Checklist de Seguridad para WordPress

Este documento presenta un checklist de seguridad para entornos WordPress, basado en las guías CIS Benchmark y documentación oficial de Apache, Nginx, PHP y WordPress. **Importante**: Siempre verificar primero en un entorno de prueba antes de producción.

#	Recomendación	Archivo / Lugar de configuración	Aplicado (✓)
1	Mantener WordPress actualizado a la última versión estable	Panel de administración de WordPress	
2	Mantener todos los plugins actualizados	Panel de administración de WordPress	
3	Mantener el theme activo y eliminar themes no usados o desactualizados	Panel de administración de WordPress	
4	Desactivar edición de archivos desde el admin (DISALLOW_FILE_EDIT)	wp-config.php	
5	Desactivar instalación/actualización de plugins y themes (DISALLOW_FILE_MODS)	wp-config.php	
6	Cambiar el prefijo de tablas de la base de datos ($table_prefix)	wp-config.php + Base de datos	
7	Desactivar XML-RPC si no se usa	functions.php o plugin	
8	Restringir acceso a REST API	functions.php o plugin	
9	Deshabilitar listados de directorios	.htaccess o config de nginx	
10	Ocultar versión de WordPress en el código fuente	functions.php	
11	Bloquear acceso a wp-config.php	.htaccess o config de nginx	
12	Proteger wp-content/uploads para evitar ejecución de PHP	.htaccess en uploads/	
13	Deshabilitar acceso a archivos sensibles	.htaccess o config de nginx	
14	Forzar HTTPS y usar HSTS	Config en Apache/nginx	
15	Usar certificados TLS válidos	Config servidor web	
16	Forzar HTTPS para wp-login.php	.htaccess o plugin	
17	Usar 2FA para usuarios admin	Plugin (Wordfence, WP 2FA)	
18	Usar contraseñas fuertes	Panel de usuario de WordPress	
19	Limitar intentos de login	Plugin Limit Login Attempts Reloaded	
20	Cambiar URL de acceso al panel admin	Plugin WPS Hide Login	
21	Eliminar o renombrar usuario "admin"	Panel de usuarios de WordPress	
22	Crear usuario temporal de rescate con rol administrador	Panel de usuarios	
23	Revisar y limpiar usuarios inactivos o con permisos excesivos	Panel de usuarios	
24	Restringir acceso a /wp-admin por IP	.htaccess o config de nginx	
25	Proteger wp-login.php con autenticación	.htaccess, .htpasswd	

Generación: Dr. Félix Alejandro Hernández Fuentes, Ing. Angie Aguilar Domínguez

	HTTP básica	
26	Deshabilitar modificación del archivo wp-config.php	Vía consola
27	Monitorizar actividad de usuarios (logins, cambios)	Plugin (WP Activity Log)
28	Hacer copias de seguridad regulares	Plugin (UpdraftPlus, Duplicator)
29	Guardar backups fuera del servidor (cloud, SFTP)	Configuración en el plugin de respaldo
30	Verificar restauración de backups periódicamente	Manual o automático mediante plugin
31	Usar plugin de seguridad con escaneo y alertas	Wordfence, Sucuri, iThemes
32	Revisar logs periódicamente	Hosting / plugin / servidor web
33	Cambiar claves de autenticación (AUTH_KEY, etc.) en wp-config.php	wp-config.php
34	Establecer permisos: 755 para carpetas y 644 para archivos	Vía FTP o consola
35	Propietario correcto: usuario del servidor web (www-data u otro)	Vía consola
36	Deshabilitar la firma del servidor	Apache/nginx config
37	Revisar php.ini y desactivar funciones peligrosas (exec, shell_exec)	Archivo php.ini
38	Limitar recursos (memory_limit, max_execution_time)	php.ini o config hosting
39	Usar cabeceras seguras: CSP, X-Frame-Options, etc.	Apache/nginx config o plugin
40	Restringir acceso a la REST API si no se usa	Plugin o functions.php
41	Limitar tamaño de archivos subidos	php.ini, WordPress, nginx
42	Evitar plugins inseguros o sin soporte	Revisión manual antes de instalar
43	Proteger cookies empleadas por Wordpress	wp-config.php
44	Mantener PHP actualizado a versión estable	Servidor/hosting
45	Usar versiones modernas de MySQL/MariaDB/PostgreSQL	Hosting / Servidor
46	No usar root como usuario de base de datos	Config MySQL/PostgreSQL
47	Cambiar puerto por defecto de la DB o limitar IPs	Config MySQL/PostgreSQL
48	Revisar endpoints personalizados en REST API	functions.php o plugin
49	Auditar plugins, themes y core regularmente	Manual + herramientas como WPScan
50	Eliminar plugins/themes no usados	Panel de administración de WordPress

Anexo A - Detalle de Configuración por Punto

Punto 1

Descripción: Mantener WordPress actualizado asegura que el núcleo reciba las últimas correcciones de seguridad y mejoras.

Objetivo: Proteger contra vulnerabilidades conocidas y mantener compatibilidad con plugins y temas.

Pasos:

1. Iniciar sesión en el panel de administración de WordPress como administrador.
2. Ir a Escritorio → Actualizaciones.
3. Si hay una nueva versión disponible, hacer clic en 'Actualizar ahora'.
4. Verificar que la actualización se haya completado sin errores.

Punto 2

Descripción: Mantener los plugins actualizados evita la explotación de vulnerabilidades conocidas en extensiones.

Objetivo: Reducir el riesgo de ataques que se aprovechen de plugins inseguros.

Pasos:

1. Acceder al panel de administración de WordPress.
2. Ir a Plugins → Plugins instalados.
3. Actualizar los plugins que tengan notificación de nueva versión.
4. Evitar usar plugins que no tengan soporte activo.

Punto 3

Descripción: Eliminar themes no usados reduce la superficie de ataque.

Objetivo: Evitar que un theme desactualizado sirva como vector de ataque.

Pasos:

1. En el panel de administración, ir a Apariencia → Temas.
2. Mantener solo el tema activo y uno de respaldo por seguridad.
3. Eliminar los demás temas.

Punto 4

Descripción: Desactivar la edición de archivos desde el administrador evita que atacantes modifiquen archivos críticos vía panel.

Objetivo: Reducir la superficie de ataque limitando modificaciones directas a los archivos del sitio.

Pasos:

1. Acceder al servidor vía SSH o panel de hosting.
2. Abrir el archivo wp-config.php ubicado en la raíz de WordPress.
3. Añadir la línea: `define('DISALLOW_FILE_EDIT', true);`
4. Guardar cambios y cerrar el archivo.

Punto 5

Descripción: Evitar instalación/actualización desde el panel previene cambios no autorizados.

Objetivo: Asegurar que los cambios se realicen solo por administradores vía servidor.

Pasos:

1. Editar el archivo wp-config.php.
2. Añadir la línea: `define('DISALLOW_FILE_MODS', true);`
3. Guardar cambios y salir.

Punto 6

Descripción: Cambiar el prefijo por defecto wp_ dificulta ataques automatizados a la base de datos.

Objetivo: Evitar inyecciones SQL dirigidas a tablas con nombres predecibles.

Pasos:

1. Editar wp-config.php y cambiar el valor: `$table_prefix = 'seguro_';`
2. Acceder a la base de datos y renombrar las tablas con el nuevo prefijo.
3. Actualizar referencias internas si es necesario.

Punto 7

Descripción: Desactivar XML-RPC evita abusos como ataques de fuerza bruta distribuidos.

Objetivo: Cerrar un vector de ataque común si no se usa la funcionalidad.

Pasos:

1. Editar el archivo functions.php del theme activo.
2. Añadir el código para desactivar XML-RPC: `add_filter('xmlrpc_enabled', '__return_false');`
3. Guardar cambios.

Punto 8

Descripción: Restringir REST API evita el acceso no autorizado a datos.

Objetivo: Limitar el uso de la API solo a usuarios autenticados o funciones específicas.

Pasos:

1. Editar functions.php o instalar un plugin de seguridad.
2. Configurar la API para permitir solo accesos autorizados.

Ejemplo de configuración/código:

```
add_filter('rest_authentication_errors', function($result) { if (!is_user_logged_in()) { return new WP_Error('rest_not_logged_in', 'Debes iniciar sesión', ['status' => 401]); } return $result; });
```

Punto 9

Descripción: Deshabilitar listados de directorios evita que atacantes vean la estructura de archivos.

Objetivo: Ocultar información sensible de la estructura interna del sitio.

Pasos:

1. Editar el archivo .htaccess o la configuración de Nginx.
2. Añadir la directiva para desactivar listados.
3. Guardar y reiniciar el servidor.

Ejemplo de configuración/código:

```
Options -Indexes
```

Punto 10

Descripción: Ocultar versión de WordPress reduce información útil para atacantes.

Objetivo: Prevenir que scripts automatizados detecten versiones vulnerables.

Pasos:

1. Editar functions.php.
2. Añadir el filtro para eliminar la versión: `remove_action('wp_head', 'wp_generator');`
3. Guardar cambios.

Punto 11

Descripción: Evita que usuarios no autorizados descarguen o vean el archivo wp-config.php.

Objetivo: Proteger credenciales y configuraciones críticas.

Pasos:

1. Abrir el archivo .htaccess en la raíz de WordPress.
2. Añadir el siguiente código y guardar cambios:

```
<files wp-config.php>
    order allow,deny
    deny from all
</files>
```

Punto 12

Descripción: Impide que se ejecute código PHP malicioso en la carpeta de cargas (o cualquier otra donde no sea necesario).

Objetivo: Proteger contra cargas maliciosas de archivos.

Pasos:

1. Crear o editar .htaccess dentro de wp-content/uploads (o la carpeta del sitio que se desea proteger).

2. Añadir el siguiente contenido:

```
<FilesMatch \.php$>
    deny from all
</FilesMatch>
```

Punto 13

Descripción: Evita el acceso a archivos como readme.html y license.txt.

Objetivo: Ocultar información que pueda dar pistas a atacantes.

Pasos:

1. En .htaccess añadir reglas para negar el acceso a esos archivos:

```
<files readme.html>
    order allow,deny
    deny from all
</files>
```

Punto 14

Descripción: Forzar HTTPS y habilitar HSTS mejora la seguridad de la conexión.

Objetivo: Proteger la transmisión de datos contra ataques de intermediarios.

Pasos:

1. En Apache: usar `Redirect permanent / https://tusitio.com/`
2. Añadir cabecera Strict-Transport-Security:

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

Punto 15

Descripción: Certificados válidos garantizan una conexión segura y confiable.

Objetivo: Evitar advertencias de seguridad en navegadores.

Pasos:

1. Usar Let's Encrypt o una autoridad certificadora reconocida.
2. Configurar el certificado en Apache o Nginx.

Punto 16

Descripción: Asegura que la página de login siempre use HTTPS.

Objetivo: Proteger credenciales durante la autenticación.

Pasos:

1. En .htaccess añadir redirección para wp-login.php.
2. O bien, configurar el plugin para forzar HTTPS.

Punto 17

Descripción: Autenticación en dos pasos añade una capa extra de seguridad.

Objetivo: Reducir el riesgo de acceso no autorizado.

Pasos:

1. Instalar un plugin como WP 2FA o Wordfence.
2. Configurar el método `(app, correo, SMS).`

Punto 18

Descripción: Contraseñas fuertes dificultan ataques de fuerza bruta.

Objetivo: Reducir la probabilidad de que credenciales sean adivinadas.

Pasos:

1. Usar, por ejemplo, al menos 12 caracteres con letras, números y símbolos (seguir la política de su dependencia).
2. Evitar contraseñas usadas en otros sitios.

Punto 19

Descripción: Limita los intentos fallidos de login para prevenir ataques de fuerza bruta.

Objetivo: Bloquear direcciones IP que intenten adivinar contraseñas.

Pasos:

1. Instalar el plugin Limit Login Attempts Reloaded.
2. Configurar, por ejemplo, máximo 5 intentos en 10 minutos (seguir la política de su dependencia).

Punto 20

Descripción: Cambia la URL del login para ocultarla a bots y atacantes.

Objetivo: Reducir intentos automáticos de acceso.

Pasos:

1. Instalar plugin WPS Hide Login.
2. Configurar nueva URL de acceso y guardarla.

Punto 21

Descripción: Elimina o renombra el usuario por defecto “admin”.

Objetivo: Reducir el riesgo de ataques de fuerza bruta sobre un usuario común.

Pasos:

1. Iniciar sesión con otro usuario administrador.
2. Crear un nuevo usuario administrador con nombre distinto.
3. Eliminar o cambiar rol del usuario "admin".

Punto 22

Descripción: Crear un usuario alterno con permisos de administrador para emergencias.

Objetivo: Garantizar acceso en caso de bloqueo de la cuenta principal.

Pasos:

1. Ir a Usuarios → Añadir nuevo.
2. Asignar rol de administrador.
3. Guardar credenciales en lugar Seguro (se puede usar un gestor de contraseñas).

Punto 23

Descripción: Revisar la lista de usuarios y permisos periódicamente.

Objetivo: Minimizar la superficie de ataque eliminando cuentas innecesarias.

Pasos:

1. Revisar último acceso de cada usuario.
2. Eliminar cuentas inactivas.
3. Ajustar roles y permisos si son excesivos.

Punto 24

Descripción: Limitar acceso solo desde direcciones IP autorizadas.

Objetivo: Bloquear intentos desde ubicaciones no confiables.

Pasos:

1. Editar .htaccess en /wp-admin.
2. Añadir directivas de restricción de IP:

```
Order deny,allow
deny from all
allow from 192.168.1.100 (Dirección IP de ejemplo)
```

Punto 25

Descripción: Requerir autenticación adicional antes de la pantalla de login.

Objetivo: Aumentar la dificultad para ataques automatizados.

Pasos:

1. Crear .htpasswd con usuario y contraseña.
2. Configurar .htaccess para proteger wp-login.php.

Punto 26

Descripción: Cambiar permisos en archivo wp-config.php

Objetivo: Evitar modificaciones del contenido del archivo.

Pasos:

1. Desde la línea de comandos ejecutar:

```
chmod 400 wp-config.php
```

Punto 27

Descripción: Llevar un registro de logins y cambios.

Objetivo: Detectar accesos sospechosos.

Pasos:

1. Instalar un plugin como WP Activity Log.
2. Configurar alertas por email o panel.

Punto 28

Descripción: Realizar backups automáticos.

Objetivo: Garantizar la recuperación en caso de incidente.

Pasos:

1. Instalar plugin como UpdraftPlus.
2. Configurar frecuencia (diaria, semanal, según la política de su dependencia).

Punto 29

Descripción: Almacenar copias de seguridad en un medio externo.

Objetivo: Evitar pérdida de datos si el servidor falla.

Pasos:

1. Configurar plugin de backups para usar Google Drive, Dropbox o SFTP.
2. Probar conexión y subida.

Punto 30

Descripción: Probar que los respaldos funcionen correctamente.

Objetivo: Garantizar la integridad y disponibilidad de la copia.

Pasos:

1. Restaurar un backup en un entorno de prueba.
2. Verificar que todo el sitio funcione.

Punto 31

Descripción: Implementar un plugin que analice el sitio y alerte sobre vulnerabilidades.

Objetivo: Detectar y prevenir amenazas activas.

Pasos:

1. Instalar plugin como Wordfence o Sucuri Security.

2. Configurar escaneos automáticos y notificaciones por correo al responsable del aplicativo.

Punto 32

Descripción: Inspeccionar registros de actividad y errores.

Objetivo: Identificar intentos de ataque o problemas técnicos.

Pasos:

1. Revisar logs desde cPanel, Plesk o consola (/var/log/apache2/ o /var/log/nginx/, ruta y nombre de archivo donde se almacenan las bitácoras).
2. Analizar errores repetitivos o accesos sospechosos (códigos de estado HTTP 500, 404, solicitud de recursos en orden alfabético, entre otros).

Punto 33

Descripción: Regenerar las claves de seguridad usadas por WordPress.

Objetivo: Invalidar cookies y sesiones activas comprometidas.

Pasos:

1. Obtener nuevas claves en https://api.wordpress.org/secret-key/1.1/salt/.
2. Sustituir en wp-config.php las líneas `AUTH_KEY`, `SECURE_AUTH_KEY`, `LOGGED_IN_KEY`, `NONCE_KEY` y sus SALTs.

Punto 34

Descripción: Configurar permisos mínimos necesarios para WordPress.

Objetivo: Prevenir modificaciones no autorizadas.

Pasos:

1. Conectar por SSH o FTP al servidor.
2. Ejecutar:

```
find /ruta/a/wordpress/ -type d -exec chmod 755 {} \;
find /ruta/a/wordpress/ -type f -exec chmod 644 {} \;
```

Punto 35

Descripción: Asegurar que los archivos pertenezcan al usuario que corre Apache/Nginx.

Objetivo: Evitar problemas de permisos y seguridad.

Pasos:

1. Identificar usuario del servicio web (www-data en Debian/Ubuntu).
2. Cambiar la propiedad:

```
chown -R www-data:www-data /ruta/a/wordpress/
```

Punto 36

Descripción: Deshabilitar la firma del servidor.

Objetivo: Evitar la exposición de la versión empleada de Apache/Nginx.

Pasos:

1. Editar el archivo de configuración (Apache/Nginx):

```
ServerTokens prod;
ServerSignature off;

server_tokens off;
```

Punto 37

Descripción: Limitar funciones PHP que puedan ser usadas maliciosamente.

Objetivo: Reducir riesgo de ejecución de comandos en el servidor.

Pasos:

1. Editar el archivo de configuración php.ini.
2. Modificar:

```
disable_functions =
exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,pars
e_ini_file,show_source
```

Punto 38

Descripción: Establecer límites de memoria y tiempo de ejecución.

Objetivo: Evitar ataques de denegación de servicio.

Pasos:

1. Editar el archivo de configuración php.ini o configuración de hosting.
2. Ajustar:

```
memory_limit = 256M
max_execution_time = 30
```

Punto 39

Descripción: Configurar cabeceras HTTP que mejoren la seguridad.

Objetivo: Prevenir ataques como XSS, clickjacking y MITM.

Pasos:

1. Editar configuración de Apache o Nginx.
2. Añadir:

```
Header set X-Content-Type-Options "nosniff"
Header set X-Frame-Options "SAMEORIGIN"
```

```
Header set Content-Security-Policy "default-src 'self'"
```

Punto 40

Descripción: Deshabilitar acceso si no se usa o limitar a usuarios autenticados.

Objetivo: Reducir exposición de datos sensibles.

Pasos:

1. Instalar plugin como Disable REST API.
2. O añadir a functions.php:

```
add_filter('rest_authentication_errors', function ($result) {
  if (!is_user_logged_in()) {
    return new WP_Error('rest_cannot_access', __('Only authenticated users can
access the REST API.'), array('status' =>
rest_authorization_required_code()));
  }
  return $result;
});
```

Punto 41

Descripción: Establecer un límite máximo de subida de archivos para prevenir abusos.

Objetivo: Evitar uso excesivo de almacenamiento y ataques de denegación de servicio.

Pasos:

1. Editar php.ini y modificar:

```
upload_max_filesize = 10M
post_max_size = 12M
```

2. Si se usa Nginx:

```
client_max_body_size 12M;
```

3. Guardar y reiniciar servidor web.

Punto 42

Descripción: No instalar plugins sin mantenimiento o con vulnerabilidades conocidas.

Objetivo: Reducir superficie de ataque.

Pasos:

1. Revisar compatibilidad y fecha de última actualización antes de instalar.
2. Verificar en https://wpscan.com/plugins.

Punto 43

Descripción: Proteger las cookies empleadas.

Objetivo: Asignar las banderas en las cabeceras empleadas.

Pasos:

1. Editar archivo de configuración wp-config.php y añadir:

```
@ini_set('session.cookie_httponly', true);
@ini_set('session.cookie_secure', true);
@ini_set('session.use_only_cookies', true;
```

Punto 44

Descripción: Usar versiones soportadas de PHP para seguridad y rendimiento.

Objetivo: Evitar vulnerabilidades conocidas.

Pasos:

1. Verificar versión con:

```
php -v
```

2. Actualizar vía repositorio oficial o gestor de paquetes de la distribución empleada.

Punto 45

Descripción: Mantener el motor de base de datos actualizado.

Objetivo: Garantizar compatibilidad y seguridad.

Pasos:

1. Verificar versión (según el manejador empleado):

```
mysql --version
psql --version
```

2. Actualizar desde repositorio oficial de acuerdo con la documentación correspondiente.

Punto 46

Descripción: Usar un usuario con privilegios mínimos necesarios.

Objetivo: Minimizar daños en caso de compromiso.

Pasos:

1. Crear usuario con permisos limitados en MySQL:

```
CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'contraseña';
GRANT ALL PRIVILEGES ON basededatos.* TO 'wpuser'@'localhost';
FLUSH PRIVILEGES;
```

Punto 47

Descripción: Evitar ataques automatizados al puerto estándar.

Objetivo: Reducir probabilidad de acceso no autorizado.

Pasos:

1. Para MySQL, editar el archivo /etc/mysql/my.cnf y cambiar `port=3307`.

2. Limitar acceso a localhost: `bind-address = 127.0.0.1`

3. Para el caso de PostgreSQL, editar el archivo pg_hba.conf

Punto 48

Descripción: Analizar endpoints creados por plugins o código propio.

Objetivo: Detectar datos expuestos innecesariamente.

Pasos:

1. Usar WP REST API Inspector o herramientas como Postman.
2. Bloquear endpoints no utilizados en functions.php.

Punto 49

Descripción: Revisar el código y estado de seguridad del sitio.

Objetivo: Detectar vulnerabilidades antes de que sean explotadas.

Pasos:

1. Al contar con la herramienta wpscan, ejecutar:

```
wpscan --url https://tusitio.com
```

2. Revisar reportes y actualizar elementos vulnerables

Punto 50

Descripción: Reducir la superficie de ataque.

Objetivo: Minimizar el número de posibles vulnerabilidades.

Pasos:

1. Desde el panel de WordPress, ir a Plugins → Plugins instalados.
2. Desactivar y borrar los que no se usen.

Referencias

Center for Internet Security. (2023). CIS WordPress Benchmark v1.0.0. https://www.cisecurity.org/benchmark/wordpress
The Apache Software Foundation. (2024). Apache HTTP Server Documentation. https://httpd.apache.org/docs/
NGINX, Inc. (2024). NGINX Documentation. https://nginx.org/en/docs/
The PHP Group. (2024). PHP Manual. https://www.php.net/manual/en/
WordPress Foundation. (2024). WordPress Codex. https://codex.wordpress.org/