**Implementación Básica de Autenticación y Autorización en una Aplicación Web**

|  |  |
| --- | --- |
| Título del informe: Implementación Básica de Autenticación y Autorización en una Aplicación Web | |
| Portafolio:  Lección 2: Implementación Básica de Autenticación y Autorización | Entorno:  Entorno desarrollado para la prueba. |
| Clasificación del documento:  Entorno vulnerable para pruebas – publica | Modulo:  Principios de seguridad en aplicaciones web |
| Autor(es): Sebastián Hernández Téllez | Fecha de elaboración: 23 de agosto de 2025 |

**1. Resumen Ejecutivo**

El laboratorio tuvo como objetivo implementar un sistema básico de autenticación y autorización en una aplicación web de gestión de publicaciones. Se desarrolló un flujo de registro e inicio de sesión con contraseñas almacenadas de forma segura mediante hashing con bcrypt, evitando guardar datos en texto claro. Se definieron dos roles: Administrador, con privilegios de creación, edición y eliminación de contenido; y Usuario común, con acceso restringido únicamente a la lectura. Se verificó que las restricciones se aplican correctamente, impidiendo accesos indebidos a usuarios no autenticados o sin privilegios. Además, se identificaron riesgos comunes y se plantearon medidas de mitigación como MFA, tokens CSRF y cabeceras de seguridad. El ejercicio demostró la importancia de incorporar la ciberseguridad desde las fases iniciales del desarrollo.

***Escenario***

Se desarrolló una aplicación web de gestión de publicaciones para un blog, con dos perfiles de usuario:

* **Administrador**: acceso total (crear, editar y eliminar publicaciones).
* **Usuario común**: acceso limitado (solo visualizar publicaciones).

El objetivo fue implementar un sistema básico de autenticación y autorización que garantice el acceso seguro, diferenciando funciones según el rol asignado.

**2. Metodología**

La práctica se diseñó bajo un enfoque de **seguridad por defecto (Secure by Design)**, aplicando los siguientes pasos:

1. **Registro de usuarios con contraseñas seguras** → almacenamiento mediante hashing con **bcrypt**.
2. **Inicio de sesión seguro** → validación de credenciales y generación de sesión autenticada.
3. **Roles de usuario** → asignación automática de *Usuario común* en el registro, con posibilidad de asignar *Administrador* desde un panel restringido.
4. **Restricción de acceso** → middleware de autorización que bloquea funciones sensibles a usuarios sin privilegios.
5. **Pruebas de seguridad** → validación de que un usuario no autenticado ni con rol insuficiente pueda acceder a recursos restringidos.

**3. Implementación Técnica**

***Autenticación con contraseñas***

* Página de **registro** con usuario y contraseña.
* Contraseñas almacenadas en la base de datos con **bcrypt** (hash + salt).
* Página de **login** que valida credenciales y crea sesión con token seguro.

***Roles de usuario***

* **Por defecto:** todos los usuarios nuevos se crean con rol “Usuario”.
* **Panel de administración:** solo accesible para Administradores, permite asignar o cambiar roles.

***Restricción de acceso***

* **Administrador:** CRUD completo de publicaciones.
* **Usuario común:** acceso en modo lectura.
* **Middleware de autorización:** comprueba sesión y rol antes de permitir acceso a rutas críticas.

**4. Pruebas Realizadas**

| **Prueba** | **Resultado esperado** | **Estado** |
| --- | --- | --- |
| Registro de usuario | Creación exitosa con hash seguro | ✅ |
| Login con credenciales válidas | Acceso autorizado y creación de sesión | ✅ |
| Login con credenciales erróneas | Acceso denegado | ✅ |
| Usuario común edita/elimina | Acceso denegado | ✅ |
| Administrador edita/elimina | Acceso permitido | ✅ |
| Acceso sin autenticación | Bloqueo y redirección a login | ✅ |

**5. Problemas Comunes y Soluciones**

* **Usuarios no autenticados accediendo a publicaciones** → Solución: Middleware que exige sesión activa.
* **Contraseñas en texto claro** → Solución: Hashing con bcrypt.
* **Elevación de privilegios por manipulación de parámetros** → Solución: Validación de roles en el servidor, no solo en la interfaz.

**6. Recomendaciones de Seguridad**

1. Implementar **autenticación multifactor (MFA)** para reforzar accesos administrativos.
2. Usar **tokens CSRF** en formularios para prevenir ataques de falsificación de peticiones.
3. Aplicar cabeceras de seguridad HTTP (**CSP, HSTS**) para mitigar XSS.
4. Establecer políticas de contraseñas robustas (mínimo 12 caracteres, complejidad).
5. Monitoreo de accesos y registros de actividad para detección temprana de anomalías.

**7. Reflexión**

El laboratorio mostró la importancia de **separar autenticación y autorización** como mecanismos distintos. Una contraseña segura no es suficiente si no existe un control de roles bien implementado. El uso de hashing robusto y restricciones basadas en permisos aporta mayor seguridad y reduce riesgos de acceso indebido.

**8. Conclusión**

La implementación realizada permitió disponer de un sistema seguro de **login y roles**, con restricciones claras entre Administradores y Usuarios comunes. Las pruebas confirmaron que la aplicación cumple con los objetivos de seguridad: contraseñas protegidas, acceso diferenciado y bloqueo de intentos no autorizados. Este ejercicio demuestra la relevancia de integrar mecanismos de autenticación y autorización desde las fases iniciales de desarrollo, conforme a los principios de la **ciberseguridad y el OWASP Top 10**.

**9. Anexo**

**Capturas de pantalla con ejemplos de código**

***Modelo de Usuario (Mongoose: hash + roles)***

**A screenshot of a computer program

AI-generated content may be incorrect.  
2) App base segura: Helmet, Rate Limit, Cookies, CSRF**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**3) JWT en cookie HttpOnly (login / logout)**

**A screen shot of a computer

AI-generated content may be incorrect.**

**A screen shot of a computer

AI-generated content may be incorrect.**

**A screen shot of a computer

AI-generated content may be incorrect.**

**4) Middlewares de Autenticación y Autorización por Rol**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**5) Rutas protegidas (evidencia de restricción por rol)**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**6) Activación MFA (TOTP) – opcional (evidencia de hardening)**

**A screen shot of a computer

AI-generated content may be incorrect.**

**A screen shot of a computer

AI-generated content may be incorrect.**

**7) Montaje del servidor (rutas + orden de middlewares)**

**A screen shot of a computer program

AI-generated content may be incorrect.**