sustantiva
CON•SENTIDO

Conceptos Básicos y Buenas Prácticas para Aplicaciones Web

Introducción

Introducción

- La gestión segura de sesiones es crucial para **salvaguardar la confidencialidad e integridad** de la información en las aplicaciones web.
- **Sesión**: Interacción activa entre un usuario y una aplicación web.
- Un manejo incorrecto puede resultar en **secuestración de sesión** (session hijacking), poniendo en riesgo los datos sensibles y la identidad digital.

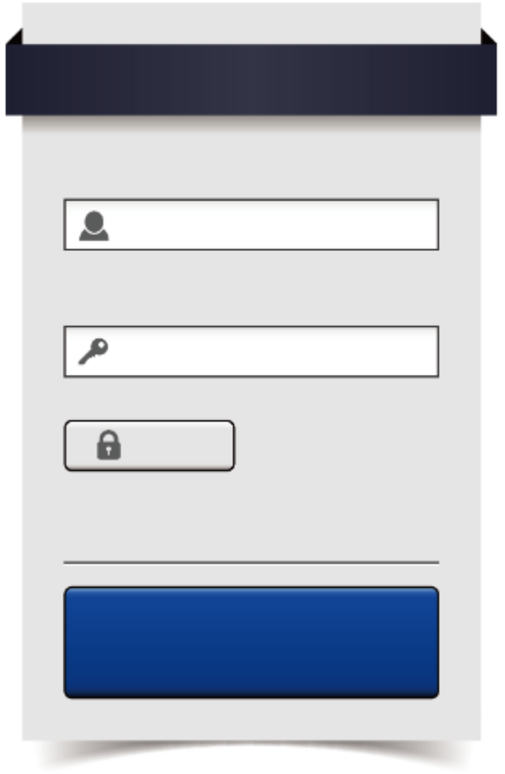
¿Qué es una Sesión?

¿Qué es una Sesión?

- **Sesión:** Periodo en que un usuario interactúa con una aplicación web.

- Objetivo de la gestión de sesiones:
  - **Crear identificadores únicos y seguros.**
  - **Mantener la sesión activa** mientras sea necesario.
  - **Cerrar y destruir la sesión** al finalizar.

Importancia: Si no se gestiona correctamente, los atacantes pueden robar la identidad del usuario.



Creación de Sesiones Seguras

Creación de Sesiones Seguras

- **Generación de Tokens:**
  - Usar **técnicas criptográficas robustas** para crear identificadores de sesión únicos.
  - Evitar la predicción de los tokens.

- **Persistencia y Destrucción:**
  - Almacenar el token de forma segura (en el servidor o en cookies).
  - Invalidar el token al finalizar la sesión o tras la expiración por inactividad.

Métodos de Almacenamiento de Sesiones

Métodos de Almacenamiento de Sesiones

- **Cookies:**
  - **Secure:** Solo se envían por conexiones HTTPS.
  - **HttpOnly:** Previene que scripts maliciosos accedan a la cookie.
  - **SameSite:** Protege contra ataques CSRF.

- **JWT (JSON Web Tokens):**
  - Compactos y autocontenidos.
  - Firmados digitalmente para evitar manipulaciones.
  - Útiles en arquitecturas RESTful y microservicios.

Prevención del Secuestro de Sesiones

Prevención del Secuestro de Sesiones

- **Uso de HTTPS:** Cifra las comunicaciones para evitar la interceptación.
- **Regeneración de Tokens:** Cambiar el token después de eventos críticos como inicio de sesión o cambio de contraseña.
- **Uso de Flags de Seguridad en Cookies:**
  - **Secure, HttpOnly, SameSite.**
- **Medidas Anti-CSRF:**
  - Verificación de cabeceras **Origin o Referer.**

Ejemplo de Vulnerabilidad (XSS y Secuestro de Sesión)

Ejemplo de Vulnerabilidad (XSS y Secuestro de Sesión)

- **Escenario:**
  - La aplicación guarda la sesión del usuario en una cookie sin **HttpOnly ni Secure.**
  - Un atacante inserta un **script malicioso** que roba la cookie de sesión a través de XSS.
- **Herramientas** como **OWASP ZAP** o **Burp Suite** pueden usarse para demostrar estas vulnerabilidades en entornos controlados.

Conclusión

Conclusión

- La gestión de sesiones segura es esencial para **proteger los datos y fortalecer la confianza del usuario.**
- Utilizar **tokens seguros, HTTPS, cookies protegidas** y técnicas como **JWT** ayuda a prevenir **errores costosos.**
- Implementar buenas prácticas no solo mejora la seguridad, sino también la calidad de la aplicación web.


Energiza!