sustantiva
CON•SENTIDO

Vulnerabilidades Comunes en Aplicaciones Web

Introducción

Introducción

- Las aplicaciones web están expuestas a múltiples ataques.
- Las vulnerabilidades más comunes incluyen:
  - **Inyección SQL (SQLi)**
  - **Cross-Site Scripting (XSS)**
  - **Cross-Site Request Forgery (CSRF)**
- El conocimiento y aplicación de buenas prácticas es clave para prevenirlas.

Inyección SQL (SQLi)

Inyección SQL (SQLi)

¿Qué es?

Inserción de código malicioso en consultas SQL.

Tipos:

- Clásica
- Basada en errores
- Blind SQL
- Basada en tiempo

Ejemplo de ataque:

http://dominio.com/login.php?usuario=admin' OR '1'='1

Mitigación de SQLi

Mitigación de SQLi

Consecuencias:

- Acceso no autorizado
- Robo de datos
- Compromiso del sistema

Cómo prevenir:

- Validación de entradas
- Sanitización con listas blancas
- Uso de Prepared Statements:

```
$stmt = $pdo->prepare('SELECT * FROM usuarios WHERE nombre_usuario = :usuario');
$stmt->execute(['usuario' => $usuario]);
```

Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS)

¿Qué es?

Inserción de scripts maliciosos en páginas vistas por otros usuarios.

Tipos:

- Reflejado
- Almacenado
- DOM-based

Ejemplo de ataque:

http://dominio.com/busqueda?q=<script>alert("Ataque XSS");</script>

Mitigación de XSS

Mitigación de XSS

Impacto:

- Robo de cookies
- Malware
- Daño a la reputación

Prevención:

- Validar y codificar entradas
- Escapar caracteres especiales:

htmlspecialchars($input, ENT_QUOTES, 'UTF-8');

Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF)

¿Qué es?

Engañar a un usuario autenticado para ejecutar acciones sin su consentimiento.

Ejemplo:

<img src="http://dominio.com/cambiar_password?pass=nuevoPass">

Mitigación de CSRF

Consecuencias:

- Cambios no autorizados
- Transferencias fraudulentas
- Pérdida de confianza

Prevención:

- Tokens únicos en formularios
- Validar Referer/Origin
- Cookies con SameSite=Strict

```
<input type="hidden" name="csrf_token" value="token_seguro_12345">
```

Conclusión

Conclusión

- Las vulnerabilidades web son frecuentes, pero prevenibles.
- Aplicar buenas prácticas:
  - Validación de entradas
  - Preparar consultas SQL
  - Codificar salidas
  - Proteger formularios con tokens
- La seguridad debe integrarse en cada etapa del desarrollo.


Energiza!