Tipo de artículo: Artículo de revisión

Tendencias actuales de las vulnerabilidades y ataques XSS

Current trends in XSS vulnerabilities and attacks

Luz Angela Fernández Gutiérrez [1*], **https://orcid.org/0009-0008-7022-3273**
Leonardo Álvarez Valencia [2], **https://orcid.org/0009-0001-5069-099X**
Nicole Alvarez Matos [3], **https://orcid.org/0009-0004-2360-0663**
Henry Raúl González Brito [4], **https://orcid.org/0000-0002-3226-9210**
Yaimí Trujillo Casañola [5], **https://orcid.org/0000-0002-3138-011X**

[1] Facultad de Ingeniería, Universidad Santiago de Cali. Colombia. luz.ferandez00@usc.edu.co
[2] Facultad de Ingeniería, Universidad Santiago de Cali. Colombia. leonardo.alvarez00@usc.edu.co
[3] Facultad de Tecnologías Interactivas, Universidad de las Ciencias Informáticas. Cuba. nicoleam@uci.cu
[4] Dirección de Seguridad Informática, Universidad de las Ciencias Informáticas. Cuba. henryraul@uci.cu
[5] Facultad de Tecnologías Interactivas, Universidad de las Ciencias Informáticas. Cuba. yaimi@uci.cu

*** Autor para correspondencia: luz.ferandez00@usc.edu.co**

Resumen

El objetivo de la investigación fue explorar las vulnerabilidades y ataques *Cross Site Scripting* (XSS), visto desde las temáticas trabajadas en investigaciones recientes y los reportes internacionales. La metodología consistió en la revisión sistemática que permite conocer el estado actual abordada a partir de las siguientes interrogantes: ¿Cuál es la tendencia en detecciones de ataques de XSS? ¿Cuáles son las principales técnicas para identificar este tipo de vulnerabilidad? ¿Cómo se manifiestan los ataques de XSS? ¿Qué herramientas se emplean para identificar este tipo de vulnerabilidad? ¿Cuáles son las medidas para mitigar que se explote este tipo de vulnerabilidad? Las bases de datos utilizadas fueron IEEE, Springer y Elsevier, tomando como período desde el año 2020 al 2024. Los resultados muestran que anualmente se reportan más de 100.000 ataques XSS según Vulners, cada año aproximadamente incrementa en un 18% y las principales técnicas para identificar las vulnerabilidades de XSS son: análisis estáticos, pruebas de penetración, pruebas pasivas, pruebas activas y pruebas de caja negra. Además de prácticas de implementación de cortafuegos en aplicaciones web con el empleo de las herramientas como Dalfox, QualysGuard, XSSER, S2XS2, DomXssMicro. Las principales medidas para mitigar la presencia este tipo de vulnerabilidades e impedir su explotación se orientaron a: implementar políticas de seguridad de contenido, cookies seguras, refuerzo de ataques de *Phishing* e implementación de WAF y frameworks de codificación a partir del tipo de XSS que se presente sea: reflejado, almacenado o DOM. Se concluye que los ataques XSS son una amenaza para la transformación digital.

Palabras clave: ataques; ciberseguridad; página; seguridad; scripts; vulnerabilidades; Web; XSS

Abstract

The objective of the research was to explore Cross Site Scripting (XSS) vulnerabilities and attacks, seen from the topics worked on in recent research and international reports. The methodology consisted of a systematic review that allows knowing the current state addressed from the following questions: What is the trend in XSS attack detections? What are the main techniques to identify this type of vulnerability? How do XSS attacks manifest themselves? What tools are used to identify this type of vulnerability? What are the measures to mitigate this type of vulnerability from being exploited? The databases used were IEEE, Springer and Elsevier taking the period from 2020 to 2024. The results show that more than 100,000 XSS attacks are reported annually according to Vulners, each year it increases by approximately 18% and the main techniques to identify XSS vulnerabilities are: static analysis, penetration testing, passive testing, active testing and black box testing. In addition to firewall implementation practices in web applications using tools such as Dalfox, QualysGuard, XSSER, S2XS2, DomXssMicro. The main

measures to mitigate the presence of this type of vulnerabilities and prevent their exploitation were aimed at: implementing content security policies, secure cookies, reinforcing Phishing attacks and implementing WAF and coding frameworks based on the type of XSS that is present, whether it is: reflected, stored or DOM. It is concluded that XSS attacks are a threat to digital transformation.

Keywords: *attacks*; *cybersecurity*; *page*; *security*; *scripts*; *threat*; *vulnerabilities*; *web*; *XSS*

Recibido: 01/07/2024
Aceptado: 14/07/2024
En línea: 19/07/2024

Introducción

En un mundo cada vez más digitalizado, las aplicaciones web abarcan amplios sectores de información sea de empresas, gobiernos, personas y un sinnúmero de datos. La tecnología se ha convertido en una modalidad global para acceder y expandir la información (Martín Martín 2024), por ende, las personas se ven obligadas a optar por medidas de respaldo en el diseño y utilización de aplicaciones web. Es importante el manejo de la información que permita implementar diferentes sistemas de seguridad, prevenga los ataques, los analice y así mismo mitigue las amenazas que se puedan presentar en aplicaciones web (Cano y Monsalve Machado 2023).

Estas plataformas se ven involucradas en varios ataques de ciberseguridad, como las inyecciones SQL, que ejecuta scripts para la inyección de datos y penetración de sistema, también están los ataques *man in the middle*, que captura los datos transferidos entre dos personas, redes o computadoras. Además, existen los ataques de *ransomware*, que implican que un agresor utilice un programa de *malware* para bloquear el acceso a los archivos de la víctima con el objetivo de extorsionar a la víctima para que pague un rescate a cambio de la recuperación de los datos y hay los ataques de *phishing* basados en la suplantación de identidad, donde una persona se hace pasar por una entidad legítima o confiable. El atacante envía enlaces maliciosos, ya sea por correo electrónico u otra vía, con el propósito de insertar malware en el dispositivo de la víctima (Castro-Maldonado y Villar-Vega 2021).

Dentro de los tipos de vulnerabilidades existentes en las aplicaciones web destaca los ataques de tipo *Cross Site Scripting* (XSS), en los cuales se inserta código malicioso en una aplicación web con el fin de comprometer la seguridad del mismo. En este tipo de ataque, los atacantes explotan las vulnerabilidades de la aplicación web para inyectar código malicioso a través de los cuadros de entrada de la aplicación, generalmente los fragmentos de script son realizados en JavaScript, y estos se incrustan en la aplicación web remota. Cuando el usuario visita de nuevo el sitio, el navegador cargará automáticamente y realizará la ejecución de estos scripts maliciosos. Este tipo de script

malicioso hará que el atacante obtenga cookies, tokens de sesión y otra información confidencial almacenada en el navegador o sitio web (Cui, Cui y Hu 2020).

El número de ataques XSS ha aumentado de 470 en 2011 a 22.000 en abril de 2022. Por ende, es crucial implementar medidas preventivas, como la validación adecuada de datos tanto en entrada como salida y el escape de caracteres, para mitigar estos riesgos y evitar pérdidas de gran impacto a futuro (Kumar y Ponsam 2023).

En primer lugar, el artículo explica el concepto de *Cross site scripting* y su posicionamiento en los últimos años. Luego, revisa los tipos de XSS y técnicas de ejecución más recientes. En tercer lugar, plantea soluciones en cuestión de buenas prácticas y aplicaciones para reconocer los ataques por adelantado, dando conocimiento sobre el alcance e impacto que puede generar este tipo de ataque, los cuales cada vez tienen más impacto y los sistemas de protección tradicionales como firewalls, Software antivirus, lista de control de acceso ya no son lo suficientemente eficaces.

El artículo se enfoca en el concepto de ataques XSS, planteando un foco de investigación con las siguientes preguntas:

- ¿Cuál es la tendencia en detecciones *Cross site scripting*?
- ¿Cuáles son las principales técnicas para identificar este tipo de vulnerabilidad?
- ¿Cómo se manifiestan los ataques XSS?
- ¿Qué herramientas se emplean para identificar este tipo de vulnerabilidades?
- ¿Cuáles son las medidas para mitigar que se explote este tipo de vulnerabilidad?

El objetivo de esta investigación es comparar diferentes herramientas de detección en términos de su capacidad para identificar vulnerabilidades XSS, además del impacto que ha generado en los últimos años dicho ataque y qué medidas implementar para mitigar los riesgos de ser víctima del *Cross Site Scripting*.

Materiales y métodos

Para el desarrollo de la investigación se combinaron los siguientes métodos:

Histórico – lógico: se empleó un análisis de tendencia de las vulnerabilidades y ciberataques desde el 2020 al 2024, teniendo en cuenta estudios recientes y reportes internacionales.

Analítico - sintético: se utilizó un análisis de los conceptos de vulnerabilidades y ataques de *Cross site scripting* como antecedentes de la investigación para desarrollar conocimientos sobre el impacto ocasionados por ciberataques.

Revisión sistemática a la bibliografía: se utilizó para reunir evidencia que demuestre la necesidad de la investigación, a partir de la revisión de las vulnerabilidades y ataques recogidos entre el 2020 y el 2024 teniendo en cuenta el manejo de bases de datos científicos como Elsevier, Springer, e IEEE Xplore.

Resultados y discusión

Los ataques de Cross-Site-Scripting se encuentran en el top 10 del ranking creado por la fundación OWASP en 2021 (Chinprutthiwong et al. 2020), que enumera las amenazas más graves en aplicaciones web. En la versión anterior del año 2017 se encontraba en el puesto número 7, pero en la del 2021 se añade a la categoría de inyección, ubicándose en la posición número 3. Según los estudios, en el 94% de las aplicaciones analizadas se detectó alguna forma de vulnerabilidad de inyección de código, con una tasa de incidencia máxima del 19% (Priyawati, Rokhmah y Utomo 2022).

Los XSS aprovecha las fallas existentes en la seguridad web, pues permite a los atacantes poder ejecutar scripts a modo de inyección de scripts, esto ocurre cuando el atacante emplea una aplicación web para enviar código malicioso, principalmente en secuencia de comandos del lado del navegador a un usuario final, el cual no tiene forma de saber que el script no es confiable y lo ejecutará debido a la proveniencia que es una fuente confiable para el usuario (Gupta y Chaudhary 2020). Los ataques XSS se manifiestan en tres tipos principales:

Ataque XSS reflejado

Este primer tipo de ataque se manifiesta del lado del cliente, su inyección se refleja fuera del servidor simulando ser parte de la solicitud que recibe el usuario, por lo general, este tipo de ataque busca entrada por medio de mensaje de correo o algún otra aplicación web, sus métodos de ejecución abarcan varios métodos, como el engaño al usuario para ejecutar un enlace, formularios maliciosos, navegación en páginas maliciosas, con el objetivo de ejecutar la carga de XSS en el navegador del usuario (Kaur, Garg y Bathla 2023).

Para evitar los ataques XSS reflejados, deben asegurarse de que todas las entradas del usuario se validen y desinfecten correctamente antes de que se muestran en la aplicación web. Esto puede hacerse mediante técnicas de validación de entradas, como el manejo de lista blanca y lista negra de caracteres o cadenas específicas, también deben utilizar técnicas de desarrollo adecuadas para codificar cualquier entrada de usuario que se muestre en la aplicación web (Rivera García 2023).

La Figura 1 representa un ejemplo de ataque de XSS reflejado, a través del spoofing de correo electrónico, donde un atacante puede convencer a un usuario para que pulse sobre un enlace dentro del mensaje. El enlace ejecuta un código JavaScript y redirigir al usuario a una aplicación web a la que tiene acceso. Si esa aplicación web presenta una vulnerabilidad XSS, permitirá la ejecución del código JavaScript reflejado a través de la redirección. De este modo, el navegador del usuario ejecutará el código dentro del contexto de confianza del dominio asociado con la aplicación y permitirá, por lo tanto, el envío de cookies o identificadores de sesión. Este envío se realizará sin violar la política de seguridad del navegador (Grau Reig 2021).

Figura 1. Escenario de un ataque XSS reflejado.

Fuente: Elaboración propia.

Ataque XSS almacenados

Existe un segundo ataque en este caso el atacante inyecta código HTML malicioso directamente en una página web o sitio vulnerable. Este ataque utiliza etiquetas de programación, como JavaScript, y los códigos se hacen permanentes en la aplicación web, afectando a todos los usuarios después de ejecutarse. Cada vez que alguien accede a una sección con el código inyectado, este se ejecuta en su navegador y realiza las acciones programadas. Este tipo de ataque es especialmente peligroso porque el código malicioso se inyecta en el contenido que se almacena en los servidores de aplicaciones web externas, de modo que los datos enviados por el atacante se almacenan de forma permanente en el servidor y se muestran a los usuarios cuando visitan el sitio, como se muestra en la Figura 2 (Vijayalakshmi y Syed Mohamed 2021).

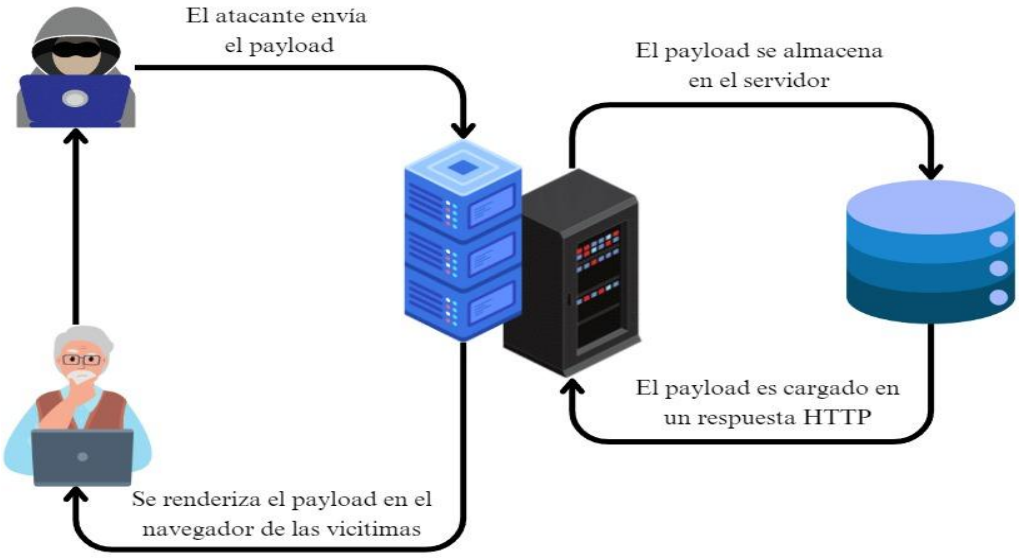

Figura 2. Escenario de un ataque XSS almacenado.

Fuente: Elaboración propia.

Para protegerse de los ataques XSS almacenados, es importante que las aplicaciones web limpien y validen correctamente todos los datos proporcionados por el usuario antes de almacenarlos o mostrarlos a otros usuarios. Para ello se pueden utilizar funciones y bibliotecas, por ejemplo, DOMPurify o Bleach, diseñadas para eliminar el código potencialmente malicioso y garantizar que sólo se acepten y almacenen entradas seguras y válidas. Se puede encontrar este tipo de XSS en secciones de comentarios de foros, feedback, descripciones o chats en vivo, entre otros recursos en donde se almacenen los parámetros que son ingresados por los usuarios (Pazos, Légaré y Beschastnikh 2023).

Ataque XSS basado en DOM

El tercer ataque consiste en el XSS basado en Modelo de Objetos del Documento (interfaz de programación que representa la estructura de un documento HTML o XML como un árbol de objetos), es un tipo de XSS donde la carga útil del ataque se ejecuta al modificar el entorno del DOM en el navegador de la víctima. A diferencia de otros tipos de XSS, en este caso, el código malicioso no se carga desde el servidor, sino que se inyecta a través de la URL y se ejecuta en el cliente, como se muestra en la Figura 3. Esto hace que el código del lado de la víctima se comporte de manera inesperada (Gupta y Chaudhary 2020).

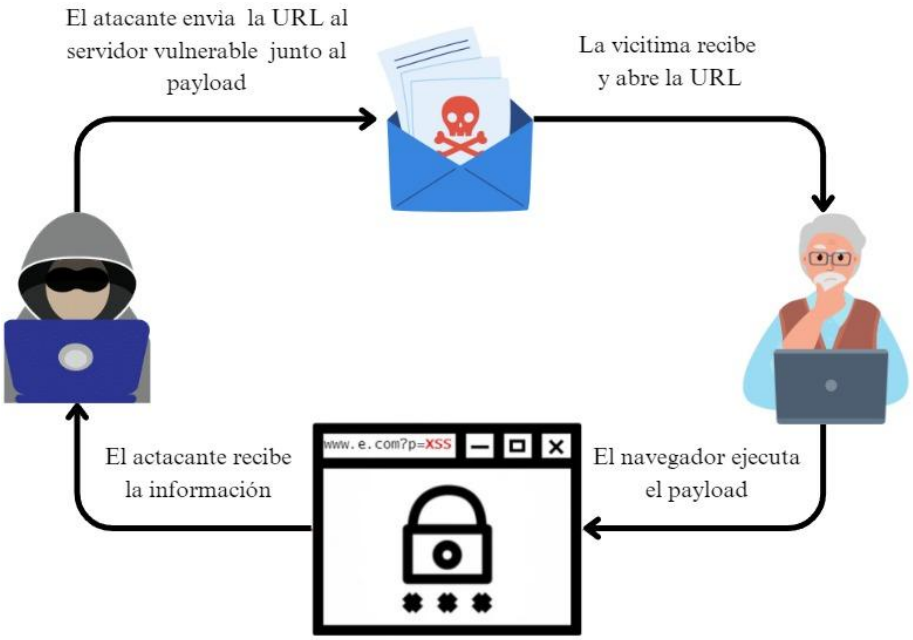

Figura 3. Escenario de un ataque XSS de tipo DOM.

Fuente: Elaboración propia.

Vulnerabilidades en XSS

La raíz de las vulnerabilidades XSS, radica en la incapacidad de las aplicaciones web para validar y manejar adecuadamente los datos no confiables antes de pasarlos al navegador. Uno de los enfoques para prevenir ataques XSS es usar expresiones regulares para identificar contenido malicioso. Sin embargo, este método no es infalible y se puede evitar utilizando herramientas de escaneo del lado del cliente y mecanismos de filtrado como NoScript (Rodríguez-Galán y Torres 2024).

Por lo tanto, es crucial evaluar la eficacia de la desinfección de la entrada del usuario e identificar cualquier vulnerabilidad XSS potencial, esta evaluación puede implicar la prueba de vulnerabilidades XSS a través de medios automáticos o manuales, incluido el uso de escáneres de vulnerabilidad y pruebas de penetración. El enfoque de métodos de prevención a ataques XSS, se centra en varios puntos como la protección de cookies, respaldar los puntos de entrada de datos, como registros donde los protocolos de uso son importantes como HTTP o HTTPS y el manejo de listas seguras. Existen diversas vulnerabilidades que permiten los ataques XSS. Entre ellas se encuentran las siguientes:

- **No validar la entrada de usuarios:** si una aplicación web no verifica que la entrada del usuario sea segura antes de utilizarla, un atacante podría inyectar código malicioso. Por ejemplo, verificar que no tenga el comando SCRIPT en ninguna de las variantes posibles (Nagarjun y Shaik 2020).
- **No implementar una política de seguridad de contenido (CSP):** esta política permite fijar una gran cantidad de restricciones, que suelen añadirse en las cabeceras de respuestas HTTP o en los archivos de configuración del servidor (Lodeiro 2022).
- **No implementar un firewall de aplicación web (WAF):** un WAF es una clase de firewall que ayuda a proteger un servidor web de los ataques cibernéticos (Apraez y Andrey 2022). Si no se llega implementar, la aplicación web serías más vulnerables a ataques de inyección como los XSS, comprometiendo así la seguridad, rendimiento y los dotas del usuario.

La detección de vulnerabilidades en aplicaciones web emplea varias técnicas destacadas por su efectividad. Una de las estrategias más importantes es la utilizada por OWASPGuard, que verifica la integridad de las solicitudes mediante un único token. Este token, que forma parte de un parámetro HTTP, se compara con el valor almacenado en el inicio de sesión del usuario, asegurando que las solicitudes sean legítimas (Zambrano et al. 2019).

El análisis estático de código, también conocido como auditoría de código fuente, es otro método crucial. Este enfoque no requiere la ejecución del programa y se centra en un análisis directo del código fuente para identificar posibles huecos de seguridad. A diferencia del análisis estático, el análisis dinámico de código se comunica con la aplicación web a través de su frontend. Su objetivo es identificar vulnerabilidades potenciales y debilidades en la arquitectura de la aplicación mediante la interacción directa con la misma.

Las pruebas de penetración representan una técnica más agresiva y activa, simulando ataques tanto de intrusos externos como internos malintencionados. Este proceso implica un análisis exhaustivo del sistema en busca de vulnerabilidades que pueden surgir debido a una configuración deficiente, fallos de hardware o software, o fallos operativos en procesos y contramedidas técnicas.

Por otro lado, las pruebas pasivas se centran en el análisis del tráfico de telecomunicaciones. Estas pruebas permiten detectar fallas y defectos de seguridad mediante el examen de los paquetes capturados, ofreciendo una visión detallada del tráfico que podría comprometer la seguridad. En contraste, las pruebas activas emplean un programador de subprocesos asignados al azar para verificar si las advertencias señaladas por un análisis predictivo del programa corresponden a errores reales.

Finalmente, las pruebas de caja negra consisten en estimular el sistema bajo prueba utilizando datos aleatorios o mutados deliberadamente. Este método busca detectar comportamientos no deseados, como la violación de la confidencialidad, poniendo a prueba la robustez y la seguridad del sistema en condiciones controladas pero impredecibles.

En la actualidad no solo se emplean técnicas para detectar estas vulnerabilidades, sino que también existen herramientas que facilitan el proceso de detección de vulnerabilidades. Algunas de esas herramientas son:

- **QualysGuard**: esta herramienta actúa como un escáner web. Su función es encontrar que los controles de un formulario de inicio de sesión se ingresen de la forma adecuada, por ejemplo, si un atacante hace una modificación a la URL del formulario, esta herramienta lo detectará, escaneará esta URL y detectará a qué tipo de vulnerabilidad pertenece (Cui, Cui y Hu 2020).
- **XSSER**: es una herramienta de seguridad para pruebas de penetración automáticas de vulnerabilidades XSS. Está diseñado específicamente para detectar y explotar vulnerabilidades de secuencias de comandos entre aplicaciones en diferentes aplicaciones, omita opciones específicas y filtre el código de inyección especial (Sarkar 2021).
- **Dalfox:** Es una herramienta escrita en Go que utiliza fuzzing, mutación y detección basada en el conocimiento del contexto para reconocer vulnerabilidades XSS (Pala et al. 2023).
- **S2XS2**: es una herramienta para detectar ataques XSS en el lado del servidor basados en el marco automático de los conceptos de inyección fronteriza y generación de políticas (Kadhim y Gaata 2020).
- **DomXssMicro**: es una herramienta que se construye a partir de plantillas extraídas de vulnerabilidades representativas y consta de seis componentes ortogonales: fuente, propagación, transformación, agregación, desencadenador y contexto. En DomXssMicro, hay 175 casos de prueba, cada uno destinado a probar atributos específicos del XSS basado en Dom (Panwar, Mishra y Patidar 2023).

Las consecuencias de un ataque XSS pueden ser graves. Los atacantes pueden robar información confidencial, como credenciales de inicio de sesión, cookies o datos personales. Además, pueden redirigir a los usuarios a aplicaciones web maliciosas o incluso tomar el control completo del navegador de la víctima (Rodríguez-Galán y Torres 2024). La formación en el campo de desarrollo de software seguro es crucial para garantizar que las aplicaciones sean robustas y resistentes a posibles ataques. Implementar herramientas de análisis de código, como escaneos estáticos o dinámicos, permite detectar vulnerabilidades antes de que lleguen a producción. Así, se pueden corregir y fortalecer los puntos débiles.

Conclusiones

El estudio realizado permitió constatar que los ataques de tipo XSS son muy recurrentes en las aplicaciones web actuales, lo que se refleja en la literatura consultada y en los principales reportes en internet como por ejemplo en el top 10 de OWASP.

Se identificaron herramientas como QualysGuard, XSSER, Dalfox, S2XS2, DomXssMicro para procesar y validar los datos proporcionados por los usuarios antes de almacenarlos o mostrarlos en la aplicación web.

Las medidas preventivas recomendadas para este tipo de ataques son: la implementación de técnicas de validación y desinfección de entradas de usuario, el uso de políticas de seguridad de contenido (CSP) y firewalls de aplicación web (WAF). Además, se deben emplear técnicas como los análisis estáticos, las pruebas de penetración, las pruebas pasivas, las pruebas activas y las pruebas de caja negra.

Los autores de esta investigación proponen continuar la investigación a través de la aplicación práctica en proyectos reales de las herramientas y medidas recomendadas como parte de esta revisión sistemática a la bibliografía. También se considera abordar enfoques de inteligencia artificial para la detección en tiempo real de ataques XSS.

Agradecimientos

Los investigadores agradecen a la Universidad Santiago de Cali por proporcionar una sólida formación durante el camino de desarrollo profesional. Sus docentes y compañeros fueron una fuente constante de inspiración y apoyo. También agradecer a la Universidad de las Ciencias Informáticas, por permitir cursar el diplomado que amplió los conocimientos y perspectivas en ciberseguridad y calidad de software.

Conflictos de intereses

Los autores no poseen conflictos de intereses.

Contribución de los autores

1. Conceptualización: Nicole Alvarez Matos, Leonardo Alvarez Valencia, Luz Angela Fernandez
2. Curación de datos: Nicole Alvarez Matos, Leonardo Alvarez Valencia, Luz Angela Fernandez
3. Análisis formal: Nicole Alvarez Matos, Leonardo Alvarez Valencia, Luz Angela Fernandez
4. Investigación: Nicole Alvarez Matos, Leonardo Alvarez Valencia, Luz Angela Fernandez
5. Metodología: Henry Raúl González Brito, Yaimí Trujillo Casañola

6. Supervisión: Henry Raul Gonzalez Brito, Yaimí Trujillo Casañola
7. Validación: Henry Raul Gonzalez Brito, Yaimí Trujillo Casañola
8. Visualización: Leonardo Alvarez Valencia
9. Redacción – borrador original: Nicole Alvarez Matos, Leonardo Alvarez Valencia, Luz Angela Fernandez Gutierrez, Henry Raul Gonzalez Brito, Yaimí Trujillo Casañola.
10. Redacción – revisión y edición: Nicole Alvarez Matos, Leonardo Alvarez Valencia, Luz Angela Fernandez Gutierrez, Henry Raul Gonzalez Brito, Yaimí Trujillo Casañola.

Financiamiento

La investigación no requirió fuente de financiamiento externa.

Referencias

apraez, L. y Andrey, J., 2022. Análisis de los WEB application firewall (WAF) como estrategia de ciberseguridad en las empresas con servicios WEB. [en línea], [consulta: 13 julio 2024]. Disponible en: https://hdl.handle.net/20.500.12494/46517.

Cano, W.D. y Monsalve Machado, S., 2023. *Ciberseguridad, reto empresarial para afrontar la era de la digitalización actual* [en línea]. bachelorThesis. S.l.: Escuela de Economía, Administración y Negocios. [consulta: 11 julio 2024]. Disponible en: https://repository.upb.edu.co/handle/20.500.11912/11318.

Castro-Maldonado, J. y VILLAR-VEGA, H., 2021. Análisis de riesgos y vulnerabilidades de seguridad informática aplicando técnicas de inteligencia artificial orientado a instituciones de educación superior. *Revista modum*, vol. 3, ISSN 2590-5430.

Chinprutthiwong, P., Vardhan, R., Yang, G. y GU, G., 2020. Security Study of Service Worker Cross-Site Scripting. *Proceedings of the 36th Annual Computer Security Applications Conference*. S.l.: s.n., pp. 643-654.

Cui, Y., Cui, J. y Hu, J., 2020. A Survey on XSS Attack Detection and Prevention in Web Applications. *Proceedings of the 2020 12th International Conference on Machine Learning and Computing* [en línea]. New York, NY, USA: Association for Computing Machinery, pp. 443-449. [consulta: 13 julio 2024]. ICMLC '20, ISBN 978-1-4503-7642-6. DOI 10.1145/3383972.3384027. Disponible en: https://doi.org/10.1145/3383972.3384027.

Grau Reig, P., 2021. *Ataques y vulnerabilidades web* [en línea]. PhD Thesis. S.l.: Universitat Politècnica de València. [consulta: 18 julio 2024]. Disponible en: https://riunet.upv.es/handle/10251/172833.

Gupta, B.B. y Chaudhary, P., 2020. *Cross-site scripting attacks: classification, attack, and countermeasures*. S.l.: CRC Press. ISBN 0-429-35132-1.

Kadhim, R. y Gaata, M., 2020. A hybrid of CNN and LSTM methods for securing web application against cross-site scripting attack. *Indones. j. electr. eng. comput. sci*, vol. 21, no. 2,

Kaur, J., Garg, U. y Bathla, G., 2023. Detection of cross-site scripting (XSS) attacks using machine learning techniques: a review. *Artificial Intelligence Review*, vol. 56, no. 11, ISSN 1573-7462. DOI 10.1007/s10462-023-10433-3.

Kumar, J.H. y Ponsam, J.G., 2023. Cross site scripting (XSS) Vulnerability detection using machine learning and statistical analysis. *2023 International Conference on Computer Communication and Informatics (ICCCI)*. S.l.: IEEE, pp. 1-9. ISBN 9798350348217.

Lodeiro, E.G., 2022. Análisis de vulnerabilidades de seguridad en páginas web. ,

Martín Martín, M.L., 2024. Inteligencia Artificial: Un estudio de su impacto en Ciberseguridad. En: Accepted: 2024-06-30T05:26:42Z [en línea], [consulta: 11 julio 2024]. Disponible en: https://openaccess.uoc.edu/handle/10609/150519.

Nagarjun, P.M.D. y Shaik, S.A., 2020. Cross-site scripting research: A review. *International Journal of Advanced Computer Science and Applications* [en línea], vol. 11, no. 4, [consulta: 17 julio 2024]. Disponible en: https://search.proquest.com/openview/cc8ef111cfe8338c73915fe372bfe070/1?pq-origsite=gscholar&cbl=5444811.

Pala, B., Pisu, L., Sanna, S.L., Maiorca, D. y Giacinto, G., 2023. A Targeted Assessment of Cross-Site Scripting Detection Tools. *ITASEC* [en línea]. S.l.: s.n., [consulta: 18 julio 2024]. Disponible en: https://ceur-ws.org/Vol-3488/paper26.pdf.

Panwar, P., Mishra, H. y Patidar, R., 2023. An analysis of the prevention and detection of cross site scripting attack. *International Journal* [en línea], vol. 11, no. 1, [consulta: 18 julio 2024]. Disponible en: https://www.academia.edu/download/97348792/ijeter051112023.pdf.

Pazos, J.C., Légaré, J.-S. y Beschastnikh, I., 2023. XSnare: application-specific client-side cross-site scripting protection. *Empirical Software Engineering*, vol. 28, no. 5, ISSN 1573-7616. DOI 10.1007/s10664-023-10323-w.

Priyawati, D., Rokhmah, S. y Utomo, I.C., 2022. Website vulnerability testing and analysis of website application using OWASP. *International Journal of Computer and Information System (IJCIS)*, vol. 3, no. 3, ISSN 2745-9659.

Rivera García, A., 2023. Sistema para determinar si un código fuente es vulnerable a XSS (cross-site scripting). [en línea]. [consulta: 11 julio 2024]. Disponible en: https://oa.upm.es/72772/.

Rodríguez-Galán, G. y Torres, J., 2024. Personal data filtering: a systematic literature review comparing the effectiveness of XSS attacks in web applications vs cookie stealing. *Annals of Telecommunications* [en línea], ISSN 1958-9395. DOI 10.1007/s12243-024-01022-8. Disponible en: https://doi.org/10.1007/s12243-024-01022-8.

Sarkar, S., 2021. Detecting Vulnerabilities of Web Application Using Penetration Testing and Prevent Using Threat Modeling. En: P.K. MALLICK, A.K. BHOI, G.-S. CHAE y K. KALITA (eds.), *Advances in Electronics, Communication and Computing*. Singapore: Springer Nature, pp. 21-32. ISBN 9789811587528. DOI 10.1007/978-981-15-8752-8_3.

Vijayalakshmi, K. y Syed Mohamed, E., 2021. Case Study: Extenuation of XSS Attacks through Various Detecting and Defending Techniques. *Journal of Applied Security Research*, vol. 16, no. 1, ISSN 1936-1610. DOI 10.1080/19361610.2020.1735283.

Zambrano, A., Guarda, T., Valenzuela, E.V.H. y Quiña, G.N., 2019. Mitigation techniques for security vulnerabilities in web applications. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*,