| ITEM | CATEGORÍA | CHECKLIST (Qué se verifica? | ¿Cómo se verifica? |
|---|---|---|---|
| A1 | Inyección | **¿Se puede acceder a los datos con seguridad? ¿Están bien definidos los roles ?** | 1. Ingresar desde la ruta de login desde la aplicación y validar que esté activo el https desde el navegador y después de ingresar validar el network (consola de desarrollador) u otra herramienta que permita capturar la transmisión de datos (SELENIUM).<br>2. Verificar desde la cuenta del administrador de la aplicación si se puede crear usuarios con diferentes roles y autenticarse luego. (posteriormente pasar la verificación número 1) |
| A2 | Pérdida de autenticación y gestión de sesiones | **Validar mecanismos de autenticación fuertes.** | 1. Revisar uso de contraseñas seguras, doble verificación, claves secretas en wp-config.php, errores de login ocultos, y acceso denegado al archivo wp-config.php. |
| A3 | Datos sensibles accesibles | **Comprobar si los datos personales o de tarjetas se protegen adecuadamente.** | 1. Verificar uso de HTTPS, cumplimiento RGPD, hosting PCI compliance, eliminación de datos sensibles y permisos adecuados en los archivos. |
| A4 | Entidad externa de XML (XXE) | **Prevenir ataques mediante archivos XML malicioso**s. | 1. Comprobar que no se usen parsers XML en PHP o extensiones como XMLWRITER.<br>2. Validar que libxml_disable_entity_loader(true); esté implementado. |
| A5 | Control de acceso inseguro | **Revisar que los usuarios no puedan acceder a funciones no autorizadas.** | 1. Probar acceso a rutas admin sin permisos, desactivar XML-RPC, bloquear rutas JSON REST no utilizadas desde .htaccess. |

| | | | |
|---|---|---|---|
| A6 | Configuración de seguridad incorrecta | **Verificar configuraciones seguras del entorno WordPress.** | 1. Validar permisos para archivos, para carpetas, para .htaccess y wp-config.php.<br>2. No dejar configuraciones ni contraseñas por defecto |
| A7 | Cross site scripting (XSS) | **Comprobar si se permite inyección de scripts en formularios o comentarios** | 1. Validar y sanear entradas usando funciones como sanitize_text_field() y escapar salidas con esc_html().<br>2. Probar con scripts como <script>alert(1)</script>. |
| A8 | Decodificación insegura | **Verificar que objetos serializados no puedan ser manipulados.** | 1. Mantener WordPress actualizado y evitar objetos no firmados digitalmente.<br>2. Asegurar integridad con mecanismos de validación. |
| A9 | Componentes con vulnerabilidades | **Asegurar que plugins, temas y componentes estén libres de fallos conocidos**. | 1. Usar herramientas como WPScan o páginas como wpvulndb.com para revisar vulnerabilidades conocidas en los plugins y temas instalados. |
| A10 | Insuficiente monitorización y registro | **¿La app registra intentos fallidos de login, accesos sospechosos o errores?** | 1. Usar plugins como WP Activity Log, revisar funciones peligrosas (eval, base64, exec) en el código y cumplir con normativas como RGPD para trazabilidad. |