Caballo de troya: Vulnerabilidades en WordPress

Javier Casares
profiles.w.org/javiercasares

#WCBCN

2023 WC BCN

Equipos

Core *make.w.org/core*

- Se encarga del código del núcleo de WordPress y de herramientas, plugins y temas.

  - Seguridad el código.

Hosting *make.w.org/hosting*

- Se encarga de la relación de la infraestructura y hosting con las empresas de alojamiento.

  - Seguridad de los servidores.

Documentación *make.w.org/docs*

- Se encarga de la documentación de todo lo relacionado con seguridad. En colaboración con todos los equipos.

  - Documentación sobre seguridad.

Seguridad *make.w.org/security*

- Se encarga de recibir y validar posibles vulnerabilidades del núcleo de WordPress y algunos de los complementos.

  - Supervisión y validación.

Tipos de Vulnerabilidades

OWASP

En general se dividen las vulnerabilidades en un ranking de TOP 10 (OWASP Top Ten).

https://owasp.org/www-project-top-ten/

Cuando hablamos de WordPress nos centramos, en general, en sólo 4 de ellas.

Tipos de Vulnerabilidades

A1 - Inyección SQL

Es un ataque en el que un atacante aprovecha una vulnerabilidad en una aplicación web que utiliza SQL para interactuar con la base de datos. Los atacantes pueden utilizar técnicas de inyección SQL para engañar a la aplicación para que ejecute código malicioso, lo que puede resultar en la divulgación de información confidencial o en la modificación o eliminación de datos importantes.

RESUMEN: Intentar colar basura en la base de datos.

A1 - Inyección SQL

```
<?php
// Parámetro de la URL que identifica el producto
$id_producto = $_GET['id'];

//Construir y ejecutar la consulta SQL
$sql = "SELECT nombre, descripcion FROM productos WHERE id =
$id_producto";
```

A1 - Inyección SQL

```php
<?php
// Parámetro de la URL que identifica el producto
$id_producto = intval($_GET['id']);

//Construir y ejecutar la consulta SQL
$sql = "SELECT nombre, descripcion FROM productos WHERE id =
$id_producto";
```

Tipos de Vulnerabilidades

A7 - Cross-site scripting (XSS)

Ocurre cuando un atacante utiliza una página web para inyectar código malicioso en el navegador de un usuario. Los atacantes pueden utilizar técnicas de XSS para robar información confidencial o para redirigir a los usuarios a sitios web maliciosos.

RESUMEN: Intentar colar basura a través de un formulario.

A7 - Cross-site scripting (XSS)

```php
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  $mensaje = $_POST['mensaje'];
  echo "<p>Este es el mensaje que nos has enviado:</p>";
  echo "<p>" . $mensaje . "</p>";
}
```

A7 - Cross-site scripting (XSS)

```php
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  $mensaje = htmlspecialchars($_POST['mensaje']);
  echo "<p>Este es el mensaje que nos has enviado:</p>";
  echo "<p>" . $mensaje . "</p>";
}
```

Tipos de Vulnerabilidades

A8 - Cross-site request forgery (CSRF)

Es un ataque en el que un atacante engaña a un usuario para que realice una acción no deseada en un sitio web. Por ejemplo, un atacante puede enviar un correo electrónico malicioso que parezca legítimo, pero que en realidad contiene un enlace a un sitio web que realiza acciones maliciosas en el navegador del usuario.

RESUMEN: WordPress incluye el sistema de “nonces” por algo.

A8 - Cross-site request forgery (CSRF)

```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  if (isset($_SESSION['usuario'])) {
    $usuario = $_SESSION['usuario'];
    $contrasena = $_POST['contrasena’];
    echo "Contraseña actualizada correctamente.";
  }
}
```

A8 - Cross-site request forgery (CSRF)

```php
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  if (isset($_SESSION['usuario']) && isset($_POST['csrf_token'])
&& $_POST['csrf_token'] === $_SESSION['csrf_token']) {
    $usuario = $_SESSION['usuario'];
    $contrasena = $_POST['contrasena’];
    echo "Contraseña actualizada correctamente.";
  }
}
$_SESSION['csrf_token'] = bin2hex(random_bytes(32));
```

Tipos de Vulnerabilidades

A3 - Exposición de datos sensibles

La exposición de datos sensibles ocurre cuando los datos confidenciales se almacenan o se transmiten de manera insegura. Esto puede permitir a los atacantes acceder a información privada, como contraseñas o números de tarjetas de crédito.

Datos de Vulnerabilidades (por tipo)

	2020	2021	2022
Vulnerabilidades	≈600	≈1.400	≈4.500
A7 - XSS	44%	50%	27%
A1 - Inyección	15%	7%	5%
A8 - CSRF	8%	11%	29%
A3 - Datos sensibles	6%	2%	21%
Otras	27%	30%	18%

Fuentes: WPVulnerability, Patchstack, Wordfence, WPScan.

Datos de Vulnerabilidades (por componente)

	2020	2021	2022
Core	3,8%	0,6%	0,6%
Plugins	82,1%	92,8%	92,8%
Temas	14,1%	6,6%	6,7%

Fuentes: WPVulnerability, Patchstack, Wordfence, WPScan.

Datos de Vulnerabilidades (por gravedad)

	2021	2022
Bajo	2%	3%
Medio	77%	84%
Alto	18%	11%
Crítico	3%	2%

Fuentes: WPVulnerability, Patchstack, Wordfence, WPScan.

Datos de Vulnerabilidades

En 2020, más del 90% de los WordPress han sido vulnerables en algún momento.

De media, el 42% de los WordPress tienen al menos un componente vulnerable.

Datos de Vulnerabilidades

En 2020, el 96,22% de las vulnerabilidades son debidas a elementos externos al núcleo de WordPress.
En 2021, aumenta al 99,42%

En 2021, el 29% de los plugins vulnerables no tuvieron ningún parche.
En 2022, el 26% nunca lo recibió.

Datos de Vulnerabilidades

En 2021, PHPMailer, provocó una actualización de seguridad en todas las versiones desde WordPress 3.7 hasta 5.8 (sí, 21 versiones mayores).

Esto generó un sistema de protección de dependencias externas en el núcleo de WordPress.

Datos de Vulnerabilidades

En 2021, dos plugins (+3M y +1M de instalaciones) tuvieron vulnerabilidades.

- All in One SEO plugin ≤ 4.1.5.2
- WP Fastest Cache ≤ 0.0.4

Se parchearon en versiones menores antes de que se hiciera pública la vulnerabilidad.

Datos de Vulnerabilidades

En 2022, 5 «page builders» tuvieron vulnerabilidades.

Dos bugs en Elementor, con más de 5M de instalaciones y con una vulnerabilidad alta (8,8 / 10).

Cómo proteger tu WordPress

Plugins / Temas de confianza
- Proveedores de confianza
- Actualizados y mantenidos

Información de vulnerabilidades
- WPVulnerability *(100% gratuito)*

Cómo proteger tu WordPress

Firewall
- Soft-rules

Ataques fuerza bruta
- MFA

Ayuda con ciberseguridad

Teléfono: **017**

Particulares – Oficina del Internauta

https://www.osi.es/es/contacto

Empresas – Instituto Nacional de Ciberseguridad

https://www.incibe.es/formulario-contacto-empresas

¡He encontrado un fallo!

1. Descubrimiento de la vulnerabilidad
   - Patchstack

   https://patchstack.com/database/report

   - Wordfence

   https://www.wordfence.com/request-cve/

   - WPScan

   https://wpscan.com/submit

¡He encontrado un fallo!

2. Investigación y confirmación
3. Notificación
4. Análisis y parche
5. Publicación del parche
6. Divulgación

Este proceso puede tardar desde días hasta años.
Por norma general el tiempo de divulgación es de 3 meses.

WordPress está para ayudarte.

¿Quieres participar en la comunidad?

Mentorización del equipo de Hosting y Documentación

profiles.w.org/javiercasares

#WCBCN

2023 WC BCN