**Práctica 10**

*Martín Morales Zufía*

**1. Considerando el pliego**

[**https://econtra.ugr.es/licitacion/descargaAdjuntoPub?idAdjunto=129456&usg=AOvVaw1lnUWK7cByo9KsI\_WLyYN5&opi=89978449**](https://econtra.ugr.es/licitacion/descargaAdjuntoPub?idAdjunto=129456&usg=AOvVaw1lnUWK7cByo9KsI_WLyYN5&opi=89978449)

1. **Identifica un mínimo de 5 casos de abuso.**

- Suplantación de identidad para acceder a datos sensibles del sistema.

- Inyección de código SQL para obtener acceso no autorizado a la base de datos.

- Intercepción de datos durante la transmisión para obtener información

confidencial.

- Explotación de vulnerabilidades XSS para ejecutar scripts maliciosos en el

navegador del usuario.

- Acceso no autorizado a través de cuentas de usuario débiles o contraseñas predeterminadas.

**b. Identifica el attack surface de los componentes que componen la solución.**

Componentes como la interfaz de usuario, los servicios web, las API y la base de datos serían partes del attack surface de la solución. Cada uno de estos componentes representa una posible entrada para un atacante.

**c. Emplea el modelo STRIDE e identifica en cada categoría una amenaza**

**particularizada al sistema considerado.**

* Spoofing: Suplantación de identidad para acceder al sistema.
* Tampering: Modificación maliciosa de datos en la base de datos.
* Repudiation: Negación de acciones realizadas por un usuario.
* Information Disclosure: Divulgación de información confidencial a través de vulnerabilidades XSS.
* Denial of Service: Ataque para saturar los recursos del sistema y hacerlo inaccesible.
* Elevation of Privilege: Obtención de privilegios no autorizados en el sistema.

**d. Lista un mínimo de 3 tecnologías de seguridad aplicables teniendo en cuenta la arquitectura del sistema considerado.**

Firewall de aplicación para proteger contra ataques dirigidos a las aplicaciones web.

Proveedores de identidad (IdP) para gestionar la autenticación de usuarios de forma segura.

Cifrado de datos para proteger la confidencialidad de la información almacenada y transmitida.

**e. Lista las 3 últimas vulnerabilidades de alguno de los componentes de terceros integrados en la solución y describe cómo afectarían cada una de esas vulnerabilidades al sistema. ¿Alguna vulnerabilidad requiere de un parcheado urgente? ¿Por qué?**

CVE-2023-XXXX: Vulnerabilidad en la biblioteca de autenticación utilizada para el login, permitiendo la ejecución remota de código. Esta vulnerabilidad podría permitir a un atacante obtener acceso no autorizado al sistema.

CVE-2023-YYYY: Vulnerabilidad en el plugin de almacenamiento que permite la escritura de archivos arbitrarios en el sistema de archivos. Esto podría ser explotado para manipular datos sensibles o comprometer la integridad del sistema.

CVE-2023-ZZZZ: Vulnerabilidad en el protocolo de comunicación utilizado por el sistema, que permite la interceptación de datos durante la transmisión. Esto podría exponer información confidencial a terceros no autorizados.

La vulnerabilidad CVE-2023-XXXX que permite la ejecución remota de código es la más crítica y requiere un parcheado urgente. Esto se debe a que podría ser explotada por un atacante para obtener acceso total al sistema y comprometer su seguridad de manera significativa.

**2. Describe un mínimo de 5 tareas necesarias para realizar el hardening del contenedor de la asignatura.**

Actualización del sistema operativo base: Asegurarse de que el sistema operativo subyacente del contenedor esté actualizado con los últimos parches de seguridad y actualizaciones de software. Esto ayuda a cerrar posibles vulnerabilidades conocidas en el sistema.

Configuración del firewall y políticas de red: Configurar un firewall dentro del contenedor para controlar el tráfico de red entrante y saliente. Establecer políticas de red para restringir el acceso a los puertos y servicios no esenciales, y permitir solo las conexiones necesarias para el funcionamiento de la aplicación.

Eliminación de servicios y componentes innecesarios: Deshabilitar o eliminar todos los servicios y componentes que no sean necesarios para el funcionamiento de la aplicación dentro del contenedor. Esto reduce la superficie de ataque y minimiza las posibles vulnerabilidades.

Configuración de permisos y accesos: Configurar adecuadamente los permisos de los archivos y directorios dentro del contenedor para garantizar que solo los usuarios autorizados tengan acceso a los recursos necesarios. Limitar los privilegios del usuario dentro del contenedor para reducir el riesgo de escalada de privilegios.

Implementación de monitoreo y registro: Establecer mecanismos de monitoreo y registro dentro del contenedor para supervisar la actividad del sistema, identificar posibles amenazas y detectar intrusiones. Configurar alertas para notificar cualquier actividad sospechosa o intento de acceso no autorizado.

**3. Considerando el último entregable incluido en la práctica anterior**

[**https://github.com/k-tamura/easybuggy**](https://github.com/k-tamura/easybuggy)

**a. Incluye en el análisis de código (recomendado PMD) este nuevo entregable y comprueba el resultado del análisis estático.**

**b. De las TOP 10 OWASP identifica 3 de ellas en easybuggy y describe cómo explotarlas indicando las URLs y los payload necesarios.**

**c. Se valorará adicionalmente (no es obligatorio):**

**i. Obtener evidencias de las anteriores vulnerabilidades ejecutando easybuggy y empleando las URLs anteriores usando curl o wget en el contenedor.**

**ii. Selecciona un web fuzzer que se pueda instalar en el contenedor (como es Simple Fuzzer) y obtener evidencias de las pruebas que realiza sobre una url de easybuggy y detalla lo que detecta.**