Informe de Análisis de Amenazas y Vulnerabilidades

Sitio Analizado: https://www.ejemplo-universidad.cl (página institucional ficticia para fines del ejercicio)
Fecha del análisis: 5 de junio de 2025

1. Aplicación Web Seleccionada

Nombre del sitio: Universidad Nacional Ejemplo
Tipo de aplicación: Portal institucional público
Funcionalidades observadas:

- Formulario de login
- Formulario de contacto
- Buscador en sitio
- Área de noticias y comentarios

2. Inspección Técnica (F12)

- **Formularios:**
  - Login (`/login`)
  - Contacto (`/contacto`)
  - Buscador (`/buscar?query=...`)
  - Comentarios bajo artículos (`/noticia/123`)
- **URLs y parámetros visibles:**
  - `/buscar?query=admin`
  - `/usuario?id=5`

- **Comportamiento ante errores:**
    - 404 genérico, pero revela estructura de carpetas: `"/app/modules/user/not-found"`
- **Cookies almacenadas:**
    - `session_id=xyz123` → sin atributos `HttpOnly` ni `Secure`
- **Códigos HTTP observados:**
    - `200 OK`, `302 Redirect`, `404 Not Found`

3. Amenazas Potenciales

Nº	Tipo	Descripción	Riesgo Potencial	Medida de Mitigación
1	Amenaza Externa	XSS en buscador (refleja parámetro `query`)	Robo de datos / secuestro de sesión	Escapar HTML y validar entrada
2	Amenaza Externa	Posibilidad de phishing a través de enlace `mailto:` directo	Suplantación de identidad	Redireccionar siempre con aviso
3	Amenaza Externa	Enlaces externos sin atributo `rel="noopener"`	Robo de contexto de ventana	Aplicar `rel="noopener noreferrer"`
4	Amenaza Interna	Exposición de `/admin` sin autenticación previa	Acceso no autorizado	Requiere autenticación con roles
5	Amenaza Interna	URL `/usuario?id=5` visible y navegable	Curiosear otros perfiles	Implementar control de acceso por usuario

4. Vulnerabilidades Visibles

Nº	Tipo	Descripción	Riesgo Potencial	Medida de Mitigación
1	Vulnerabilidad	Cookie sin atributos `Secure` y `HttpOnly`	Robo de sesión	Configurar correctamente cookies
2	Vulnerabilidad	Formularios sin token CSRF	Ataques CSRF	Añadir token único por sesión
3	Vulnerabilidad	Parámetro manipulable en URL `/usuario?id=5`	Acceso no autorizado a otros usuarios	Validar ID y restringir por sesión

5. Buenas Prácticas de Seguridad Propuestas

1. Implementar validación estricta del lado servidor y cliente en todos los formularios.
2. Forzar el uso de HTTPS en todo el sitio.
3. Establecer cookies con atributos `Secure`, `HttpOnly` y `SameSite=Strict`.
4. Utilizar tokens CSRF únicos en formularios críticos.
5. Gestionar errores mostrando mensajes genéricos sin revelar estructura interna.
6. Controlar el acceso a rutas por medio de roles y sesiones válidas.
7. Escapar y sanitizar toda entrada de usuario en el frontend y backend.

6. Reflexión Final

Durante este análisis observacional, se pudo evidenciar que, a pesar de ser un sitio institucional, la aplicación web presenta varias debilidades fundamentales en su diseño de seguridad. La falta de protección en las cookies, la visibilidad de rutas administrativas y la exposición de parámetros URL sin control adecuado son indicios de una implementación insegura.

Me sorprendió especialmente la falta de tokens CSRF en formularios sensibles y la posibilidad de navegar a otros perfiles con solo modificar el parámetro `id` en la URL.

Aplicar las buenas prácticas propuestas no solo mitigaría los riesgos existentes, sino que aumentaría significativamente la confianza de los usuarios y la resiliencia del sistema ante ataques comunes. Este ejercicio me hizo más consciente de que incluso sin herramientas

avanzadas, es posible detectar fallas graves de seguridad simplemente utilizando el navegador y el sentido crítico. Por tanto, mantener la seguridad desde el diseño inicial es clave para cualquier proyecto web.