

Actividad [#2] – [Deserialización Insegura] [“Auditoría Informática”]

Ingeniería en Desarrollo de Software

**Tutor: Jessica Hernández Romero**

**Alumno: Manuel Enrique Ramirez Lopez**

**Fecha: 07/02/2023**

***Indice***

*[Introduccion](#_Toc2200)* [1](#_Toc2200)

*[Descripción](#_Toc11747)* [2](#_Toc11747)

*[Justificación](#_Toc4061)* [3](#_Toc4061)

*[Investigación](#_Toc6286)* [4](#_Toc6286)

*[Ataque al sitio](#_Toc17935)* [5](#_Toc17935)

*[Conclucion](#_Toc11388)* [7](#_Toc11388)

*[Referencias](#_Toc7000)* [8](#_Toc7000)

# *Introduccion*

La deserialización insegura es un problema de seguridad que puede surgir en aplicaciones que utilizan la serialización y la deserialización de objetos. La serialización es el proceso de convertir un objeto en una secuencia de bytes que se pueden almacenar o transmitir, mientras que la deserialización es el proceso inverso de reconstruir un objeto a partir de un bytes.

Sin embargo, si el proceso de deserialización no está implementado de forma segura, puede permitir un atacante ejecutar código malicioso en el sistema del usuario, lo que puede resultar en comprometer la seguridad, como la fuga de información confidencial o el control total del sistema por parte del atacante.

Por lo tanto, es importante que los desarrolladores presten atención a la seguridad durante el proceso de serialización y deserialización de objetos para prevenir la deserialización insegura y proteger a los usuarios y sus datos.

# *Descripción*

¿Que tipo de experiencia se busca obtener?

El como con un simple programa por llamarlo de una forma ambigua podemos acceder a un sistema “controlado” en una simulación donde podemos alterar informacion que es vital para los usuarios activos que maneja la plataforma, donde con un simple acceso o registro que puede realizar cualquier persona en el mundo podemos ir escalando por los diversos rangos que existan en la plataforma y o obtener algun beneficio gracias a esta vulnerabilidad, el objetivo principal es el saber como funciona este tipo de vulnerabilidad donde entre otras cosas debemos estar preparados para saber afrontar y prevenir esta vulnerabilidad y así asegurar la integridad de nuestro sistema y informacion de nuestros clientes.

# *Justificación*

El objetivo principal de esta practica es la obtención de experiencia en base a acciones de vulnerabilidad, es decir el saber defenderse cuando uno sabe como atacar para así maximizar la defensa en puntos claves donde nosotros desde el otro bando podríamos vulnerar la seguridad.

# *Investigación*

***¿Cómo funciona este ataque?***

El interceptar informacion no encapsulada seria la forma mas sencilla de ejemplificar este tipo de ataque donde el atacante busca obtener acceso mediante terceros con autorización autentica donde envía ligas/ enlaces con código malicioso donde busca que haga clic y este agregue datos de acceso y por medio de la transferencia de datos “cookie” tomar dicha informacion y suplantar su acceso y ya una ves dentro poder hacer y desacer lo que tenga a su alcance.

***¿Cómo afecta este ataque a los software y sus usuarios?***

La deserialización insegura puede ser explotada por un atacante para ejecutar código malicioso en el sistema del usuario, lo que puede resultar en la fuga de información confidencial o el control total del sistema por parte del atacante. Los usuarios que utilizan software vulnerable a la deserialización segura están en riesgo de sufrir daños graves. Por lo tanto, es importante que los desarrolladores aborden este problema de seguridad e implementen medidas de protección adecuadas en sus aplicaciones.

***¿Cómo se puede prevenir?***

Algunas medias para prevenir:

Validación de entrada: Validar cuidadosamente todas las entradas antes de deserializarlas para asegurarse de que cumplan con los requisitos esperados y no contengan datos maliciosos.

Uso de bibliotecas seguras: Utilice bibliotecas de serialización que hayan sido dañadas de forma segura y hayan sido probadas rigurosamente.

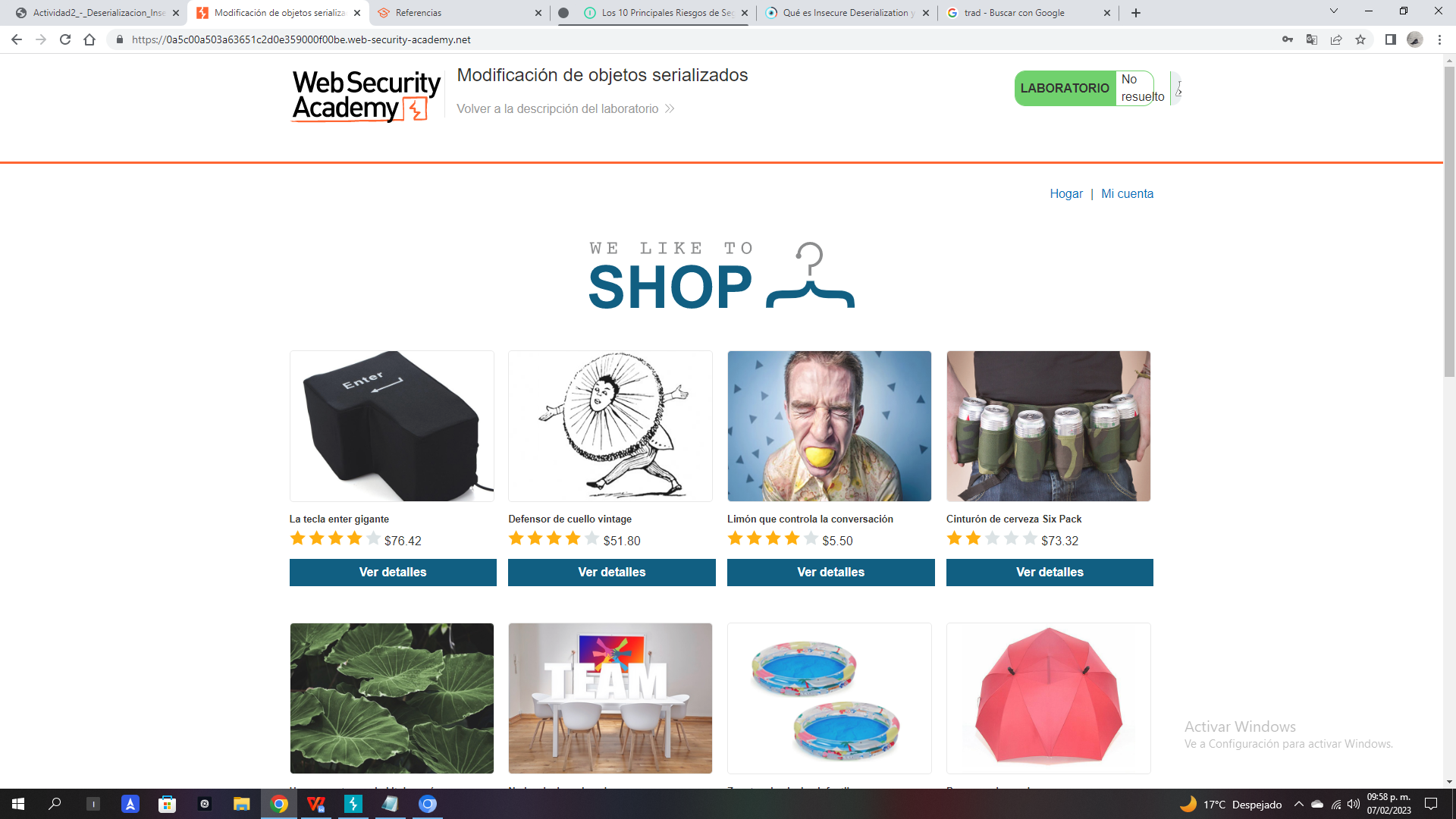
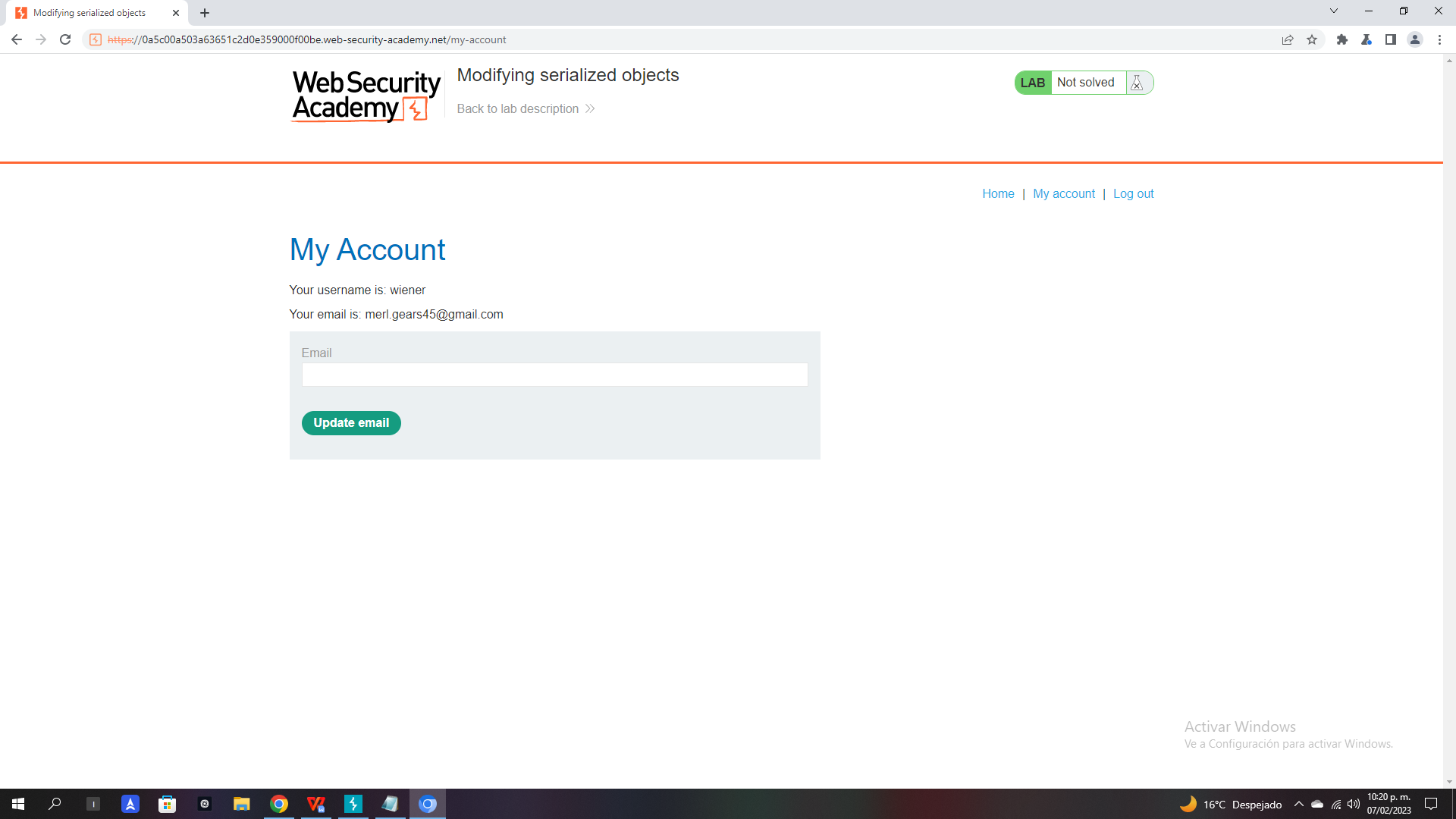
Desactivar la ejecución de código no confiable: Desactive la ejecución automática de código durante la deserialización para evitar la ejecución de código malicioso.

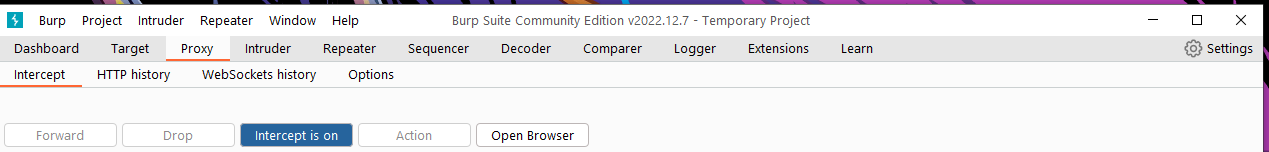
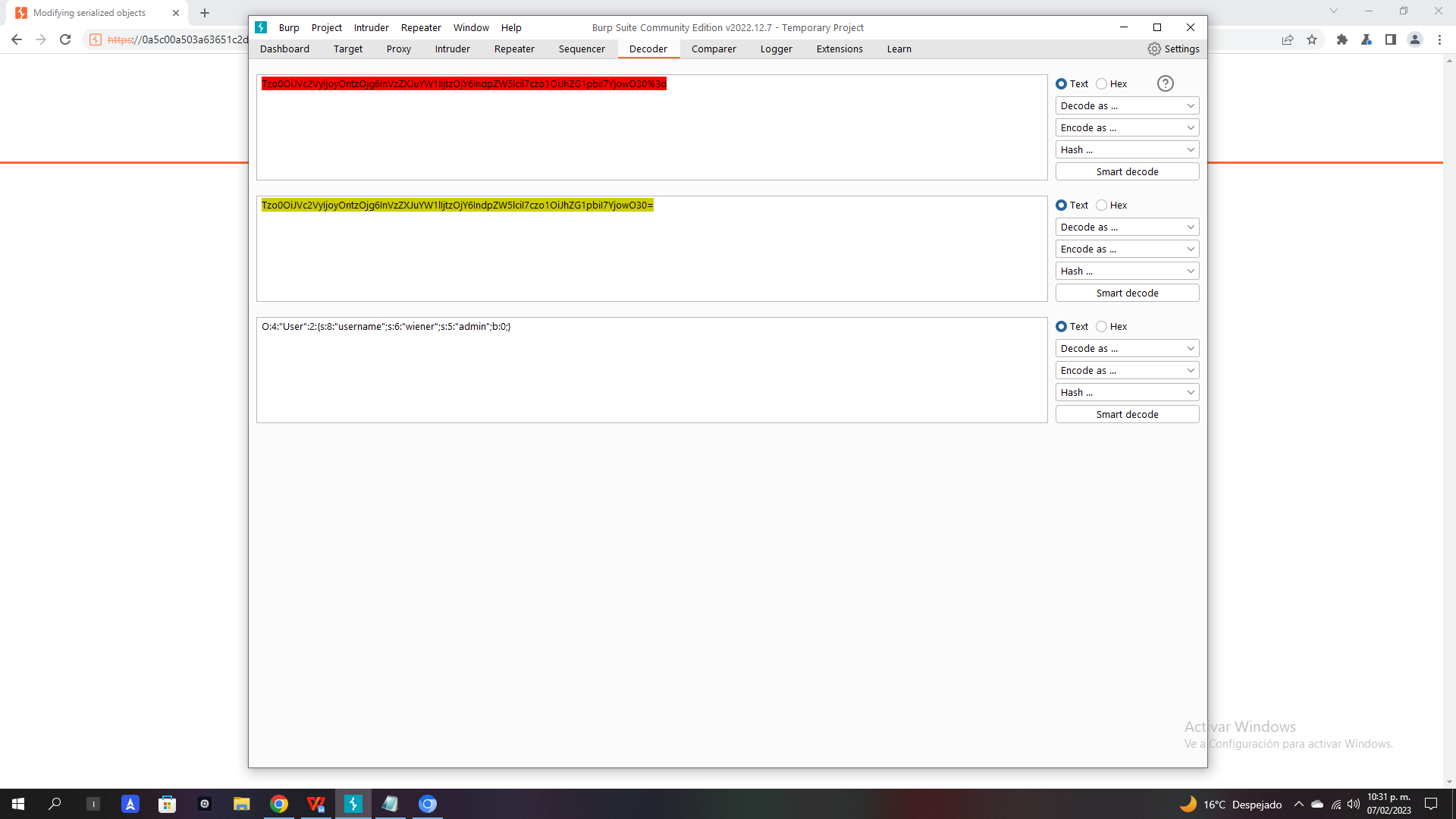
Verificación de firmas digitales: Verifique las firmas digitales de los objetos deserializados para asegurarse de que provienen de una fuente confiable.

Monitoreo y registro: Monitoreo y registro de las actividades relacionadas con la deserialización para detectar posibles ataques y tomar medidas para maximizar-los.

Es importante destacar que la prevención de la deserialización insegura es un proceso continuo y requiere la implementación de medidas de seguridad adecuadas, la revisión periódica de las prácticas de seguridad existentes y la actualización de las mismas en función de las nuevas amenazas y riesgos.

# *Ataque al sitio*



**

# *Conclucion*

En conclusión, la deserialización insegura es un problema de seguridad crítico que puede afectar a los usuarios y el software. Si no se aborda adecuadamente, puede resultar en graves consecuencias, como la fuga de información confidencial, la ejecución de código malicioso y el control total del sistema por parte de un atacante. Por lo tanto, es importante que los desarrolladores aborden este problema de seguridad e implementen medidas de protección adecuadas en sus aplicaciones. Las medidas preventivas incluyen la validación de entrada, el uso de bibliotecas seguras, la desactivación de la ejecución de código no confiable, la verificación de firmas digitales y el monitoreo y registro de actividades relacionadas con la deserialización.

# *Referencias*

Sanchez, D. (2022, 16 noviembre). *Vulnerabilidad de deserialización insegura de Java en Apache MINA SSHD (CVE-2022-45047)*. S2 Grupo. <https://s2grupo.es/vulnerabilidad-de-deserializacion-insegura-de-java-en-apache-mina-sshd-cve-2022-45047/>

Cantelli, F. (2022, 27 octubre). *Qué es Insecure Deserialization y cómo prevenirla*. Hackmetrix Blog. <https://blog.hackmetrix.com/insecure-deserialization/>

1. (2019, 2 abril). *Los 10 Principales Riesgos de Seguridad para las aplicaciones WEB según OWASP*. El blog de Omatech. <https://www.omatech.com/blog/2019/04/01/riesgos-de-seguridad-owasp/>