LINEAMIENTOS Y GUÍAS TÉCNICAS PARA LA ADMINISTRACIÓN SEGURA DE SITIOS WEB

Aprobado con Resolución Administrativa

AGETIC/RA/0045/2025, de 16 de Junio de 2025

AGENCIA DE GOBIERNO ELECTRÓNICO Y TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

ÍNDICE

LINEAMIENTOS Y GUÍAS TÉCNICAS PARA LA ADMINISTRACIÓN SEGURA DE SITIOS WEB

INTRODUCCIÓN

La Agencia de Gobierno Electrónico y Tecnologías de Información y Comunicación tiene la función de evaluar la seguridad de los sistemas de información de las entidades públicas, comunicar y otorgar información acerca de incidentes y vulnerabilidades, además de realizar otras tareas orientadas a la mejora de la seguridad de la información de las entidades del sector público, funciones que están establecidas en el Decreto Supremo 2514.

Además la norma establece obligaciones en materia de seguridad informática a las entidades del sector público sobre la notificación de incidentes y la solución de vulnerabilidades.

El Centro de Gestión de Incidentes Informáticos comunica y otorga información acerca de incidentes y vulnerabilidades identificadas mediante mecanismos de monitoreo, evaluaciones de seguridad y reportes de organismos de similar función.

Los casos gestionados por el CGII evidencian una gran cantidad de vulnerabilidades a causa de errores de configuración en el servidor web y en el CMS que derivaron en incidentes de seguridad, producto del uso de contraseñas débiles o por defecto, software de servidor y aplicación desactualizados, y uso de componentes con vulnerabilidades, entre otras.

Con el objetivo de prevenir y reducir la ocurrencia de incidentes informáticos en entornos web, se elaboran los lineamientos y las guías técnicas para la administración segura de sitios web que implementan CMS (Sistemas de Gestión de Contenidos). Las prácticas descritas no limitan la adopción e implementación de otras acciones orientadas a la seguridad de la información.

El documento cubre prácticas de seguridad del sistema operativo, software de servidor web, gestor de base de datos y sitios web que implementan CMS en general.

1. SEGURIDAD DEL SISTEMA OPERATIVO

1.1. Instalar actualizaciones de seguridad

Verificar con regularidad si existen actualizaciones para el sistema operativo disponibles en sus repositorios oficiales, con especial atención a parches de seguridad de riesgo crítico y alto. En sistemas críticos se recomienda aplicar las actualizaciones en ambiente de prueba (test) para posteriormente aplicarlas en el entorno de producción.

1.2. Controles de acceso y autenticación

Mantener el control de las cuentas de usuario habilitadas en el sistema, así como los privilegios de cada una, las altas y bajas de usuarios deben estar documentadas y autorizadas. El acceso a sistemas críticos deberían estar autorizados por direcciones IP y MAC.

Las cuentas no deben ser compartidas por sus titulares, ya que elimina la responsabilidad y aumenta el riesgo de accesos y acciones no autorizadas en el sistema.

Utilizar un par de claves como método de autenticación en lugar de la contraseña, restringir la autenticación del usuario root o administrador y limitar los intentos de inicio de sesión fallidas.

Las conexiones remotas deben realizarse por una red privada virtual y/o protocolos seguros como SSH.

1.3. Deshabilitar/desinstalar servicios innecesarios

Instalar únicamente software necesario para el despliegue de la aplicación, sitio o servicio web, puesto que la falta de control sobre el software instalado podría exponer a ataques y explotación de vulnerabilidades. También es recomendable deshabilitar protocolos, servicios para compartir archivos.

2. SEGURIDAD DEL SERVIDOR WEB

2.1. Instalar actualizaciones de seguridad

Instalar con regularidad actualizaciones de seguridad, con especial atención de aquellas que impliquen la ejecución remota de código o cuenten con puntuación CRÍTICA/ALTA dentro de la valoración Common Vulnerability Scoring System (CVSS) y publicado en la base de Common Vulnerabilities and Exposures (CVE).

Los proveedores de software de servidor web publican regularmente avisos de parches de seguridad para sus productos, al cual se recomienda suscribirse para estar al día con avisos de actualizaciones.

Es importante considerar las recomendaciones del proveedor respecto al procedimiento a seguir para aplicar correctamente las actualizaciones. Para servicios críticos es recomendable probar el parche primero en ambiente test y luego en producción.

2.2. Deshabilitar el listado de directorios

En algunos software de servicio web viene habilitado por defecto el listado de directorios del sitio web que permite navegar, visualizar y descargar los archivos desde el directorio raíz del sitio, estos podrían contener datos sensibles y confidenciales como ser: nombres de usuario, contraseñas, parámetros de configuración, entre otros. Incluso se han identificado copias de respaldo en el mismo directorio, lo cual es una mala práctica en entornos web.

Consultar el numeral 2.1 de la Guía de configuración segura de servidores web para deshabilitar esta función.

2.3. Quitar archivos/funciones que puedan exponer datos relevantes

Algunos lenguajes de programación implementan funciones, que se usan comúnmente en entornos de desarrollo, que permiten visualizar información sobre la configuración del sitio a través de un archivo (por ejemplo: info.php) a efectos de verificar la correcta instalación y configuración en el sistema. Sin embargo este tipo de archivos debe ser removido o restringido en el ambiente de producción ya que devela información de las versiones, el sistema operativo y otros datos que podrían ser usados por un actor de amenaza.

Consultar el numeral 2.2 de la Guía de configuración segura de servidores web .

2.4. Quitar/restringir directorios y archivos de control de versiones

Restringir el acceso a directorios de control de versiones (por ejemplo: .git y .svn) debido a que contienen datos de configuración del repositorio de código. Entre algunos datos que se podrían exponer están: nombres de usuario, rutas, parámetros de configuración, historial de commits, código fuente, entre otros.

Consultar el numeral 2.3 de la Guía de configuración segura de servidores web para restringir el acceso.

2.5. Restringir el acceso a archivos de variables de entorno

Existen archivos de configuración que son usados para el almacenamiento de variables de entorno en ambientes de desarrollo, pruebas, producción y otros. Estos podrían contienen datos como nombres de usuario y contraseñas para la conexión con la base de datos, tokens de acceso, cuentas de correo electrónico, direcciones IP, nombres de hosts y otros datos necesarios para la funcionalidad del software.

Estos datos expuestos podrían ser usados para ataques específicos , por ello es importante restringir el acceso a estos y a archivos como ser “.env”, “.printenv”, “composer.json”, “web.config”, “main.yml”, entre otros.

Consultar el numeral 2.4 de la Guía de configuración segura de servidores web para restringir el acceso a estos archivos.

2.6. Configurar el registro de logs de acceso y error del servidor web

Una de las configuraciones imprescindibles en el servidor web es el registro de logs, que permitirán registrar datos relacionados a la funcionalidad del servidor, las solicitudes atendidas y toda información de actividad del servicio publicado, así como problemas que hayan podido ocurrir durante la operación del servicio.

Los registros son importantes en caso de incidentes de seguridad, ya que permitirá analizar y correlacionar eventos. Cada software de servidor tiene configuraciones diferentes para el registro de logs.

Consultar el numeral 2.5 de la Guía de configuración segura de servidores web para ver un ejemplo de configuración de logs.

2.7. Deshabilitar el despliegue de errores

El modo de depuración ayuda a identificar fallas en entornos de desarrollo y pruebas, sin embargo las aplicaciones web con la depuración de errores habilitada en entorno de producción podrían exponer variables de entorno, rutas, direcciones IP y código fuente, producto de excepciones no controladas.

Por lo que se debe deshabilitar el modo depuración para entornos de producción. Para deshabilitar esta característica consultar el numeral 2.6 de la Guía de configuración segura de servidores web.

2.8. Quitar contenido por defecto

Durante el despliegue de un sitio web se hace uso de distintos componentes, entre ellos, software del servidor web con directorios por defecto, manuales de instalación y uso; software de aplicación con funcionalidades de ejemplo, incluso la reutilización de código de terceros que luego de ser probada no se elimina.

Es importante eliminar todo archivo, código, módulo, complemento o componente que no se utilice, puesto que el mismo podría exponer información de versiones del software utilizado, puertas traseras, vulnerabilidades conocidas.

2.9. Realizar copias de seguridad

Otra de las acciones más importantes en la administración segura de un sitio, aplicación o servicio web es realizar copias de seguridad manuales o automatizadas de archivos funcionales y de configuración del sitio web, archivos de configuración del servidor web y sistema operativo. La periodicidad de las copias estará en función de la criticidad de la información.

Es importante que las copias se resguarden en medios y ubicaciones distintas al servidor web, incluyendo copias desconectadas de la red, las copias deben ser

probadas regularmente a efectos de verificar que las mismas se generan y se restauran correctamente, las copias permitirán responder en tiempos y acciones ante incidentes de seguridad informática.

2.10. Deshabilitar la información de versión del servidor web y sistema operativo

La versión del software del servidor puede ser usada para verificar si el servidor tiene vulnerabilidades conocidas y si existen exploits públicos que podrían ser usados para comprometer la seguridad del servidor, servicios y/o aplicaciones web que se ejecutan en el sistema.

Quitar la información de la versión del servidor web impedirá que se revele la versión del sistema operativo. Para realizar esta acción puede consultar el numeral 2.7 de la Guía de configuración segura de servidores web.

2.11. Habilitar cabeceras HTTP de seguridad

Agregar capas de seguridad al servidor web reducirá el riesgo de sufrir ataques, por ello es importante implementar directivas de seguridad en las cabeceras HTTP del servidor, entre las cabeceras más importantes se encuentran: Strict-Transport-Security (HSTS), Content-Security-Policy (CSP), X-Frame-Options y X-XSS-Protection.

Para configurar las cabeceras en un servidor web puede consultar el numeral 2.8 de la Guía de configuración segura de servidores web.

2.12. Instalar y configurar certificado SSL/TLS

Instalar y configurar el certificado de seguridad permitirá proteger la confidencialidad de datos de autenticación y otra información sensible que se intercambie entre el cliente y el servidor mediante el protocolo seguro HTTPS.

La alternativa para la implementación de certificados gratuitos, automatizados y abiertos es la autoridad de certificación Let’s Encrypt.

Consultar el numeral 2.9 de la Guía de configuración segura de servidores web para la instalación y configuración de certificados de seguridad.

3. SEGURIDAD DEL SISTEMA GESTOR DE BASE DE DATOS

3.1. Instalar actualizaciones de seguridad

El sistema gestor de base de datos debe ser actualizado regularmente, con atención especial a parches de seguridad de severidad crítica y alta. La actualización debe seguir las recomendaciones y procedimientos establecidos por el proveedor.

En sistemas críticos la aplicación de parches debe ejecutarse primeramente en entorno de pruebas y luego en entorno de producción.

3.2. Controlar el acceso y la autenticación

Limitar el acceso al servicio mediante listas de control de acceso, el servicio no debe ser publicado a internet, asimismo sólo usuarios y hosts autorizados deberían poder acceder al servicio. Del mismo modo, se debe controlar el acceso y publicación a internet de aplicaciones web que gestionan la administración de bases de datos como phpmyadmin, phppgadmin o similares.

Usar contraseñas seguras y únicas para todos los usuarios, es importante mantener el registro, control de todos los usuarios con sus respectivos permisos autenticación y autorización, diferenciando si el acceso es local o remoto.

3.3. Establecer roles y privilegios

Establecer roles y privilegios para cada usuario a nivel del sistema gestor, bases de datos, tablas, vistas y otros. De ningún modo se debería establecer permisos de alto nivel para la conexión con la aplicación o usar la cuenta de súper administrador predeterminada, puesto que un ataque exitoso a la aplicación podría aprovechar los privilegios y comprometer todas las bases de datos.

3.4. Habilitar el registro de logs

Es importante verificar que el registro de logs esté habilitado en el sistema gestor de base de datos, siendo relevante lo datos referidos a eventos como ser: la creación de usuarios, bases de datos, tablas, vistas, funciones, conexión origen, dirección IP, fecha y hora del evento, entre otros datos relevantes.

Los registros de auditoría permitirán identificar, correlacionar y analizar actividad no autorizada junto con problemas de integridad de datos.

3.5. Realizar copias de seguridad

La realización de copias de seguridad en bases de datos es crítico, ya que en función de la importancia de los datos, se deben establecer la frecuencia y redundancia de las copias.

Las copias deben resguardarse en medios y ubicaciones diferentes al sistema gestor de base de datos, incluyendo copias desconectadas de la red. Las copias deben ser probadas regularmente para verificar su correcta restauración. Soluciones como Backup Archiving Recovery Open Sourced - BAREOS permiten gestionar las copias de respaldo de manera automatizada.

4. MONITOREO DE SERVICIOS Y RECURSOS

Monitorear el estado de los recursos de hardware del servidor, así como el estado de disponibilidad de servicios web, esto permitirá prevenir y responder adecuadamente ante problemas o ataques al sitio web. Se podría considerar más de un monitoreo para los servicios críticos, el primero instalado en la infraestructura de la institución con una visibilidad mayor respecto al uso y estado de los servicios, puesto que podrían

instalarse agentes en cada host a monitorear. El segundo monitoreo (respaldo) como servicio gratuito en la nube, con funcionalidades básicas que no requieren la instalación de agentes y basado en verificación de estados HTTP del servidor web.

Entre las alternativas libres para instalación en infraestructura de la institución están Zabbix para el monitoreo de redes, servicios, y goaccess para análisis de registros del servidor web (en caso de apache o nginx) por terminal o interfaz web.

5. SEGURIDAD EN SISTEMAS DE ADMINISTRACIÓN DE CONTENIDOS - CMS

5.1. Implementar ambiente de pruebas

Es una buena práctica implementar un ambiente de pruebas o test, que será una réplica del sitio en producción, a efectos de probar las actualizaciones de seguridad y garantizar que la disponibilidad en ambiente de producción no se vea afectada.

5.2. Instalar actualizaciones de seguridad

Esta práctica es fundamental en sistemas de administración de contenidos. Regularmente se publican actualizaciones de seguridad para el sistema base (core) y los componentes (plugins). Un alto porcentaje de sitios comprometidos tienen como causa raíz la falta de actualización del core y/o plugins.

Preferentemente las actualizaciones deben realizarse en el entorno de pruebas en función a las notas, pasos y requisitos de instalación que publica el CMS. En algunos sistemas es posible habilitar las actualizaciones automáticas. Se recomienda suscribirse a listas de avisos del CMS para estar al día con notificaciones de nuevas actualizaciones disponibles, el CGII publica periódicamente alertas y avisos de seguridad sobre vulnerabilidades en https://www.cgii.gob.bo/

5.3. Implementar plugins y componentes de fuentes confiables

Otro de los factores que inciden en el compromiso de sitios web es el uso de plugins y componentes de fuentes no confiables, es decir que no proceden del sitio web oficial del CMS. A partir de investigaciones realizadas en incidentes, el CGII identificó que estos componentes contenían puertas traseras para el acceso de actores maliciosos.

Es importante usar plugins descargados desde los sitios web oficiales del CMS, algunos otorgan la verificación de seguridad para complementos. No se deben usar plugins de paga pirateados o de fuentes desconocidas por el riesgo inminente de código malicioso.

5.4. Realizar copias de seguridad

Algunos CMS facilitan la realización de copias de respaldo con herramientas integradas en la instalación o como complementos, en cualquier caso, al tratarse de sitios web es fundamental realizar copias de seguridad al menos una vez por día, esto dependerá de la periodicidad de información a publicar en el sitio web. La copia debe considerar tanto

archivos del sitio web como de la base de datos. Estas copias deben resguardarse en ubicaciones diferentes al servidor web, incluyendo copias desconectadas de la red.

Contar con copias de respaldo hará que el proceso de respuesta ante un incidente sea más eficiente, porque se podrá restaurar una copia no comprometida y realizar acciones correctivas sobre la misma.

5.5. Controlar el acceso y la autenticación

Las credenciales de acceso no deben ser compartidas y mantener el control de qué usuarios tienen éstas. Evitar el uso de nombres de usuario predecibles o por defecto como: “admin”, “administrador”, etc. Revisar frecuentemente si existen usuarios nuevos y sus privilegios.

Todas las cuentas de usuario deben usar contraseñas fuertes y únicas, algunos CMS implementan esta funcionalidad como obligatoria, siendo importante habilitar el segundo factor de autenticación (2FA).

Los mensajes de error en la autenticación deben ser genéricos, no se debe revelar específicamente si el nombre o contraseña son incorrectas.

5.6. Protección contra ataques automatizados

El formulario de autenticación debe contar con protección por código captcha, esto evitará ataques de fuerza bruta o diccionario que busquen obtener nombres de usuario o contraseña válidos, asimismo se recomienda limitar el número de intentos de inicio de sesión fallidos.

5.7. Remover/deshabilitar contenido por defecto

Se recomienda eliminar el contenido por defecto como: temas, plugins, páginas, directorios y archivos que no se usan. En general cada CMS viene con una guía de instalación que indica los ajustes de seguridad que se deben realizar luego de una instalación exitosa.

5.8. Guías técnicas de seguridad

Se recomienda implementar las configuraciones de seguridad para sitios web, implementados en Sistemas de Administración de Contenidos, que se encuentran en las siguientes guías técnicas:

- Guía técnica de seguridad WordPress
- Guía técnica de seguridad Drupal

ANEXO 1 - GUÍA DE CONFIGURACIÓN SEGURA DE SERVIDORES WEB

1. INTRODUCCIÓN

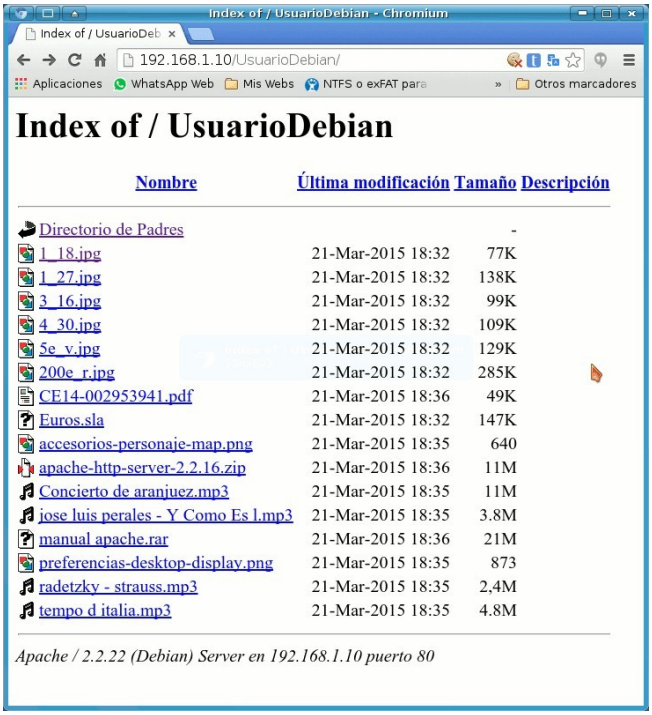
El servidor web es propenso a ciberataques por su exposición a internet, las cuales se podrían dar por la explotación de vulnerabilidades, configuraciones por defecto que se hayan dejado con la instalación inicial.

2. PRÁCTICAS DE SEGURIDAD

A continuación se presentan las prácticas de seguridad que se deben aplicar a servidores web, la guía hace referencia a Apache2 y Nginx por su amplio uso.

2.1. Deshabilitar el listado de directorios

Esta configuración es propia del servidor Apache2 y viene por defecto en la instalación. Permite listar los directorios y archivos del sitio web permitiendo: navegar, descargar y visualizar el contenido de archivos desde el directorio raíz del sitio, que podría contener datos sensibles o confidenciales como ser: nombres de usuario, contraseñas, copias de seguridad y archivos de configuración.



Las siguientes configuraciones fueron realizadas en Apache 2.4 mediante un usuario con privilegios sudo. Dependiendo del caso se puede elegir una de las siguientes opciones para deshabilitar el listado de directorios.

- Deshabilitar el módulo autoindex.

- Deshabilitar a través del archivo de configuración del sitio.
- Deshabilitar a través del archivo htaccess.

2.1.1. Deshabilitar el módulo autoindex

Deshabilitar la funcionalidad autoindex a nivel global:

sudo a2dismod autoindex

Después de ejecutar el comando, se mostrará el mensaje de advertencia el cual se tiene que responder con la siguiente frase:

To continue type in the phrase 'Yes, do as I say!' or retry by passing '-f': ***Yes, do as I say!***

Reiniciar el servidor.

sudo systemctl restart apache2.service

2.1.2. Deshabilitar a través del archivo de configuración del sitio

Este método deshabilita esta funcionalidad solo para el sitio en cuestión. Por ejemplo si se tiene el sitio www.sitio-de-prueba.com con el archivo de configuración (virtualhost) sitio-de-prueba.conf.

Agregar en el archivo la siguiente directiva:

```
<VirtualHost *:80>

...

<Directory /var/www/sitio-de-prueba>

        Options -Indexes

</Directory>

...

</VirtualHost>
```

Guardar y recargar la configuración.

sudo systemctl reload apache2.service

2.1.3. Deshabilitar a través del archivo .htaccess

Es una alternativa similar al archivo de configuración.

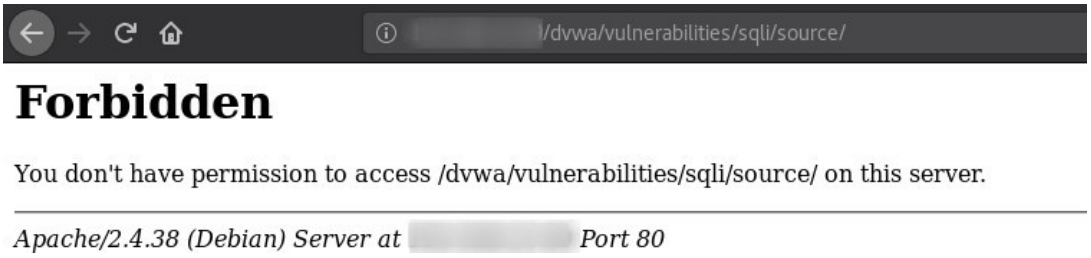
- Agregar en el archivo .htaccess del sitio respectivo:

Options -Indexes

- Guardar el archivo y reiniciar el servidor.

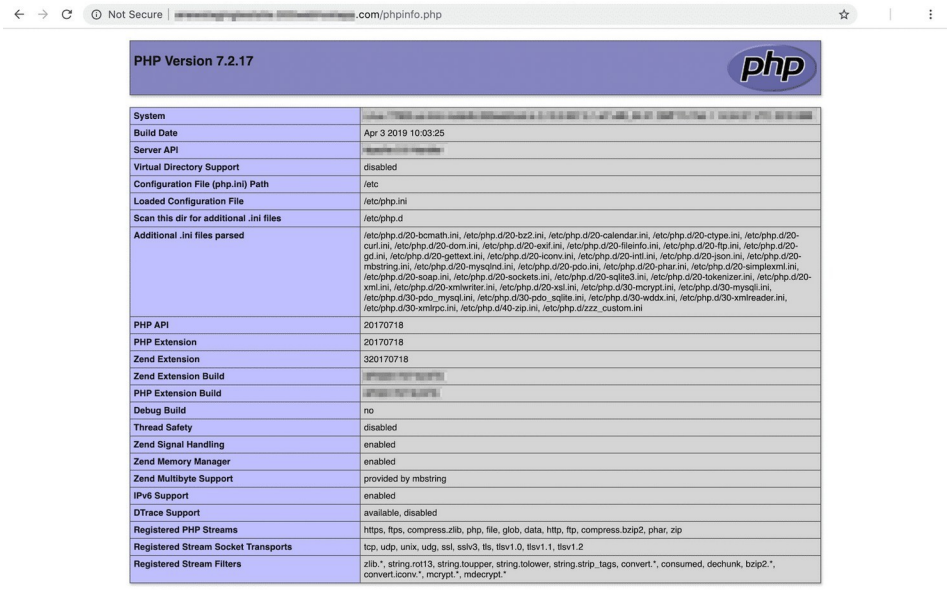
  $ sudo systemctl restart apache2.service

El resultado de aplicar una de las configuraciones anteriores, será la restricción de listar archivos y carpetas.



2.2. Quitar archivo info.php y deshabilitar la función phpinfo()

Afecta a aplicaciones desarrolladas bajo el lenguaje de programación PHP, una práctica que se usa en entorno de desarrollo es imprimir las características de php en un archivo con nombre info.php a efectos de verificar la correcta instalación y configuración de PHP en el sistema. Sin embargo este archivo debe ser removido del ambiente de producción ya que devela información de las versiones, el sistema operativo y otros datos que podrían ser usados por un actor de amenaza.



Estas soluciones fueron probadas en PHP versión 7.2.

Acceder al archivo /etc/php/7.2/apache2/php.ini y añadir la siguiente directiva:

expose_php = Off

Una opción alternativa es deshabilitar mediante el comando “disable_funtions”, para ello se puede añadir “phpinfo”:

disable_functions = ..., phpinfo

2.3. Quitar/restringir directorios y archivos de control de versiones .git

Una mala práctica en entornos de producción es exponer directorios como “.git”, “.svn” que contienen datos de configuración del control de versiones del código. Entre algunos datos que se podrían exponer están nombres de usuarios, rutas, parámetros de configuración, historial de commits, entre otros datos. Por ello es importante quitar o restringir el acceso.

Para restringir el acceso al archivo .git en apache, agregar la siguiente directiva en el archivo apache2.conf:

<Directory ~ “\.git”>

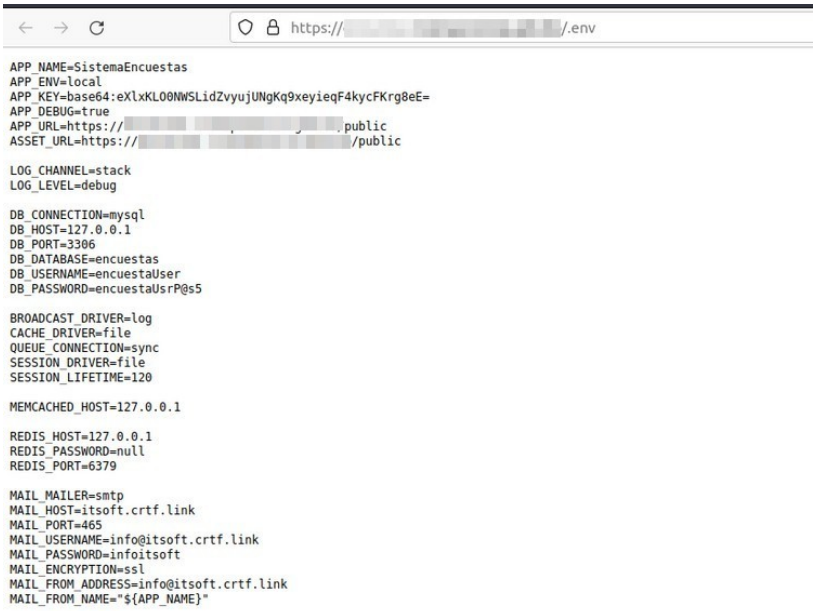
Order allow,deny

Deny from all

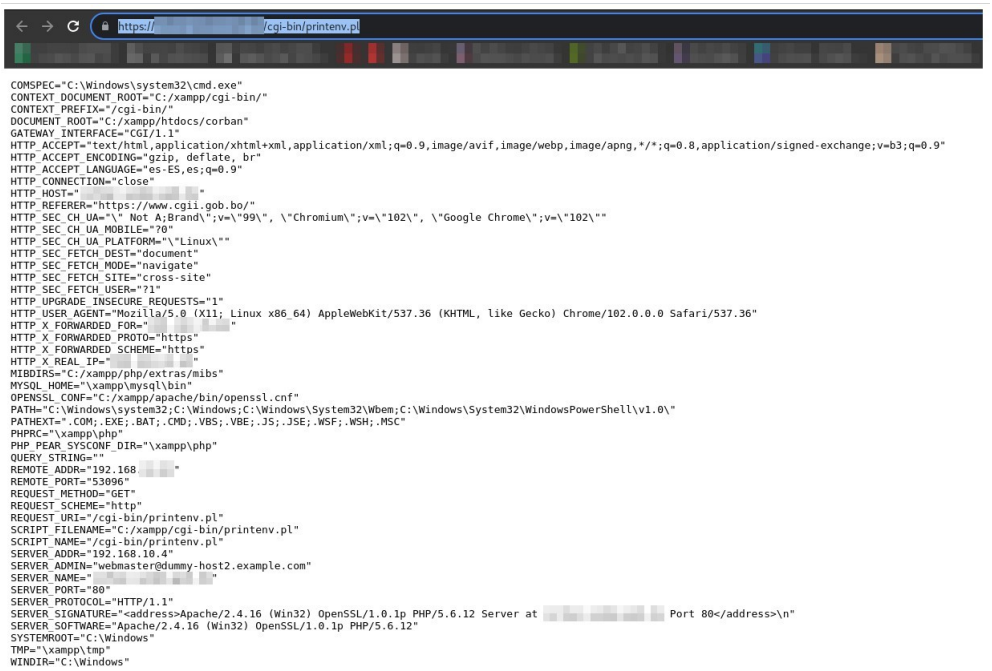
</Directory>

2.4. Restringir acceso a los archivos de configuración “.env” “printenv.pl”

El archivo .env es usado en proyectos para el almacenamiento de variables de entorno en ambiente development, test, production y otros. Contiene datos como nombres de usuario y contraseña para la conexión con la base de datos, tokens de acceso, cuentas de correo electrónico, direcciones IP, nombres de hosts y otros necesarios para la funcionalidad del software.



El archivo printenv.pl despliega las variables de entorno actuales.



A continuación cambiar los permisos de los archivos .env y printenv.pl a 400 ó 440 para restringir el acceso a los usuarios públicos.

chmod 440 .env

Otra opción para restringir el acceso a los archivos .env y printenv.pl es agregar las siguientes líneas al archivo de configuración del sitio web o su respectivo .htaccess:

<Directory /cgi-bin>

order deny,allow

deny from all

</Directory>

También es posible utilizar la siguiente configuración:

<Directory /cgi-bin>

Require all denied

</Directory>

Para el archivo .env:

<Files .env>

Order allow,deny

Deny from all

</Files>

2.5. Configurar el registro de logs de acceso y error del servidor web

2.5.1. Habilitar el registro de logs en apache

En el archivo de configuración de apache (/etc/apache2/sites-enabled/{subdominio}.conf), agregar las siguientes directivas:

ErrorLog ${APACHE_LOG_DIR}/{subdominio}-error.log

CustomLog ${APACHE_LOG_DIR}/{subdominio}-access.log combined

Se aconseja tener un error.log diferente por sitio web administrado.

2.5.2. Configurar el nivel de reporte de logs en apache

La configuración de loglevel permite seleccionar el nivel de detalle en el que se registrarán los logs, para ello se debe actualizar la variable “Loglevel” en el archivo apache2.conf (/etc/apache2/apache2.conf):

LogLevel info

Se puede seleccionar las siguientes opciones:

- warn: Advertencias
- info: Mensajes informativos.
- debug: Mensajes de depuración (producirá una gran cantidad de información).
- error: Errores producidos mientras se procesaba la solicitud.

2.5.3. Habilitar el registro de logs en PHP

Editar el archivo php.ini para habilitar el registro de logs:

log_errors = On;

error_log = /var/log/apache2/error_log;

2.5.4. Configurar el nivel de reporte de logs en PHP

Para configurar el nivel de reporte de logs en PHP se debe actualizar la variable "error_reporting" en el archivo php.ini:

error_reporting = E_ALL | E_STRICT;

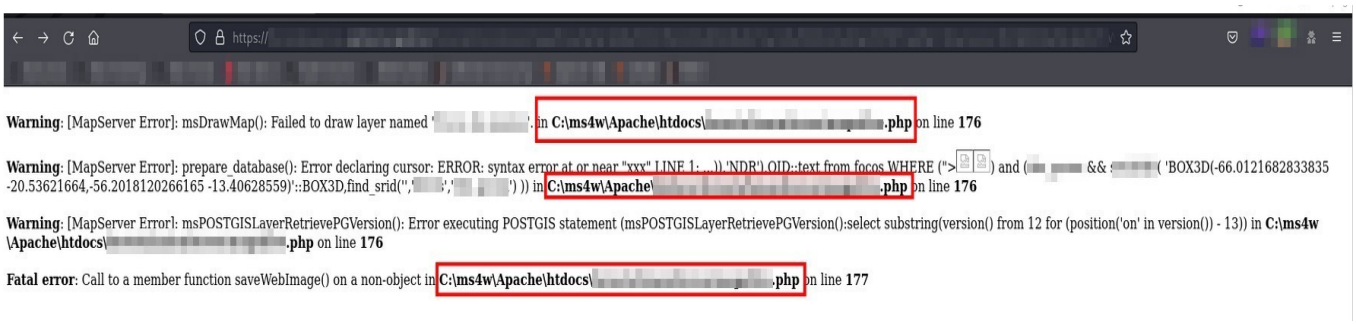
El nivel de reporte puede tener los siguientes valores más comunes:

- E_ERROR: Errores fatales que no se pudieron recuperar, como un problema de asignación de memoria. Los mismos detienen la ejecución del programa.
- E_WARNING: Errores durante la ejecución del programa (no fatales).
- E_ALL: Todos los errores, advertencias y noticias (por defecto).
- E_STRICT: Habilita la sugerencia de cambios al código para mejorar la interoperabilidad y compatibilidad futura .
- E_PARSE: Errores de análisis en tiempo de ejecución.
- E_NOTICE: Avisos en tiempo de ejecución. Indica que la secuencia de comandos encontró algo que podría indicar un error.

Para ver más opciones, visitar la documentación oficial de PHP.

2.6. Deshabilitar el despliegue de errores

Es importante deshabilitar el despliegue de errores por excepciones no controladas en la aplicación web, debido a que cualquier mensaje de error mostrado al usuario final incluye no sólo la información del servidor, sino también un mensaje de excepción detallada que podría incluir variables de entorno, rutas, código fuente del recurso donde se produjo el error.



Para deshabilitar el despliegue de errores en PHP se debe editar el archivo php.ini cambiando a Off la directiva “display_errors”:

display_errors = Off;

2.7. Deshabilitar la información de versión del servidor web

Por defecto apache y nginx despliegan su versión utilizada cuando la solicitud a una página del sitio web responde con código de errores 40X.

Para evitar desplegar la versión de apache, se debe editar el archivo de configuración “/etc/apache2/apache2.conf” y agregar las siguientes directivas:

ServerTokens Prod

ServerSignature Off

El resultado será el siguiente:

En el caso de NGINX se debe editar el archivo nginx.conf actualizando la variable “server_tokens” con el valor “off”:

```
http{

    ...

    server_tokens off;

    ...

}
```

Finalmente se debe reiniciar nginx:

```
systemctl restart nginx
```

2.8. Habilitar cabeceras HTTP de seguridad

Las cabeceras de seguridad agregan capas de seguridad adicionales al servidor web, para configurar las cabeceras de seguridad más importantes en Apache2 se debe agregar en el archivo de configuración del sitio web (por ejemplo: /etc/apache2/sites-enabled/example.conf):

```
<IfModule mod_headers.c>

    // Cabecera HTTP Strict Transport Security (HSTS)

    Header set Strict-Transport-Security "max-age=31536000;
includeSubDomains; preload"

    // Content Security Policy (CSP)

    Header always set Content-Security-Policy "default-src 'self'; font-src
*;img-src * data:;     script-src *; style-src *;"

    //X-XSS-Proteccion

    Header set X-XSS-Protection "1; mode=block"

    //X-Frame-Options

    Header always set X-Frame-Options "SAMEORIGIN"

</IfModule>
```

Seguido, habilitar el módulo “headers”:

```
sudo a2enmod headers
```

Reiniciar Apache:

sudo systemctl restart apache2

2.9. Instalar y configurar certificado SSL/TLS

Let's Encrypt es un servicio que ofrece certificados SSL gratuitos a través de una API. Certbot es un cliente de Let's Encrypt, que tiene varias formas de validar el dominio, busca certificados y configura automáticamente Apache y Nginx.

Primero se deberá instalar Certbot:

sudo apt install certbot

Ejecutar Certbot:

sudo ufw allow 80

Ejecutar certbot en un servidor web temporal (--standalone) para obtener los certificados, en el ejemplo se usa el parámetro 'http', para https se utiliza tls-sni (--preferred-challenges), e introducir el nombre de dominio (-d):

sudo certbot certonly --stantalone --preferred-challenges http -d entidad.gob.bo

Para la configuración automática de Apache se puede usar:

sudo cerbot --apache -d entidad.gob.bo

Para configurar la aplicación, primero debe verificar los archivos relacionados al certificado:

sudo ls /etc/letsencrypt/live/entidad.gob.bo

Salida:

cert.pem chain.pem fullchain.pem privkey.pem README

2.9.1. Renovación automática del certificado

Los certificados de Let's Encrypt son válidos por 90 días, para hacer la renovación automáticamente se debe agregar un script a /etc/cron.d el cual se ejecutará 2 veces al día. Para completar la renovación es necesario aplicar los cambios editando el archivo:

sudo nano /etc/letsencrypt/renewal/entidad.gob.bo.conf

Agregar la opción renew_hook que permite realizar tareas posteriores a la obtención del certificado:

renew_hook = systemctl reload

<nombre del servicio>

Ejecutar certbot para comprobar que no haya errores:

sudo certbot renew --dry-run

2.9.2. Desabilitar protocolos inseguros

En el archivo de opciones de SSL, encontrará que los protocolos SSLv2 y SSLv3 ya están deshabilitados. Esto se debe a las inseguridades de estos protocolos y no deben utilizarse. Además, TLS v1.0 también es un protocolo heredado y no debe usarse. Sin embargo, aún es necesario para permitir que funcionen los navegadores web más antiguos. Se lo debe habilitar sólo si es absolutamente necesario, para deshabilitar debe editar los archivos /etc/apache2/mods-enabled/ssl.conf, comentando y reemplazando:

SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1

Por:

SSLProtocol TLSv1.2

y en el archivo /etc/letsencrypt/options-ssl-apache.conf comentar y reemplazar:

SSLProtocol all -SSLv2 -SSLv3

Por:

SSLProtocol +all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1

2.9.3. Deshabilitar la compresión ssl

En 2012, la vulnerabilidad del ataque CRIME demostró que la compresión TLS no se puede implementar de forma segura. Por lo tanto, no debe usarse y la única solución es deshabilitar la compresión TLS por completo. Verifique que lo siguiente esté en su archivo de configuración.

SSLCompression off

2.10. Proteger archivos de configuración

Es necesario que se configure correctamente los permisos de los archivos de configuración que contienen información sensible, de modo que se encuentre protegido contra accesos no autorizados, en ese sentido se recomienda considerar los siguientes permisos:

- Para archivos de configuración:

  Archivos de configuración (composer.json, config.yml, etc) - 640

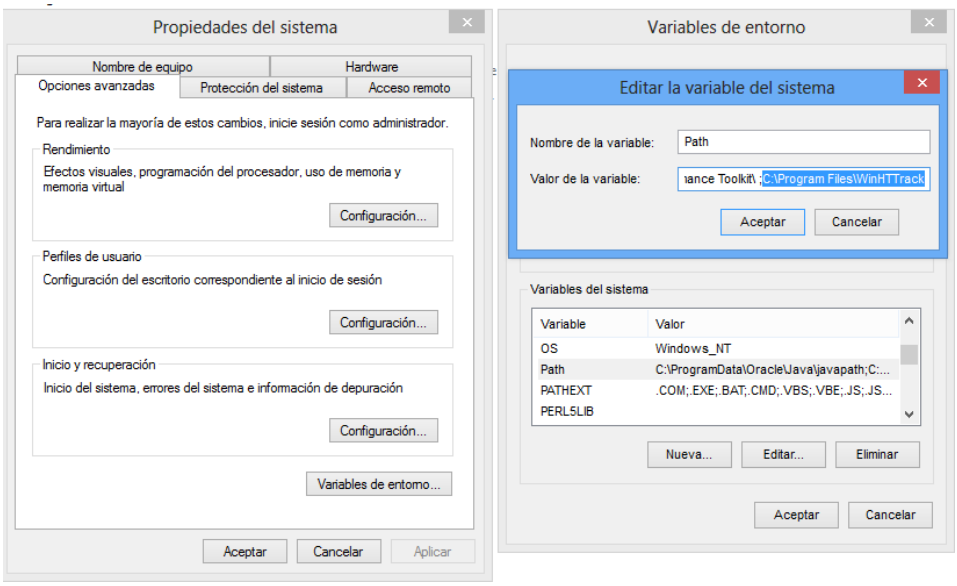
  configuration.php, settings.php: 440 o 400

- Para otros archivos - 644
- Otras carpetas - 755

2.11. Copia estática del sitio web

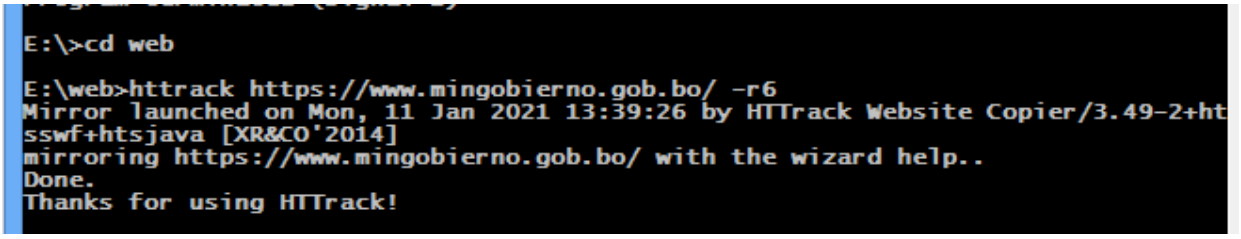
Se recomienda generar una copia estática reciente del sitio web para que pueda ser usada en caso de un incidente de seguridad. Para realizar esta tarea se establecen los siguientes pasos:

- Descargar Httrack https://www.httrack.com/page/2/en/index.html y después de instalar, modificar la variable de entorno PATH, agregar la ruta del Httrack “C:\ Program Files\WinHTTrack”:



➢ Ejecutar el comando para crear la copia estática del sitio web:

httrack https://[dominio.gob.bo]/ -r6



Se recomienda realizar estas copias estáticas del sitio web cada vez que tenga un cambio significativo en su contenido.

ANEXO 2 - GUÍA TÉCNICA DE SEGURIDAD WORDPRESS

1. INTRODUCCIÓN

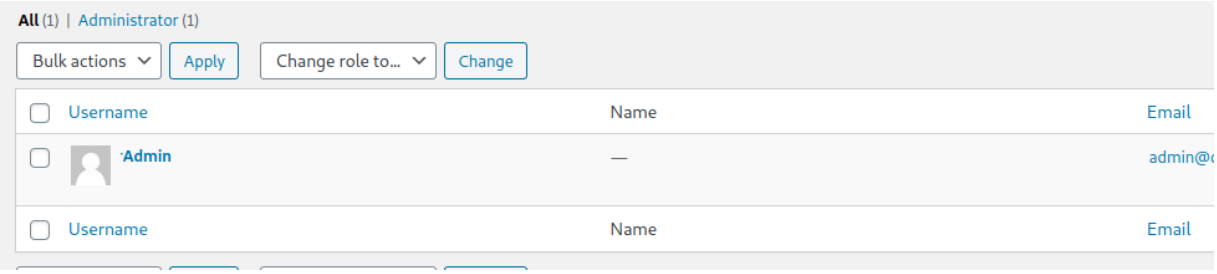
WordPress es un sistema de gestión de contenido (CMS, Content Management System), que permite crear sitios web, su popularidad ha logrado que resulte muy atractivo para los “ciberatacantes”, con el fin de explotar vulnerabilidad

2. ASEGURANDO WORDPRESS

Para mitigar el riesgo de ataques a Wordpress, recomendamos las siguientes buenas prácticas de seguridad.

2.1. Configurar el control de acceso de usuarios

Ingresar al panel de administración de usuarios del sitio web: https://[mi-dominio.gob.bo]/wp-admin/users.php y eliminar a los usuarios administradores que no se usan actualmente:



2.2. Uso de contraseñas seguras

Uno de los vectores de ataque más recurrentes son las contraseñas inseguras debido al uso de patrones predecibles. Una contraseña segura tiene las siguientes características:

- 12 o más caracteres,
- Uso de mayúsculas
- Uso de números
- Uso de caracteres especiales

Para asegurar el uso de contraseñas seguras puede instalar el plugin *“Password Policy”*

2.3. Autenticación de dos factores

La autenticación de dos factores es un método para proteger cuentas de usuario ante robo o filtración de contraseñas e inclusive ataques de fuerza bruta. La protección se realiza a través del envío de códigos o autorización por medios alternativos donde solo el dueño legítimo de la cuenta tendría acceso.

Una opción para su implementación en WordPress es a través del plugin *"Two Factor Auth".*

2.4. Límite de intentos de inicio de sesión

Los atacantes maliciosos suelen utilizar técnicas de fuerza bruta o ataques de diccionario al formulario de inicio de sesión para obtener acceso al panel de administración del sitio. Para proteger el sitio en WordPress se recomienda limitar la cantidad de intentos de inicio de sesión, esto puede realizarse a través del plugin "Limit Login Attempts Reloaded".

2.5. Protección por CAPTCHA

La implementación de código captcha evita ataques de fuerza bruta contra formularios de inicio de sesión, también sirve para evitar ataques automatizados de envío de formularios en el sitio web.

Para su implementación en WordPress puede utilizar el plugin *"Google reCaptcha"*

2.6. Actualización del core de Wordpress

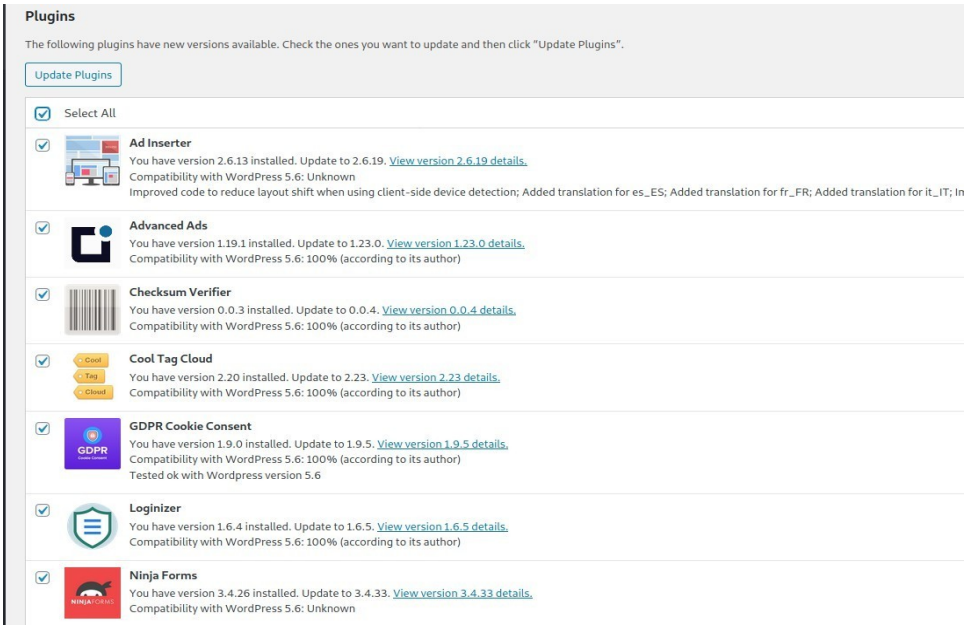
Ingresar a su sitio web https://[mi-dominio.gob.bo]/wp-admin/update-core.php y hacer click en el botón "Actualizar ahora".

2.7. Actualizar los plugins de WordPress

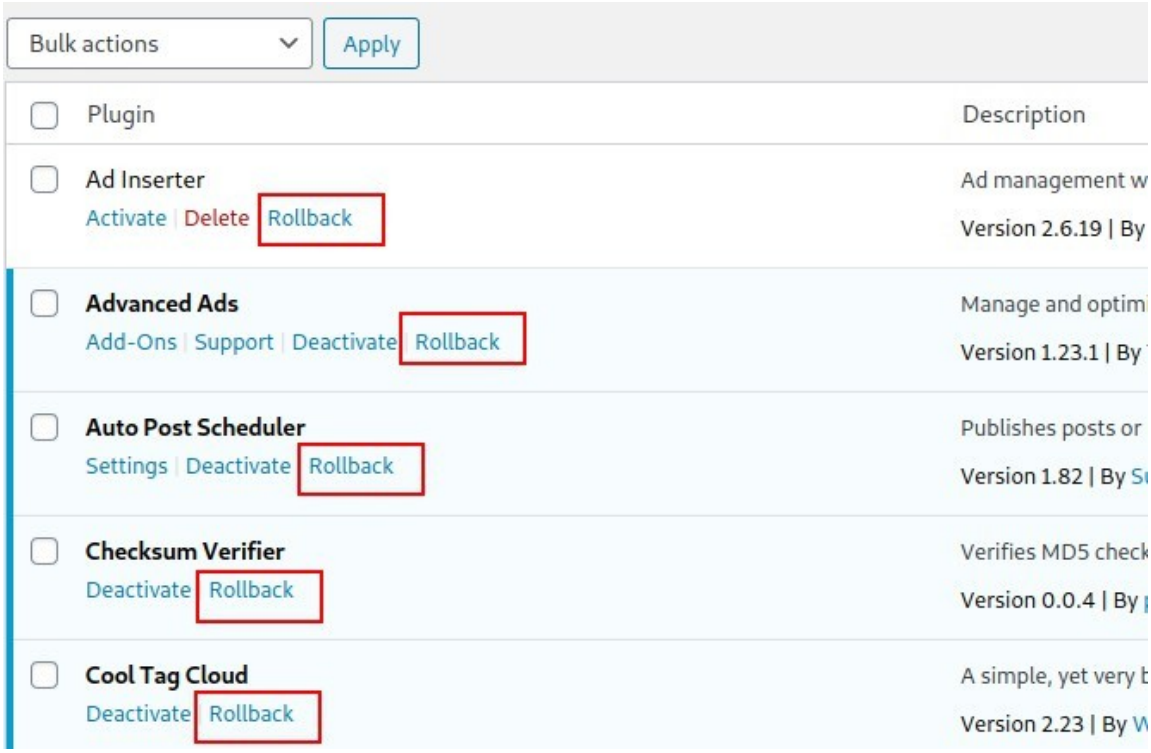
Previamente instalar y activar el plugin WP-Rollback (https://es.wordpress.org/plugins/wp-rollback/) que será útil en caso que la actualización de algún plugin no sea exitosa.

Realizar un backup de la base de datos.

Ingresar al sitio https://[mi-dominio.gob.bo]/wp-admin/update-core.php y seleccionar todos los plugins y presionar el botón “Actualizar plugins”.



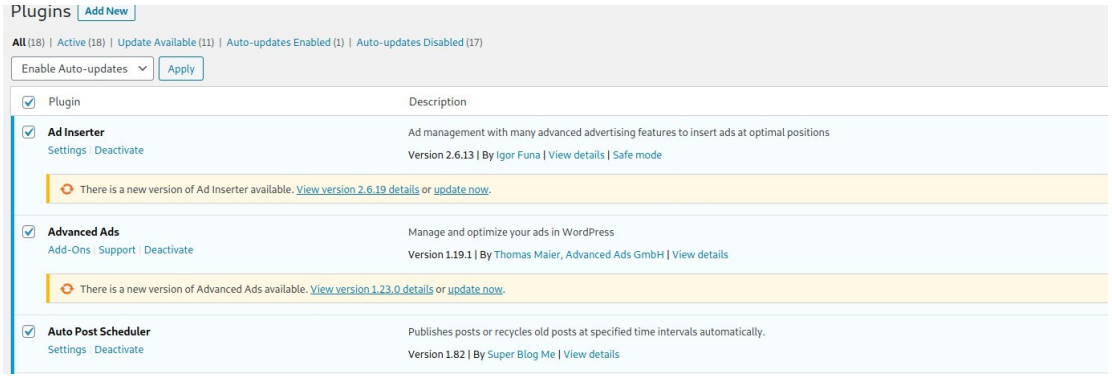
En caso de existir un error durante el proceso de actualización, realizar un ROLLBACK (https://[mi-dominio.gob.bo]/wp-admin/plugins.php).



2.8. Habilitar actualizaciones de seguridad automáticas

Ingresar al panel de administración https://[mi-dominio.gob.bo]/wp-admin/plugins.php y seleccionar todos los plugins.

Seleccionar la acción “habilitar actualizaciones automáticas” y ejecutar “Aplicar”.



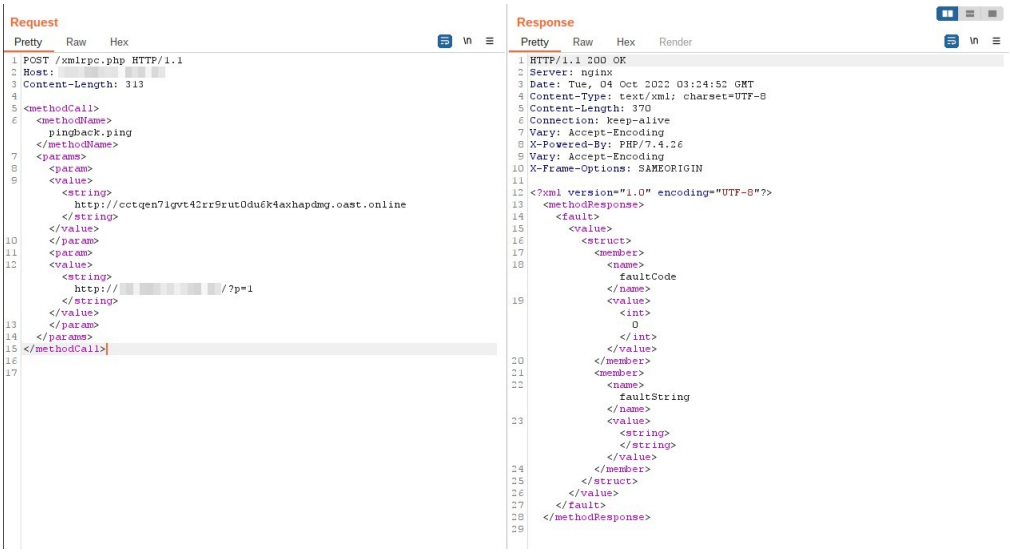
Borrar los themes no usados mediante la URL https://[mi-dominio.gob.bo]/wp-admin/themes.php

2.9. Deshabilitar XML-RPC

WordPress tiene características para interactuar de forma remota con el sitio web, XML- RPC es una función de WordPress que permite la transmisión de datos con HTTP actuando como mecanismo de transporte y XML como mecanismo de codificación.

En la actualidad la funcionalidad del archivo xmlrpc.php ha disminuido considerablemente y su exposición y configuración insegura representa un riesgo en la seguridad del sitio web, provocando las siguientes vulnerabilidades:

- Ataques de fuerza bruta.
- Denegación Distribuida de Servicio.
- XMLRPC pingback.ping



2.9.1. Restringir el acceso al archivo xmlrpc.php

Para restringir el acceso al archivo xmlrpc.php en apache2, se debe editar el archivo de configuración del sitio web o mediante el archivo .htaccess agregando las siguientes líneas:

<files xmlrpc.php>

order allow,deny

deny from all

</files>

Después reiniciar apache:

systemctlrestartapache2

2.9.2. Deshabilitar la función XML-RPC

Si se determina que la función XML-RPC no es necesaria, se recomienda deshabilitarla, para ello se debe editar el archivo wp-config.php y agregar la siguiente línea:

add_filter('xmlrpc_enabled','_return_false');

Otra opción para deshabilitar la función XML-RPC es instalando y activando el complemento “Disable XML-RPC”.

2.10. Restringir acceso al archivo de depuración

WordPress puede generar un archivo llamado *“debug.log”* con los logs del CMS sin embargo se debe configurar para que no sea accesible por cualquiera o en otro caso se puede deshabilitar esta función.

- Para deshabilitar esta función puede acceder al archivo *“wp-config.php”* y deshabilitar WP-DEBUG.
- Para restringir el acceso puede modificar el archivo .htaccess o a través de la configuración del servidor web.

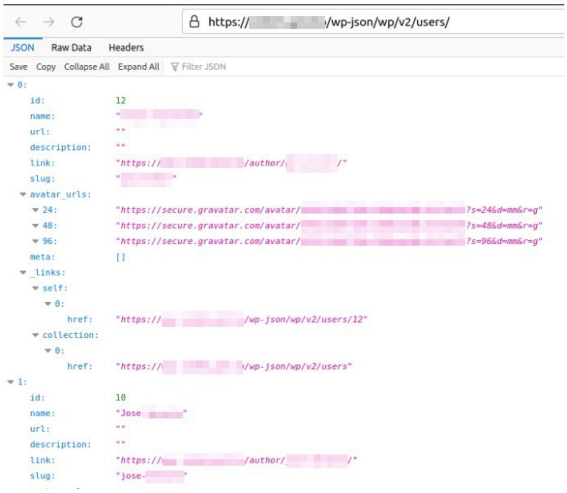
2.11. Restringir acceso al archivo wp-cron.php

Se recomienda deshabilitar o restringir el acceso al archivo *“wp-cron.php”* que podría generar sobrecarga en el servidor y la exposición de patrones internos de ejecución.

- Para deshabilitar esta función puede acceder al archivo *“wp-config.php”* y modificar el archivo a DISABLE_WP_CRON=true.
- Para restringir el acceso puede modificar el archivo .htaccess o a través de la configuración del servidor web.

2.12. Enumeración de usuarios

WordPress podría exponer datos sensibles (como: los usuarios del sistema), a través de su API REST nativa, si no está configurado de forma segura.



Para solucionar esta vulnerabilidad se debe actualizar a la versión 4.7.1 o posterior de WordPress.

Una opción, en caso de no hacer uso de la API REST de WordPress, es deshabilitarlo, para ello se debe agregar las siguientes líneas en el archivo de configuración de wordpress wp- config.php:

add_filter('rest_enabled','_return_false');

add_filter('rest_jsonp_enabled', '_return_false');

Asimismo se puede instalar el plugin *"REST API Toolbox"* que permitirá la deshabilitación de la API.

2.13. Ocultar la versión del CMS

La versión de la instancia de WordPress instalada podría ser utilizada para verificar si el CMS tiene vulnerabilidades conocidas o si existen exploits públicos que podrían ser usados para comprometer el sitio web.

Para ocultar la versión del CMS puede instalar el plugin *"WP Hide & Security Enhancer"*

ANEXO 3 - GUÍA TÉCNICA DE SEGURIDAD DRUPAL

1. INTRODUCCIÓN

Todo sistema de gestión de contenidos está propenso a ataques producto de vulnerabilidades descubiertas o configuraciones por defecto que se hayan dejado con la instalación. Por estas razones se presentan las prácticas de seguridad que debemos seguir para contar con un sitio web más seguro bajo Drupal.

2. ASEGURANDO DRUPAL

Para mitigar el riesgo de ataques a Drupal, recomendamos aplicar las siguientes buenas prácticas de seguridad.

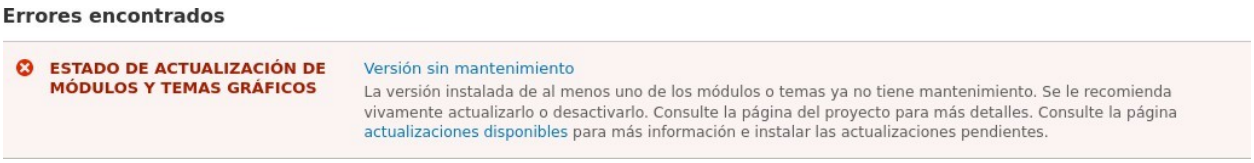
2.1. Activar notificaciones de actualizaciones de seguridad

Habilitar en la administración de Drupal notificaciones al correo electrónico sobre actualizaciones más recientes de seguridad de módulos contribuidos y del core.

Administrar > Informes > Actualizaciones disponibles, pestaña “configuración”

Revisar de forma periódica el informe de estado de Drupal donde se muestra el estado de cada módulo y del core sobre actualizaciones funcionales y de seguridad. Aquellos módulos que ya no cuentan con mantenimiento desde sus repositorios de origen se deben desactivar o migrar a uno con soporte.

Administración > Informes



Se recomienda seguir la cuenta de seguridad de Drupal en Twitter (@drupalsecurity) para estar al pendiente de las actualizaciones.

2.2. Instalar actualizaciones de seguridad

Como buena práctica de seguridad se debe tener un ambiente de pruebas donde se realicen las actualizaciones para verificar el correcto funcionamiento del sitio web antes de su pase a producción.

- Revisar el panel de actualizaciones disponibles.

- Replicar el ambiente de producción al ambiente de pruebas.
- Proceder con la actualización del core y módulos en el ambiente de pruebas.
- Verificar que las actualizaciones se hayan aplicado correctamente.
- Realizar las pruebas funcionales.
- En caso de que todo esté correcto, replicar el ambiente de pruebas a producción en un horario donde no afecte a los usuarios.

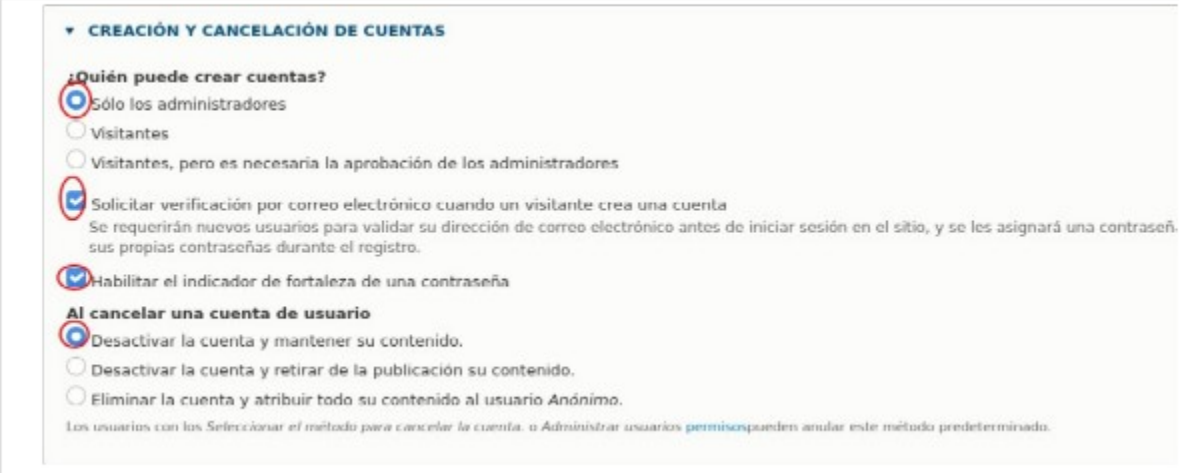
- Drupal publica actualizaciones de seguridad todos los miércoles y pueden ser revisadas a través de la página:

  https://www.drupal.org/security

2.3. Gestionar usuarios

En Drupal se encuentra habilitada por defecto la opción de creación de cuentas por usuarios anónimos. Una buena práctica de seguridad es deshabilitar esta opción.

Administración > Configuración > Usuarios > Configuración de la cuenta



2.3.1. Configurar permisos

- Crear nuevos roles de acuerdo a las necesidades funcionales del sitio, estableciendo permisos específicos en módulos instalados.

  Administración > Usuarios > Permisos

PERMISO	USUARIO ANÓNIMO	USUARIO AUTENTICADO	ADMINISTRADOR
de contenido.			
Revertir todas las revisiones Para revertir una revisión, también necesita permiso para editar el elemento de contenido.	☐	☐	☑
Ver todas las revisiones Para ver una revisión, también necesita permiso para ver el elemento de contenido.	☐	☑	
Ver contenido publicado	☑	☑	
Ver el contenido propio sin publicar	☐	☑	

2.4. Proteger formularios con captcha

El captcha previene intentos automatizados de inicios de sesión y envíos masivos (spam) de datos a través de formularios.

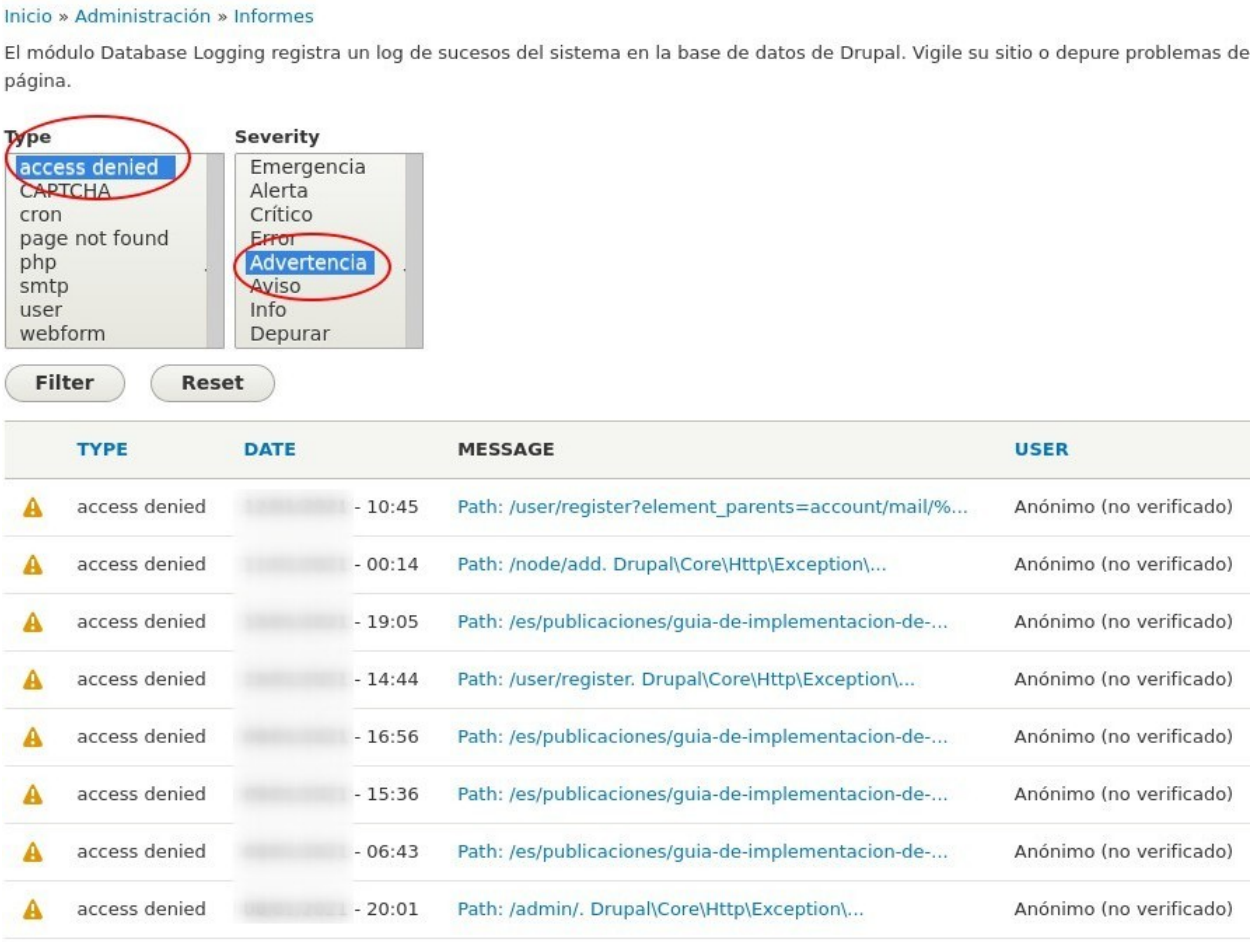
- Instalar el módulo captcha (https://www.drupal.org/project/captcha) y habilitar.
- Entre las configuraciones se puede establecer el tipo de desafío del captcha matemático, imagen o personalizada con preguntas y respuestas pre establecidas.

- Por ejemplo se habilita el captcha en el formulario de inicio de sesión, seleccionando el tipo de desafío.

2.5. Utilizar módulos con sello de seguridad

Sólo utilizar módulos que estén en fase estable y priorizar aquellos con el sello verde del Security Team de Drupal.

2.6. Revisar el registro reciente de mensajes

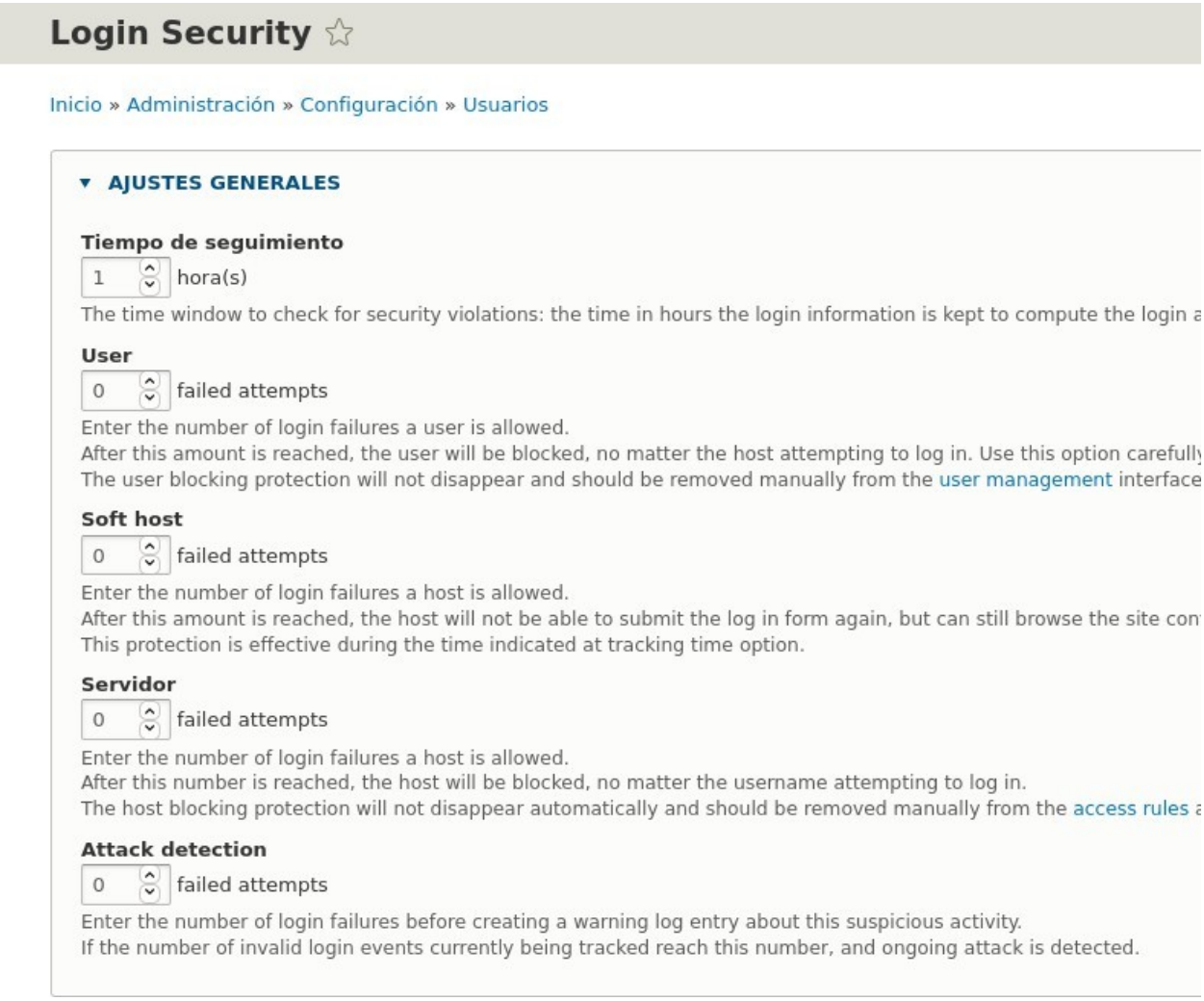


Revisar periódicamente el registro de mensajes de Drupal para identificar actividades sospechosas como intentos de inicio de sesión, errores de PHP, envío de formularios y otros datos importantes.

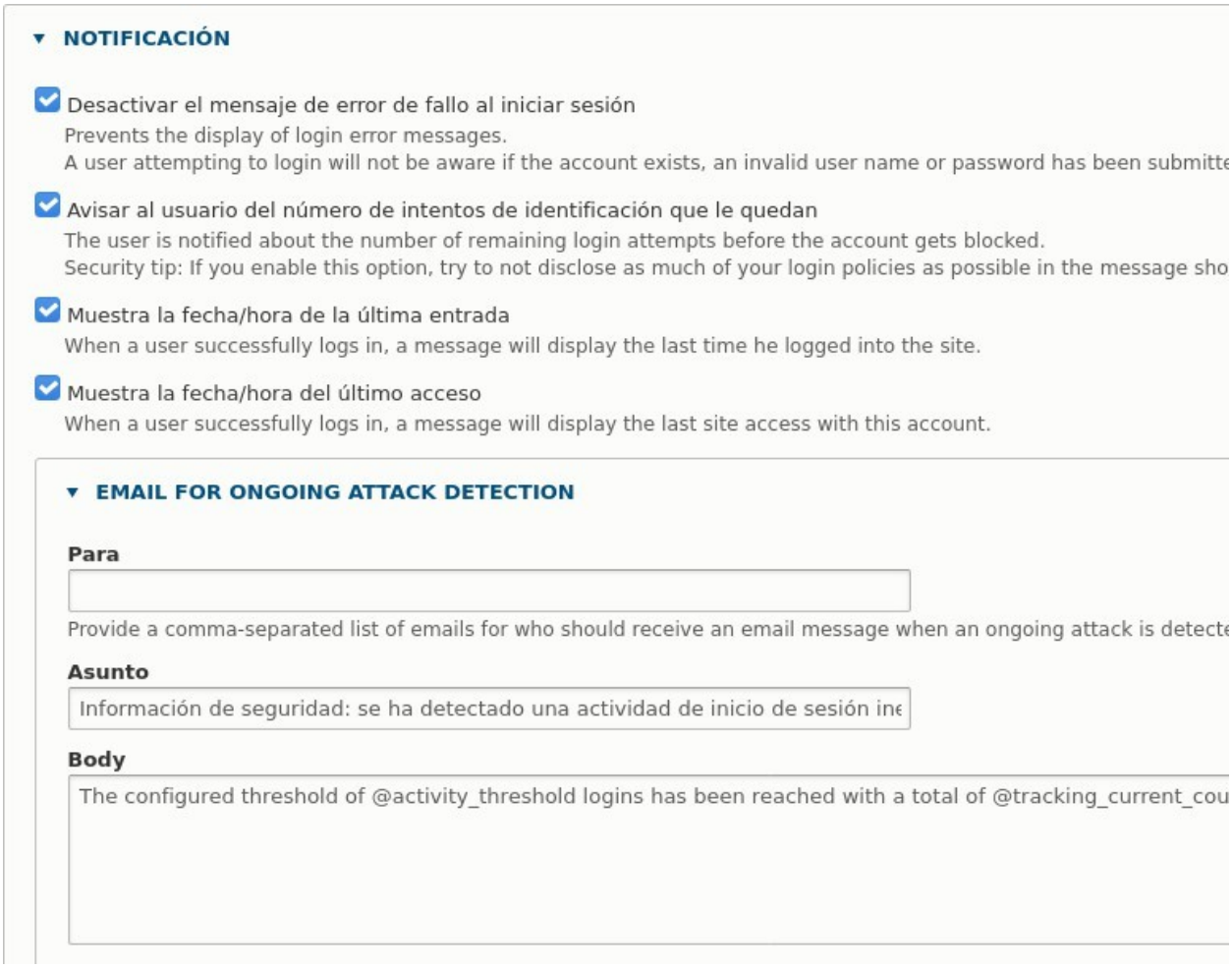
2.7. Módulo login security

El módulo Login Security permite configurar la forma en la que los usuarios se autentican en el sitio.

- Instalar el módulo login security (https://www.drupal.org/project/login_security) y habilitarlo simultáneamente con el módulo Ban que viene de forma predeterminada con el core de Drupal.
- Configurar en ajustes generales el número de intentos de inicio de sesión fallidos por usuario, host y también la detección de ataques.



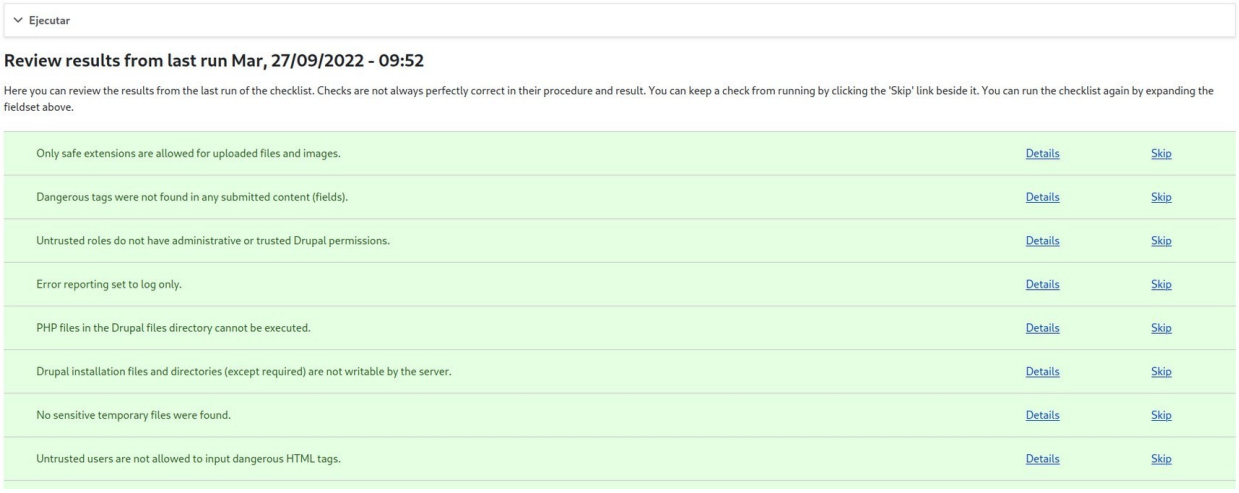
- Configurar notificaciones al correo electrónico, se recomienda deshabilitar los mensajes de error al iniciar sesión, avisar al usuario el número de intentos de inicio de sesión que le restan, mostrar la fecha y hora de último acceso.



2.8. Módulo security review

Este módulo prueba la configuración de Drupal en busca de vulnerabilidades de seguridad. Para aplicar este módulo se recomienda probar primero en el ambiente de pruebas.

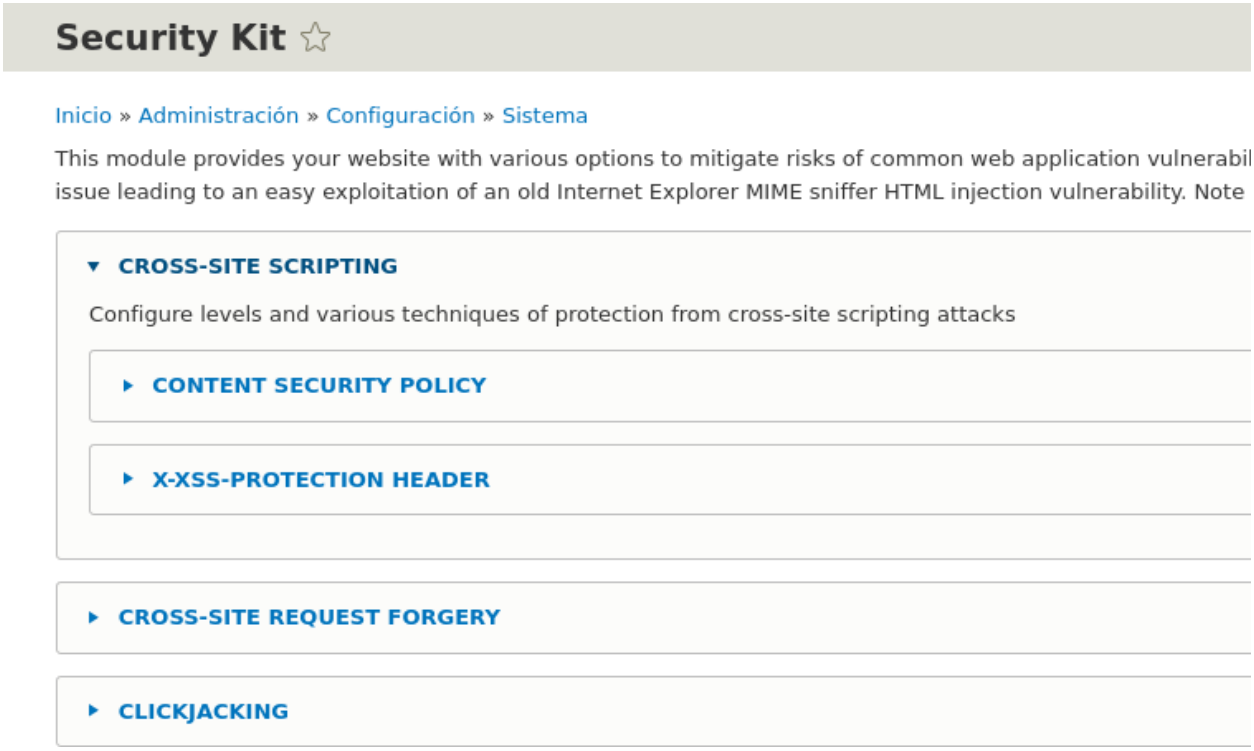
Instalar el módulo security review (https://www.drupal.org/project/security_review) y habilitarlo.



2.9. Módulo security kit

Security Kit permite proteger el sitio web de una amplia variedad de ataques como Cross- Site Scripting, Cross-Site Request Forgery, Clickjacking. Para aplicar este módulo se recomienda probar primero en el ambiente de pruebas.

- Instalar el módulo security kit (https://www.drupal.org/project/seckit) y habilitarlo.



- Configurar las opciones del módulo para mitigar riesgos de seguridad.

Si se desea ampliar las opciones de configuración de seguridad, se recomienda consultar los siguientes recursos:

- https://developer.mozilla.org/es/docs/Web/HTTP/CSP
- https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
- https://developer.mozilla.org/es/docs/Web/HTTP/Headers/X-Frame-Options
- https://developer.mozilla.org/es/docs/Web/HTTP/Headers/Strict-Transport-Security