# Universidad Autónoma Tomás Frías Facultad de Ciencias Puras

**Carrera de Ingeniería Informática Seguridad de Software (INF 633)**



**Práctica No**



**3**

**Fecha de publicación:** 8/11/2024

**Fecha de presentación:** 15/11/2024

**Formato de presentación:** La explicación debe hacerla en formato Markdown. Documentación y código fuente deberá ser subido a GitHub y entregado mediante plataforma virtual de la UATF hasta la fecha señalada.

NOMBRE: Natalia Coro Vasquez

# Implementación de Prácticas de Seguridad en el Desarrollo de Aplicaciones Web

**Objetivo:**

Aplicar prácticas de codificación segura para mitigar vulnerabilidades comunes en el desarrollo de aplicaciones web.

# Instrucciones:

1. **Validación y Sanitización de Entradas:**
   * Describe los pasos necesarios para validar y sanitizar entradas de usuario en una aplicación web.

* **Validación**: Confirma que los datos cumplen con los requisitos esperados (por ejemplo, formato de correo electrónico).
* **Sanitización**: Elimina o codifica caracteres especiales para evitar la ejecución de scripts maliciosos.
  + Implementa un ejemplo en un lenguaje de programación a elección, donde demuestres la validación de correos electrónicos y la sanitización de un campo de texto para prevenir ataques XSS.



# Control de Acceso y Autenticación Segura:

* + Explica los beneficios de la Autenticación Multifactor (MFA) y la autenticación basada en tokens (JWT).
* **Autenticación Multifactor (MFA)**:
* Agrega una capa extra de seguridad al requerir múltiples métodos de verificación.
* Mejora la protección contra ataques de fuerza bruta y el acceso no autorizado.
* **Autenticación Basada en Tokens (JWT)**:
* Utiliza tokens para verificar la identidad de un usuario sin necesidad de gestionar sesiones.
* Los tokens JWT son seguros, portátiles y se pueden usar para verificar al usuario en diferentes servicios.
  + Crea un código en cualquier lenguaje de programación que demuestre la creación de un JWT para un usuario.



# Gestión de Sesiones y Cookies:

* + Investiga sobre los atributos de seguridad HttpOnly y Secure en cookies y explica cómo ayudan a proteger la sesión de usuario.

**Atributos de Seguridad en Cookies**:

* **HttpOnly**: Impide que las cookies sean accedidas mediante JavaScript, evitando ataques XSS.
* **Secure**: Asegura que las cookies solo se envíen por conexiones HTTPS, protegiéndolas de intercepciones.
  + Implementa una configuración de sesión en una aplicación web que utilice estos atributos.

**EJEMPLO EN JAVASCRIP:**

const express = require('express');

const app = express();

app.get('/establecer-cookie', (req, res) => {

res.cookie('session\_id', 'valor\_unico', {

httpOnly: true,

secure: true,

maxAge: 3600000 // 1 hora

});

res.send("Cookie de sesión establecida con HttpOnly y Secure.");

});

app.listen(3000, () => {

console.log("Servidor en funcionamiento en http://localhost:3000");

});

# Protección de Datos Sensibles:

* + Explica la diferencia entre hashing y cifrado, y da ejemplos de cuándo usar cada uno.

**Diferencia entre Hashing y Cifrado**:

* **Hashing**: Convierte datos en una cadena de longitud fija. Es irreversible y adecuado para almacenar contraseñas. Ejemplo: SHA-256.
* **Cifrado**: Convierte datos en una cadena que puede ser revertida con una clave secreta. Se usa para proteger datos en tránsito, como información personal.
  + Implementa una función en PHP que use password\_hash y password\_verify para manejar contraseñas de usuarios de forma segura.

