Seguridad en Sistemas Web

Mikel Egaña Aranguren

mikel-egana-aranguren.github.io

mikel.egana@ehu.eus

Universidad del País Vasco
Euskal Herriko Unibertsitatea
BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

Seguridad en Sistemas Web

https://doi.org/10.5281/zenodo.4302267

https://github.com/mikel-egana-aranguren/EHU-SGSSI-01

Pen-testing

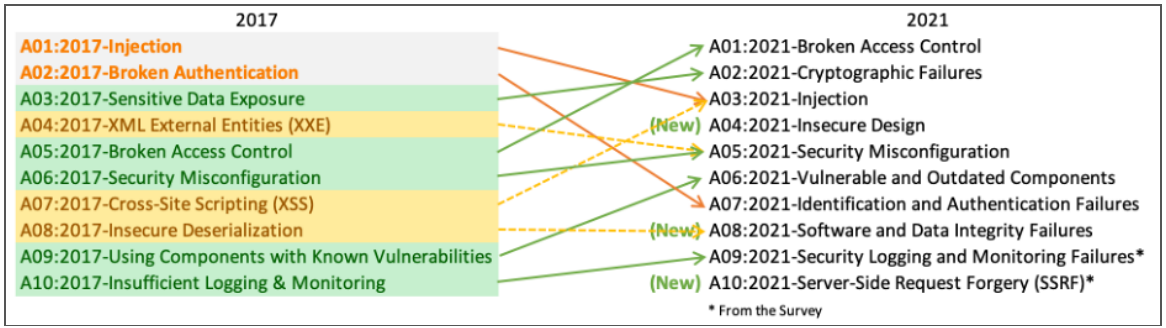
Penetration testing: intentar penetrar en un sistema utilizando las mismas herramientas que un potencial atacante, para descubrir vulnerabilidades y arreglarlas

Principales vulnerabilidades en Sistemas Web

La Open Web Application Security Project (OWASP) analiza las vulnerabilidades más comunes

Informe periódico: OWASP Top Ten (~~2017~~, 2021)

Principales vulnerabilidades en Sistemas Web



A01:2021 - Rotura de control de acceso (Broken Access control)

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

- CWE-200: Exposure of Sensitive Information to an Unauthorized Actor
- CWE-201: Insertion of Sensitive Information Into Sent Data
- CWE-352: Cross-Site Request Forgery
- …

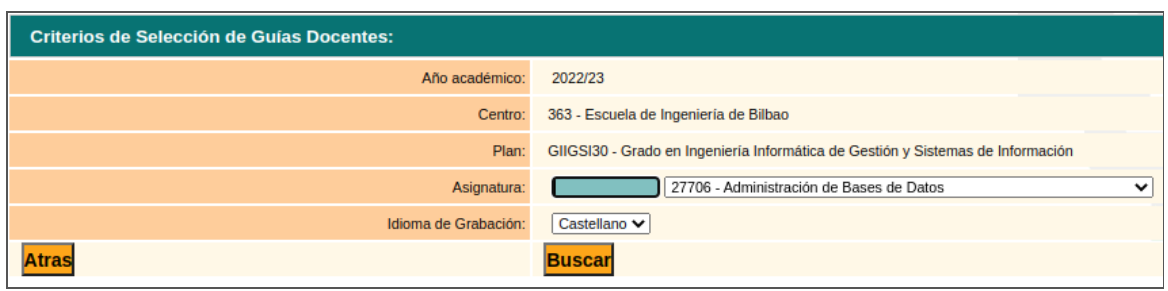
A01:2021 - Rotura de control de acceso (Broken Access control)

El control de acceso obliga al usuario a actuar solo dentro de los límites establecidos por los permisos definidos

Fallar en el control de acceso resulta en consecuencias graves

A01:2021 - Broken Access control - vulnerabilidades

Violación del principio de "denegación por defecto": el acceso solo debería ser garantizado a ciertas capacidades, roles, o usuarios, pero está al alcance de cualquiera



A01:2021 - Broken Access control - vulnerabilidades

Burlar el control de acceso mediante parametros URL o las peticiones APIs

Acceder a una cuenta personal e incluso modificar los datos con el identificador del usuario

Acceso API sin control para los metodos HTTP POST, PUT, y DELETE

A01:2021 - Broken Access control - vulnerabilidades

Elevación de privilegio: acutar como administrador estando logeado como usuario

Replaying o modificación de JSON Web Token (JWT) para elevar privilegios

Mala configuración de CORS permite acceso API desde origenes no autorizados

A01:2021 - Broken Access control - prevención

Denegación por defecto para todos los recursos, excepto públicos

Implementar mecanismos de acceso de control una sola vez y extenderlos a toda la aplicación, minimizando el uso de CORS

No permitir listado de directorios y asegurarse de que no hay archivos de metadatos (ej. git) ni de backups en el nivel root de la aplicación

A01:2021 - Broken Access control - prevención

Logear todos los intentos de entrada, alertar a los administradores cuando sea necesario (ej. muchos accesos fallidos)

Limitar la tasa de peticiones de API para evitar ataques mediante programas

A01:2021 - Broken Access control - prevención

Identificadores de sesión:

- De servidor (stateful): invalidar al deslogearse
- Stateless JWT tokens: muy poca vida
- Para JWT de tiempo más largo, seguir el protocolo OAuth de cara a revocar el acceso

A01:2021 - Broken Access control - ejemplo

Llamada a SQL para obtener información de cuenta, con datos sin verificar:

```
01. pstmt.setString(1, request.getParameter("acct"));
02. ResultSet results = pstmt.executeQuery( );
```

El atacante solo tiene que usar la URL

https://example.com/app/accountInfo?acct=notmyacct

A02:2021 - Fallos criptográficos (https://owasp.org/Top10/A02_2021-Cryptographic_Failures/)

https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

- CWE-259: Use of Hard-coded Password
- CWE-327: Broken or Risky Crypto Algorithm
- CWE-331 Insufficient Entropy
- …