

# Comprobación de Seguridad en WordPress con Wazuh

## Introducción

En esta guía, aprenderás a configurar y utilizar Wazuh para comprobar la seguridad de una instalación de WordPress. Wazuh es una plataforma de seguridad de código abierto que permite realizar comprobaciones de la configuración de seguridad (SCA, Security Configuration Assessment) con el objetivo de identificar vulnerabilidades y configuraciones incorrectas en tu sitio de WordPress.

El uso de SCA en Wazuh facilita la detección de configuraciones débiles, permisos inadecuados, credenciales expuestas y otras fallas de seguridad que pueden comprometer un sitio web. A lo largo de esta guía, se explicarán los pasos necesarios para la instalación y configuración de Wazuh, la implementación del agente en el servidor donde se ejecuta WordPress y la correcta integración con WP-CLI, herramienta clave para gestionar WordPress desde la línea de comandos.

## Requisitos Previos

Antes de comenzar, asegúrate de contar con los siguientes requisitos:

1. Un servidor con WordPress instalado.
2. Acceso *root* o permisos de administrador en el servidor.
3. Docker instalado (si optas por la versión en contenedores de Wazuh).
4. WP-CLI instalado en el servidor (herramienta de línea de comandos para administrar WordPress).

## Instalación de Wazuh

### Opción 1: Instalación en Docker

Si prefieres instalar Wazuh utilizando Docker, sigue estos pasos:

- Clonar el repositorio de Wazuh Docker:

```
git clone <https://github.com/wazuh/wazuh-docker.git> -b v4.9.2
cd wazuh-docker/single-node
```

- Generar los certificados para el indexador:

```
sudo docker compose -f generate-indexer-certs.yml run --rm generator
```

- Iniciar los contenedores de Wazuh:

```
sudo docker compose up -d
```

- Acceder a la interfaz de Wazuh en el navegador:

https://localhost #credenciales por defecto admin: SecretPassword

The screenshot displays the Wazuh dashboard interface. At the top, there's a navigation bar with links to Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. The main content area is divided into several sections:

- Overview:** Shows a summary of alerts. Under "LAST 24 HOURS ALERTS", there are four categories: Critical severity (0), High severity (0), Medium severity (46), and Low severity (1). Each category includes a description of the rule level range.
- Endpoint Security:** Includes modules like Configuration Assessment, File Integrity Monitoring, Malware Detection, and Threat Hunting.
- Threat Intelligence:** Includes modules like MITRE ATT&CK and VirusTotal.
- Security Operations:** Includes modules like PCI DSS, GDPR, HIPAA, and NIST 800-53.
- Cloud Security:** Includes modules like Docker, Amazon Web Services, Google Cloud, and GitHub.

A "HELP" dropdown menu is visible in the top right corner, showing links to Documentation, Slack Channel, Projects on GitHub, and Google Group.

## Opción 2: Instalación mediante script de la documentación.

Si prefieres instalar Wazuh de manera manual, sigue la guía oficial proporcionada por la AGETIC:

- [Guía de instalación de Wazuh](#)

Debes tomar en cuenta la recomendación de instalar siempre la última versión disponible de Wazuh y comenzar con la instalación con privilegios de root o un usuario sudo.

## Configuración de WP-CLI

Para que Wazuh pueda interactuar con WordPress y realizar la comprobación de seguridad, es necesario instalar **WP-CLI** en el servidor. Sigue estos pasos:

- Descargar e instalar WP-CLI:

```
curl -O <https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar>
php wp-cli.phar --info
chmod +x wp-cli.phar
sudo mv wp-cli.phar /usr/local/bin/wp
```

- Verificar que WP-CLI esté instalado correctamente:

```
wp --info
```

Si la instalación fue exitosa, deberías ver la versión de WP-CLI reflejada en la salida del comando.

# Creación de Políticas de Seguridad para WordPress

## Paso 1: Crear el Archivo de Políticas

La comprobación de seguridad en WordPress se basa en la aplicación de políticas definidas en un archivo YAML. A continuación, se describe cómo crearlas e implementarlas en Wazuh.

- Crear el Archivo de Políticas.

```
sudo su
mkdir /home/local_sca_policies/
touch /home/local_sca_policies/custom_wordpress_policy.yml
nano /home/local_sca_policies/custom_wordpress_policy.yml
```

- Añade el siguiente contenido al archivo para definir las reglas de seguridad que se comprobarán en tu sitio Wordpress:

```
# Security Configuration Assessment
# Hardening policies for WordPress installations

policy:
  id: "wordpress_assessment"
  file: "custom_wordpress_policy.yml"
  name: "Comprobación de configuración de seguridad para instalaciones de WordPress."
  description: "Guía para establecer una configuración segura en instalaciones de WordPress."
  references:
    - https://wordpress.org/support/article/hardening-wordpress/
    - https://wpsecuritychecklist.org/items/

requirements:
  title: "Verificar que el endpoint sea un host de WordPress y que tenga instalada la herramienta wp-cli."
  description: "Requisitos para ejecutar el escaneo de SCA contra la política de configuración de WordPress."
  condition: all
  rules:
    - 'f:$wp_install_dir/wp-config.php'
    - 'c: wp --info --allow-root -> r:^WP\pCLI\sversion\p'

variables:
  $wp_install_dir: /var/www/html # Example: /var/www/html
  $wp_host: http://172.19.0.5:8000/ # Examples: https://example.com
  $wp_user: admin # Example: admin
  $wp_security_plugin: wordfence # Example: wordfence

checks:
  - id: 100000
    title: "Fortalecimiento de WordPress: Asegurar que la versión de WordPress esté actualizada."
    description: "La versión instalada de WordPress debe ser la última versión disponible en https://wordpress.org/download/."
    rationale: "Pueden descubrirse nuevas vulnerabilidades en WordPress. Es importante actualizar
```

a la última versión para evitar que se exploten vulnerabilidades descubiertas en versiones antiguas."

remediation: "Actualizar WordPress a la última versión."

condition: all

rules:

- c:runuser -l \$wp\_user -c "wp core check-update --path=\$wp\_install\_dir" ->

r: WordPress\sis\sat\sthe\slatest\sversion

- id: 100001

title: "Fortalecimiento de WordPress: Asegurar que los permisos del archivo .htaccess estén configurados en 644."

description: "Esta política verifica los permisos del archivo .htaccess en el directorio raíz de la instalación de WordPress."

rationale: "Usuarios no autorizados podrían leer el archivo .htaccess si los permisos no son lo suficientemente estrictos. Además, permisos demasiado restrictivos pueden causar errores al cargar un sitio de WordPress."

remediation: "Establecer los permisos del archivo en 644 ejecutando el comando chmod 644 \$wp\_install\_dir/.htaccess"

condition: all

rules:

- c:stat -c '%a' \$wp\_install\_dir/.htaccess -> r:644

- id: 100002

title: "Fortalecimiento de WordPress: Asegurar que la depuración de WordPress esté desactivada."

description: "Esta política verifica si la depuración de WordPress está habilitada en el archivo wp-config.php."

rationale: "Cuando WP\_DEBUG está habilitado, se muestra información detallada sobre errores en las páginas del sitio web. Esto puede incluir información sobre errores, funciones obsoletas y código vulnerable que puede ser explotado por actores maliciosos."

remediation: "Desactivar la depuración de WordPress estableciendo la variable WP\_DEBUG en wp-config.php en false."

condition: none

rules:

- c:runuser -l \$wp\_user -c "wp config list WP\_DEBUG --path=\$wp\_install\_dir" -> r:true|1

- id: 100003

title: "Fortalecimiento de WordPress: Asegurar que no haya archivos de respaldo (.zip, .back, .bac, .old) en el directorio raíz de la instalación de WordPress."

description: "Esta política verifica si hay archivos de respaldo o comprimidos del sitio web o plugins en el directorio raíz de la instalación de WordPress."

rationale: "Se pueden crear archivos de respaldo de algunos archivos de configuración sensibles, como wp-config.php.bak, antes de editar la configuración en vivo. Dado que estos archivos ya no terminan en .php, no son procesados por el motor de PHP y pueden ser leídos por cualquiera. Esto puede llevar a la divulgación de información sensible."

remediation: "Realizar una limpieza de medios para eliminar bases de datos, archivos antiguos y de respaldo. Además, dejar solo los archivos necesarios en el directorio raíz de la instalación de WordPress."

condition: none

rules:

- c:sh -c "cd \$wp\_install\_dir; ls -la" -> r:.zip|.back|.backup|.bak|.old|.previous|.sql

- id: 100004

title: "Fortalecimiento de WordPress: Asegurar que no se utilicen nombres de cuentas administrativas comunes."

description: "Esta política verifica si se utilizan nombres de cuentas administrativas comunes (por ejemplo, admin, administrator, webmaster)."

rationale: "El uso de nombres de cuentas administrativas comunes aumenta la probabilidad de un ataque de fuerza bruta exitoso."

remediation: "Renombrar todas las cuentas administrativas predeterminadas y utilizar nombres de cuentas administrativas poco comunes."

condition: none

rules:

- c:runuser -l \$wp\_user -c "wp user list --field=user\_login --path=\$wp\_install\_dir" -> r:admin|administrator|backup|webmaster

- id: 100005

title: "Fortalecimiento de WordPress: Asegurar que la navegación de directorios esté deshabilitada."

description: "Esta política verifica si se puede listar el contenido de directorios sensibles (por ejemplo, wp-includes, wp-admin, wp-content)."

rationale: "Cuando la navegación de directorios está habilitada en un servidor web, puede llevar a la divulgación de información sensible y permitir el listado del contenido de directorios privilegiados."

remediation: "Deshabilitar la navegación de directorios agregando 'Options All -Indexes' en el archivo .htaccess de esta instalación de WordPress."

condition: all

rules:

- c:cat \$wp\_install\_dir/.htaccess -> r:Options\sAll\s\Indexes

- id: 100006

title: "Fortalecimiento de WordPress: Asegurar que los permisos de las carpetas de WordPress estén configurados en 755."

description: "Esta política verifica los permisos de las carpetas en las instalaciones de WordPress."

rationale: "El uso de permisos incorrectos en las carpetas de una instalación de WordPress puede dejar los archivos en esos directorios expuestos a modificaciones no autorizadas."

remediation: "Establecer todas las carpetas en el directorio de WordPress en 755 usando el comando chmod."

condition: all

rules:

- c:stat -c '%a' \$wp\_install\_dir/wp-admin -> r:755
- c:stat -c '%a' \$wp\_install\_dir/wp-includes -> r:755
- c:stat -c '%a' \$wp\_install\_dir/wp-content -> r:755
- c:stat -c '%a' \$wp\_install\_dir/wp-content/plugins -> r:755
- c:stat -c '%a' \$wp\_install\_dir/wp-content/themes -> r:755

- id: 100007

title: "Fortalecimiento de WordPress: Asegurar que no haya plugins desactualizados."

description: "Esta política verifica que no haya plugins de WordPress desactualizados en esta instalación de WordPress."

rationale: "Los plugins desactualizados pueden tener vulnerabilidades que actores maliciosos pueden explotar para tomar el control de un sitio de WordPress y, posteriormente, del servidor."

remediation: "Actualizar todos los plugins de WordPress."

condition: none

rules:

- c:runuser -l \$wp\_user -c "wp plugin list --field=update --path=\$wp\_install\_dir" ->

r:available

- id: 100008

title: "Fortalecimiento de WordPress: Asegurar que no haya temas de WordPress"

desactualizados."

description: "Esta política verifica que no haya temas de WordPress desactualizados en esta instalación de WordPress."

rationale: "Los temas desactualizados pueden tener vulnerabilidades que actores maliciosos pueden explotar para tomar el control de un sitio de WordPress y, posteriormente, del servidor."

remediation: "Actualizar todos los temas de WordPress."

condition: none

rules:

- c:runuser -l \$wp\_user -c "wp theme list --field=update --path=\$wp\_install\_dir" ->

r:available

- id: 100009

title: "Fortalecimiento de WordPress: Asegurar que un plugin de seguridad esté instalado y activo."

description: "Esta política verifica si el plugin de seguridad de WordPress especificado por la organización (\$wp\_security\_plugin) está instalado."

rationale: "Los plugins de seguridad pueden proporcionar una medida de protección contra exploits comunes dirigidos a sitios web de WordPress, como ataques de fuerza bruta e inyecciones SQL. La presencia de un plugin de seguridad reducirá significativamente la superficie de ataque."

remediation: "Instalar y activar el plugin de seguridad \$wp\_security\_plugin."

condition: all

rules:

- c:runuser -l \$wp\_user -c "wp plugin is-active \$wp\_security\_plugin --path=\$wp\_install\_dir; echo \$" -> r:0

- id: 100010

title: "Fortalecimiento de WordPress: Asegurar que el archivo wp-config.php no sea accesible públicamente."

description: "Esta política verifica si el archivo wp-config.php es accesible desde el navegador."

rationale: "El archivo wp-config.php contiene credenciales y configuraciones críticas. Si es accesible públicamente, un atacante podría extraer información sensible."

remediation: "Asegurar que el servidor web bloquea el acceso a wp-config.php mediante reglas en .htaccess o configuración de Nginx."

condition: all

rules:

- c:curl -I \$wp\_host/wp-config.php -> r:403|404



```

- id: 100011
  title: "Verificar si xmlrpc.php está accesible desde la web."
  description: "Si xmlrpc.php responde a solicitudes HTTP, podría ser un riesgo de seguridad."
  rationale: "Si este archivo no es necesario, es recomendable bloquear el acceso desde el
servidor web."
  remediation: "Bloquear el acceso a xmlrpc.php en el servidor web configurando las reglas
adecuadas en .htaccess o nginx."
  condition: all
  rules:
    - c: curl -s -o /dev/null -w "%{http_code}" $wp_host/xmlrpc.php -> r: ^(200|403)$
- id: 100012
  title: "Verificar si la API REST de WordPress expone usuarios."
  description: "Si el endpoint /wp-json/wp/v2/users devuelve datos en JSON y no un error de
acceso, los nombres de usuario pueden ser extraídos."
  rationale: "Exponer usuarios a través de la API REST facilita ataques de fuerza bruta y
enumeración de cuentas."
  remediation: "Restringir el acceso al endpoint o deshabilitar la API REST si no es
necesaria."
  condition: all
  rules:
    - c: curl -s $wp_host/wp-json/wp/v2/users | grep -E '"id":| "name":' -> r: .*

```

Este archivo establecerá las configuraciones que serán revisadas por Wazuh, como permisos de archivos, configuración de autenticación, cada regla tiene su información acerca de su funcionamiento en el apartado "*description*".

## Paso 2: Configurar el Agente de Wazuh

Para que Wazuh pueda realizar el análisis de seguridad en la instalación de WordPress, es necesario configurar el agente en el servidor para que envíe los registros hacia el servidor y sea procesado.

- Instalar el agente de Wazuh en el servidor:

```

wget <https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.9.2-1_amd64.deb>
sudo dpkg -i wazuh-agent_4.9.2-1_amd64.deb
sudo apt --fix-broken install

```

- Editar la configuración del agente para habilitar el análisis SCA:

```
sudo nano /var/ossec/etc/ossec.conf
```

```
Wazuh - Agent - Default configuration for debian 12
More info at: https://documentation.wazuh.com
Mailing list: https://groups.google.com/forum/#!forum/wazuh
→

<ossec_config>
  <client>
    <server>
      <address>172.21.0.3</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>debian, debian12</config-profile>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <crypto_method>aes</crypto_method>
  </client>
```

- Agregar la configuración necesaria dentro del archivo de configuración del agente:

```
<policies>
  <policy>/home/local_sca_policies/custom_wordpress_policy.yml</policy>
</policies>
```

```
<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
  <policies>
    <policy>/home/local_sca_policies/custom_wordpress_policy.yml</policy>
  </policies>
</sca>
```

- Reiniciar el agente de Wazuh para aplicar los cambios:

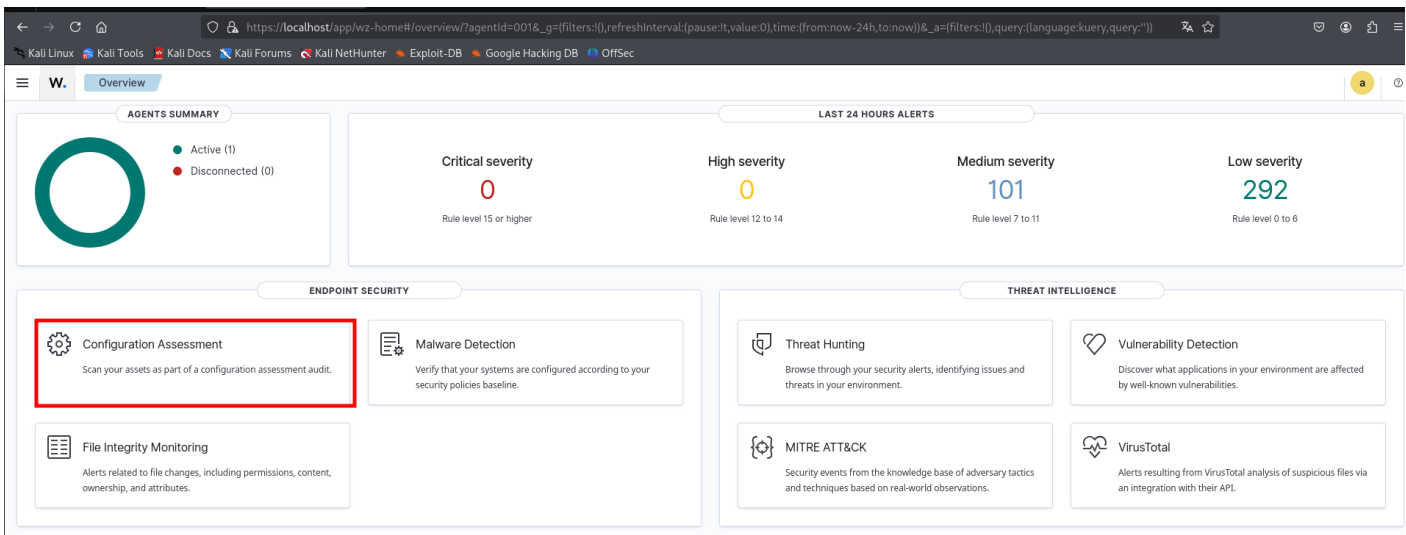
```
/var/ossec/bin/wazuh-control restart
```

```
root@c456798be056:/home# /var/ossec/bin/wazuh-control restart
Killing wazuh-modulesd ...
Killing wazuh-logcollector ...
Killing wazuh-syscheckd ...
Killing wazuh-agentd ...
Killing wazuh-execd ...
Wazuh v4.9.2 Stopped
Starting Wazuh v4.9.2 ...
Started wazuh-execd ...
Started wazuh-agentd ...
Started wazuh-syscheckd ...
Started wazuh-logcollector ...
Started wazuh-modulesd ...
Completed.
root@c456798be056:/home#
```

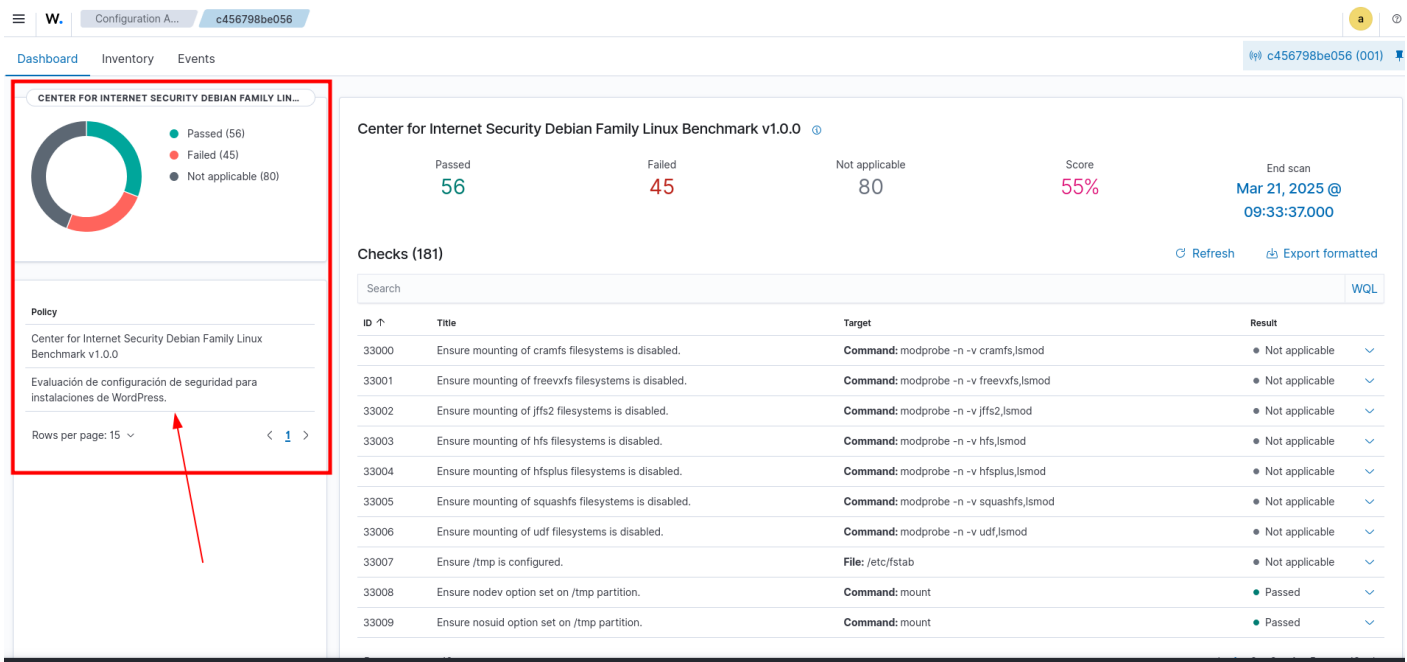
## Verificación de Resultados

Una vez configurado el análisis de seguridad en Wazuh, puedes verificar los resultados desde la interfaz gráfica.

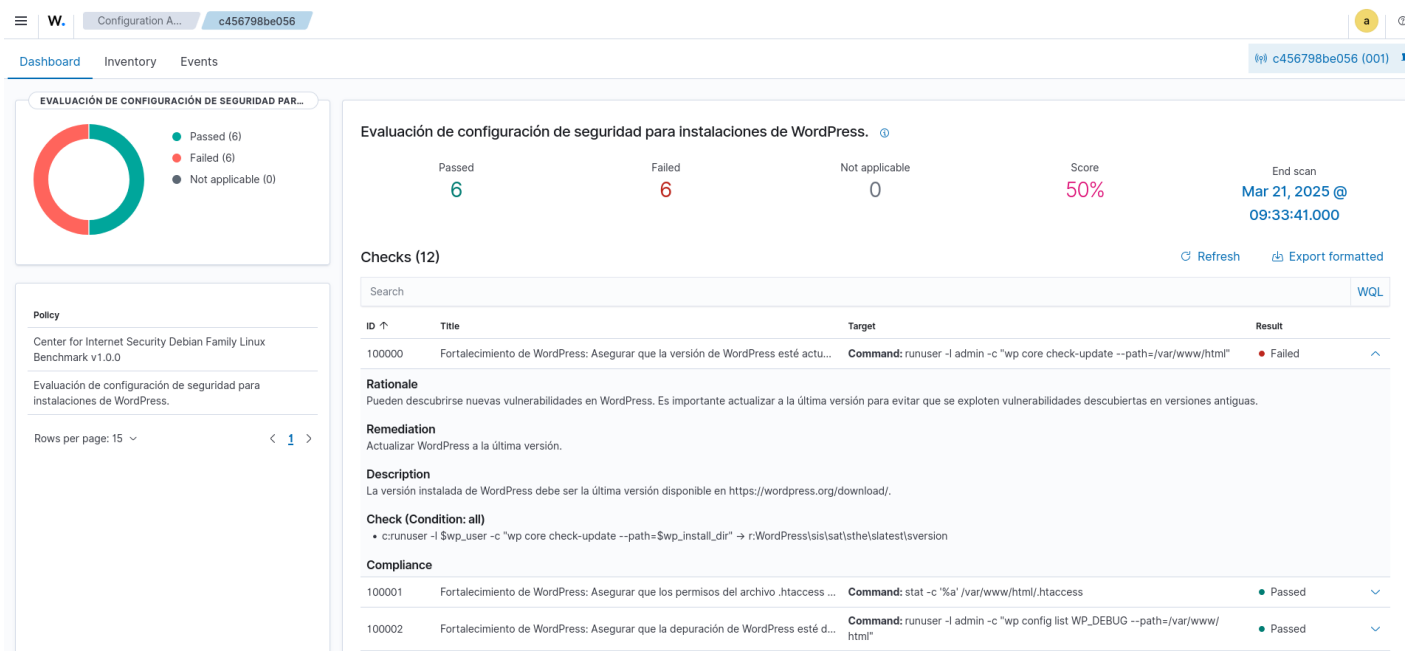
- Accede a la interfaz de Wazuh.
- Navega a Security Configuration Assessment.



- Revisa los resultados del módulo SCA de seguridad en WordPress.



- Aplica las recomendaciones de Wazuh para corregir vulnerabilidades identificadas.



Con esta guía, has aprendido a instalar y configurar Wazuh para comprobar la seguridad de una instalación de WordPress. Implementando las políticas de seguridad adecuadas y utilizando WP-CLI, puedes obtener información detallada sobre posibles vulnerabilidades y corregirlas a tiempo para fortalecer la seguridad de tu sitio.

Para obtener más información sobre la configuración avanzada de Wazuh, consulta la documentación oficial.

## Recomendaciones.

- Mantén tu servidor de Wazuh actualizado para garantizar compatibilidad con nuevas versiones de WordPress.
  - Verifica periódicamente los análisis SCA para detectar configuraciones inseguras antes de que puedan ser explotadas.
  - No mantengas las configuraciones predeterminadas tras una instalación.
- 

Revision #12

Created 19 marzo 2025 09:20:18 by Jhon Alan

Updated 31 marzo 2025 11:00:08 by Ricardo Alberto