Informe de Auditoría de Seguridad (Entrega 2)

Asignatura: Sistemas de Gestión de Seguridad de la Información **Autor:** Equipo Rojo **Fecha:** 29/10/2025

---

1. Introducción y Resumen

Este documento es el informe de auditoría que se pide para la Entrega 2. El objetivo principal era analizar la aplicación web que hicimos en la Entrega 1 para ver qué fallos de seguridad tenía.

Para hacerlo, hemos usado la herramienta OWASP ZAP (v2.16.1). Después de configurar el proxy y escanear la web (que corría en `http://localhost:81`), ZAP ha encontrado **12 tipos de alertas** diferentes.

Los problemas más importantes que hemos visto son los de riesgo "Medio", como la **Ausencia de Tokens Anti-CSRF** y problemas con las **cookies (no tenían `HttpOnly` ni `SameSite`)**. También hay un montón de fallos de "Configuración de Seguridad Insuficiente", como cabeceras (headers) que faltan.

Lo bueno es que, aunque hemos lanzado el escaneo activo, **ZAP no ha encontrado ningún fallo de Inyección SQL**.

---

2. Herramientas y Proceso de Auditoría

- **Herramienta:** Hemos usado OWASP ZAP (Zed Attack Proxy), la versión 2.16.1.
- **Objetivo:** La aplicación web de la Entrega 1, ejecutándose en `http://localhost:81`.

El proceso que hemos seguido ha sido este:

1. **Configuración:** Primero tuve que configurar el proxy de Firefox (en `localhost:8080`) y cambiar los ajustes (`about:config`) para que pudiera "ver" el tráfico de `localhost`.
2. **Exploración Manual:** Hemos navegado por toda la web (haciendo login, visitando el perfil, etc.) para que ZAP "aprendiera" todas las páginas y formularios que tiene el sitio.

3. **Escaneo Activo:** Una vez ZAP tenía el mapa del sitio, hemos lanzado el "Active Scan" para que intentara atacar la web y buscar fallos de forma automática.

---

3. Resultados de la Auditoría (Alertas Encontradas)

El escaneo de ZAP ha terminado con 12 tipos de alertas. Las agrupo aquí por el riesgo que ZAP les ha dado:

Fallos de Riesgo Medio (Los más importantes)

ZAP encontró 4 alertas de riesgo medio, casi todas relacionadas con la gestión de la sesión y el control de acceso:

- **Ausencia de Tokens Anti-CSRF:** Una solicitud falsa entre sitios en un ataque que compromete y obliga a una víctima a enviar su solicitud HTTP a un destino objetivo sin su conocimiento o intención para poder realizar una acción como víctima.

  Detectado en :http://localhost:81/register

- **Cookie Sin Flag HttpOnly:** Se ha establecido una cookie sin el flag HttpOnly, lo que significa que JavaScript puede acceder a la cookie. Si un script malicioso puede ser ejecutado en esta página, entonces la cookie será accesible y puede ser transmitida a otro sitio.

  Detectado en: http://localhost:81/sitemap.xml

- **Cookie sin el atributo SameSite:** Se ha establecido una cookie sin el atributo SameSite, lo que significa que la cookie puede ser enviada como resultado de una solicitud 'cross-site'. El atributo SameSite es una medida eficaz para contrarrestar la falsificación de peticiones entre sitios, la inclusión de scripts entre sitios y los ataques de sincronización.

- **Falta de cabecera Anti-Clickjacking (X-Frame-Options):** La web se puede meter en un `<iframe>` en una página maliciosa, engañando al usuario para que haga clic en cosas que no quiere. La respuesta no protege contra ataques de "ClickJacking". Se debe incluir Content-Security-Policy con la directiva "frame-ancestors" o X-Frame-Options.

Fallos de Riesgo Bajo (Configuraciones)

Estos son principalmente "Configuraciones de seguridad insuficientes" que hay que arreglar:

- **Cabecera Content Security Policy (CSP) no configurada:** La Política de seguridad de contenido (CSP) es una capa adicional de seguridad que ayuda a detectar y mitigar ciertos tipos de ataques. CSP proporciona un conjunto de encabezados HTTP estándar que permiten a los propietarios de sitios web declarar fuentes de contenido aprobadas que los navegadores deberían poder cargar en esa página.

- **Falta encabezado X-Content-Type-Options:** La cabecera Anti-MIME-Sniffing X-Content-Type-Options no se ha establecido en 'nosniff'. Esto permite que las versiones anteriores de Internet Explorer y Chrome realicen MIME-sniffing en el cuerpo de la respuesta, lo que puede provocar que el cuerpo dé la respuesta se interprete y se muestre como un tipo de contenido distinto del tipo de contenido declarado.

- **Petición de Autenticación Identificada:** ZAP simplemente informa de que ha encontrado el formulario de login.

  Detectado en: http://localhost:81/login

Alertas Informativas (Pistas para un atacante)

Estas no son vulnerabilidades directas, pero dan demasiada información:

- **El servidor divulga información: "X-Powered-By"**: El acceso a tal información podría facilitarle a los atacantes la identificación de otros marcos/componentes de los que su aplicación web depende y las vulnerabilidades a las que pueden estar sujetos tales componentes.
- **El servidor filtra información de versión: "Server"** (también dice qué servidor Apache es usado).
- **Divulgación de Información - Información sensible en URL** (ZAP ha visto IDs o datos en la URL).
- **Divulgación de información - Comentarios sospechosos** (hemos dejado comentarios en el HTML/JS que ZAP ha visto).
- **Respuesta de Gestión de Sesión Identificada:** ZAP informa de que ha detectado cómo el servidor gestiona las cookies de sesión, la respuesta dada contiene un token de gestión de sesión.

---

4. Vulnerabilidades NO Detectadas (Inyección SQL)

El documento del trabajo pedía explícitamente buscar vulnerabilidades de **Inyección**. Lancé el "Active Scan" de ZAP contra todas las páginas y formularios, pero la herramienta **no reportó ningún fallo de Inyección SQL**.

Esto es un resultado positivo. Como se pide en el trabajo, la explicación es que el código de la Entrega 1 ya se hizo pensando en esto: **toda la comunicación con la base de datos se hace usando consultas preparadas (prepared statements) de PDO**. Esto evita que la entrada del usuario se pueda ejecutar como código SQL, neutralizando el ataque.

---

5. Conclusión

La auditoría ha funcionado. ZAP ha encontrado 12 problemas claros que hay que solucionar. Aunque no hay fallos críticos de Inyección, los problemas de CSRF y las cabeceras de seguridad son importantes y hay que arreglarlos.

Todos estos hallazgos serán la base para el trabajo de la Entrega 3, donde tendremos que solucionar cada uno de ellos y volver a pasar el escáner.