

WordPress, webinar seguridad

- Nominalia: <https://www.escueladeinternet.com/>
- <https://www.escueladeinternet.com/seguridad-wordpress-hackeo>

vulnerabilidades

- en themes o plugins
- en núcleo WP
- PHP → versión > 8.1
- robo credenciales/suplantación
- fugas de información (logs de error que exponen información sensible)

Amenazas

- Malware y puertas traseras
- DDOS y fuerza bruta
- inyección SQL
- XSS
- subidas de ficheros no controladas
- secuestro de sesiones (no https o mal configurado)
- redirecciones maliciosas
- ataques por IA

prácticas recomendadas

- Actualizar core, plugins, temas...
 - eliminar aquellos que no se usan (el código PHP está ahí)
 - no instalar plugins abandonados (más de 9 meses sin actualizarse)
- Instalar de sitios oficiales o confiables
- PHP 8.1
- acceso seguro
 - limitar número de accesos erróneos
 - 2FA
 - cambiar usuario «admin»
 - contraseñas largas y únicas
 - roles apropiados
- sistema de archivos
 - 664 archivos
 - 600 wp-config.php
 - 400 .htaccess
 - 755 carpetas
 - desactivar feed rss y atom
 - desactivar XML-RPC
 - ocultar versión WP
 - borrar archivos de identificación de WP (readme, license)
 - deshabilitar la exploración de directorios
- Debug y log de errores
 - desactivar por defecto
 - wp-config.php

- .htaccess
- Copias de seguridad
 - diarios
 - guardar fuera del servidor (sobre todo cambios importantes)
 - algún plugin para copia interna de restauración rápida
- Monitorizar
 - algún plugin de activity log
 - alertas por mail para acciones críticas
 - análisis malware y cambio de archivos
 - escanear semanalmente
- Otros:
 - cambiar prefijo BDD
 - cabeceras de seguridad
 - deshabilitar enumeración de usuarios
 - themeforest + envato (para tener temas actualizados de ese sitio de temas. Hay que vincular con la cuenta que compró el tema)

ejemplo práctico

análisis

- Herramientas → Salud del sitio → Información
 - Wordpress
 - versión
 - Tema activo
 - versión
 - Servidor
 - versión PHP
 - versión servidor web
- plugin **WordFence**
 - desactivar en Opciones generales de WordFence → **pausar actualizaciones en vivo cuando la ventana pierde el foco**
 - opciones de exploración y planificación → alta sensibilidad
 - iniciar nueva exploración → listado de vulnerabilidades

actualizaciones

- borrar caché antes de actualizar para detectar errores producidos por una actualización (WP Fastest Cache)
- activar plugin de mantenimiento (WP Maintenance)
- Escritorio → actualizaciones
 - plugins (no actualizar todas de golpe, dificulta localizar un error)
 - temas
 - núcleo WP

corrección vulnerabilidades

- Eliminar o sustituir abandonados o no en uso
- Planificar sustitución
- plugin sin soporte:
 - alternativas equivalentes
 - buscar solución a través de IA (claude.ai, por ejemplo)

securización

- Limitar el número de accesos erróneos
 - .htaccess (requiere IP Fija en el cliente)
 - WordFence
 - protección contra ataques de fuerza bruta
- 2FA activo
 - WordFence tiene
- Cambiar «admin»
 - plugin: <https://wordpress.org/plugins/change-username/>
 - manual:
 - crear nuevo
 - eliminar aniguo, asignar el contenido a otro usuario
- Desactivar XML-RPC-API
 - <https://wordpress.org/plugins/disable-xml-rpc-api/> (Neatma (Amin Nazemi))
 - .htaccess:

```
<Files xmlrpc.php
Order Deny,Allow
Deny from all
</Files>
```

- vinculado al tema:

[functions.php](#)

```
// Desactivar XML-RPC completamente
add_filter('xmlrpc_enabled', '__return_false');
// Eliminar headers de XML-RPC
remove_action('wp_head', 'rsd_link');
remove_action('wp_head', 'wlwmanifest_link');
```

- contraseñas largas y únicas
 - WordFence
- Asignar roles apropiados
 - plugin (para necesidades especiales): <https://es.wordpress.org/plugins/user-role-editor/>
- Desactivar feeds y atom
 - plugin <https://es.wordpress.org/plugins/disable-feeds-wp/>
- Desactivar enumeración de usuarios
 - WordFence
- Desactivar logs de WP y PHP

- .htaccess

```
php_flag display_errors Off
php_flag display_startup_errors Off
php_flag log_errors Off
php_value error_reporting 0
```

- wp-config.php

```
define('WP_DEBUG', false);
```

```
define('WP_DEBUG_LOG', false);  
define('WP_DEBUG_DISPLAY', false);  
@ini_set('display_errors', 0);
```

- en caso de necesitar el fichero debug.log:

[.htaccess](#)

```
# Proteger debug.log  
<Files debug.log>  
Order allow,deny  
Deny from all  
</Files>
```

- Ocultar versión de WP
 - WordFence
- Borrar ficheros de identificación
 - Manual:
 - readme.html
 - licente.txt
 - licencia.txt
 - wp-config-sample.php
 - Bloqueo en .htaccess:

```
<FilesMatch "^(readme\.html|license\.txt|licencia\.txt|wp-config-  
sample\.php)$"> Order Allow,Deny  
Deny from all  
</FilesMatch>
```

- Plugin: <https://es.wordpress.org/plugins/wp-hide-security-enhancer/>
- Desactivar exploración de directorios
 - buscar en Google: site:<URL> intitle:<index of>
 - WordFence

- [.htaccess](#)

```
Options -Indexes
```

- Cabeceras de seguridad (contra XSS)

- [.htaccess](#)

```
<IfModule mod_headers.c>  
# -----  
#  SEGURIDAD BÁSICA DE NAVEGADOR  
# -----  
# Evita que el navegador intente adivinar tipos MIME  
Header always set X-Content-Type-Options "nosniff"  
# Previene ataques de clickjacking  
Header always set X-Frame-Options "SAMEORIGIN"  
# Referrer-Policy (limita la información enviada en el encabezado  
Referer)  
Header always set Referrer-Policy "strict-origin-when-cross-origin"
```

```
# Obliga a usar HTTPS (solo si tu sitio usa SSL)
Header always set Strict-Transport-Security "max-age=31536000;
includeSubDomains; preload"
# -----
# PERMISSIONS POLICY (ANTES Feature-Policy)
# Controla las APIs y características disponibles en el navegador
# -----
Header always set Permissions-Policy "accelerometer=(), camera=(),
geolocation=(), gyroscope=(), microphone=(), payment=(), usb=(),
fullscreen=(self)"
# -----
# CONTENT SECURITY POLICY (CSP)
# Define qué recursos puede cargar el navegador.
# Ajusta las URLs de scripts, estilos, imágenes, etc.
# -----
Header always set Content-Security-Policy "
default-src 'self';
script-src 'self' 'unsafe-inline' 'unsafe-eval' https;;
style-src 'self' 'unsafe-inline' https;;
img-src 'self' data: https;;
font-src 'self' https: data;;
connect-src 'self' https;;
frame-ancestors 'self';
base-uri 'self';
form-action 'self';
object-src 'none';
upgrade-insecure-requests;
"
</IfModule>
```

- Plugin: <https://es.wordpress.org/plugins/http-headers/>

Desinfección

- contención
- backup
- desinfección

contención

- modo mantenimineto
- cambiar todas las contraseñas
 - wordpress admins
 - BDD
 - FTP
- cerrar todas las sesiones activas
 - regenerar las claves de seguridad wp-config.php: https://api.wordpress.org/secret_key/1.1/salt/

backup

- en el estado en que esté...
- ficheros, BDD

- Updraft Plus, WP Vivid

desinfección

- restaurar ficheros originales del Core
- 2 plugins para limpiar:
 - WordFence
 - GOTMLS
- caso web no funcional
 - comprobar versión WP exacta: /wp-includes/version.php → \$wp_version = «X.X.X»
 - descargar la misma versión desde <https://ca.wordpress.org/download/releases/>
 - borrar todo excepto:
 - carpeta wp-content
 - ficheros wp-config.php y .htaccess
 - subir la versión limpia
 - revisar los ficheros que no hemos eliminado:
 - ejemplos .htaccess:

.Fragmentos de código malicioso que podemos encontrar

```
# =====
# EJEMPLO 1: Redirección maliciosa oculta
# Redirige tráfico de buscadores a sitios de spam
# =====
# CÓDIGO MALICIOSO
RewriteEngine On
RewriteCond %{HTTP_REFERER} ^.*google.*$ [NC,OR]
RewriteCond %{HTTP_REFERER} ^.*yahoo.*$ [NC,OR]
RewriteCond %{HTTP_REFERER} ^.*bing.*$ [NC]
RewriteRule ^(.*)$ http://spam-pharma-site.com/$1 [R=301,L]

# =====
# EJEMPLO 2: Backdoor con auto_prepend_file
# Inyecta código PHP en TODAS las páginas
# =====
# CÓDIGO MALICIOSO
php_value auto_prepend_file "/tmp/malicious.php"
php_value auto_append_file "/var/www/backdoor.php"

# =====
# EJEMPLO 3: Redirección condicional por User-Agent
# Muestra contenido diferente a bots vs usuarios
# =====
# CÓDIGO MALICIOSO
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{HTTP_USER_AGENT} (google|googlebot|bingbot) [NC]
RewriteRule ^(.*)$ http://malicious-seo-site.com/doorway.php?page=$1 [L]
</IfModule>

# =====
# EJEMPLO 4: Bloqueo de herramientas de seguridad
# Impide que plugins de seguridad funcionen
# =====
# CÓDIGO MALICIOSO:
<FilesMatch "(wordfence|sucuri|ithemes|malcare)">
Order Allow,Deny
Deny from all
</FilesMatch>

# =====
# EJEMPLO 5: Ocultación de archivos maliciosos
# Permite acceso a shells sin ser detectados
# =====
# CÓDIGO MALICIOSO:
<FilesMatch "^(shell|wso|c99|r57|b374k)\.php$">
Order Allow,Deny
Allow from all
Satisfy Any
</FilesMatch>
```

45

- caso web funcional
 - Usar plugins para desinfectar wp-content
 - WordFence
 - Opciones Avanzadas de exploración → asegurarnos que no hay exclusiones (si ya estaba instalado)
 - GOTMLS (mejor escaneando BDD)
 - revisar snippets
 - plugin «Fragmentos de código» o equivalentes
 - nombres de archivo en minúsculas
 - desactivar barra navegador
 - activar emoticonos
 - Año actual
 - Temas
 - personalizar JS
 - Widgets: Apariencia → Widgets

plugins recomendados

- WordFence (seguridad)
- GOTMLS (seguridad)
- WP Maintenance (Modo mantenimiento)
- Disable XML-RPC-API
 - Per Neatma (Amin Nazemi)
- NS Cloner - Site Copier

From:

<https://matebcn.eu/> - **miguel angel torres egea**

Permanent link:

<https://matebcn.eu/web:security:wordpress:seguridad>

Last update: **22/10/2025 09:15**

