alive
002	Proyecto	10.0.2.10	Wazuh v4.11.1	wazuh-server	Debian GNU/Linux 12	Apr 27, 2025 @ 19:07:00.000	Apr 27, 2025 @ 19:22:43.000

Hardware information

- 2 cores
- Intel(R) Core(TM) i7-10810U CPU @ 1.10GHz
- 1.92GB RAM

Operating system information

- Linux
- #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26)
- x86_64
- 6.1.0-25-amd64
- Debian GNU/Linux 12 (bookworm)