ISBN: 978-628-96613-1-6. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2025.1.1.526

Evaluation of methodologies in web application security: A systematic literature review

Bances Quevedo, Jhair Julio[1]; Farroñay Quesquen, Alfredo[2]; Garcés Rosendo, Eduardo Jesús[3]
Osores-Granda, Oscar Enrique.[4]
[1,2,3,4] Universidad Tecnológica del Perú (UTP), Facultad de Ingeniería en Sistemas e informática, Lima – Perú,
U17206483@utp.edu.pe , U23250923@utp.edu.pe, C18503@utp.edu.pe, C22834@utp.edu.pe

Abstract*– The increasing reliance on web applications in organizations requires effective protection of sensitive data to maintain user trust. However, the diversity of methodologies to evaluate the security of these applications makes it difficult to select the most effective ones, exposing them to vulnerabilities such as SQL injection and Cross-Site Scripting attacks. This study aimed to analyze how static and dynamic analysis methodologies, together with automated and manual tools, contribute to identifying and mitigating these vulnerabilities. Through a systematic review of the literature, structured under the PICO technique, searches were carried out in databases such as Scopus, obtaining 1,279 initial documents. Through a PRISMA flowchart and considering the inclusion and exclusion criteria, 53 final studies were selected for analysis. The results highlight the need to develop standardized criteria that facilitate the choice of more effective methodologies to guarantee the security of web applications. However, a lack of consensus on optimal approaches was identified, representing a significant challenge for security professionals. In conclusion, although there are promising tools and methods, the diversity and absence of standardization limit their practical implementation, evidencing the importance of new research to close these gaps and move towards safer web environments.

***Keywords*-- cyberattacks, vulnerabilities, SQL injection, Cross-Site Scripting, methodologies.**

Evaluación de Metodologías en la seguridad de aplicaciones web: Una revisión sistemática de la literatura

Bances Quevedo, Jhair Julio[1]; Farroñay Quesquen, Alfredo[2]; Garcés Rosendo, Eduardo Jesús[3]
Osores-Granda, Oscar Enrique.[4]
[1,2,3,4] Universidad Tecnológica del Perú (UTP), Facultad de Ingeniería en Sistemas e informática, Lima – Perú,
U17206483@utp.edu.pe , U23250923@utp.edu.pe, C18503@utp.edu.pe, C22834@utp.edu.pe

Resumen – La dependencia creciente de las aplicaciones web en las organizaciones exige una protección efectiva de datos confidenciales para mantener la confianza de los usuarios. Sin embargo, la diversidad de metodologías para evaluar la seguridad de estas aplicaciones dificulta la selección de las más eficaces, exponiéndolas a vulnerabilidades como inyección SQL y ataques Cross-Site Scripting. Este estudio tuvo como objetivo analizar cómo las metodologías de análisis estático y dinámico, junto con herramientas automatizadas y manuales, contribuyen a identificar y mitigar estas vulnerabilidades. Mediante una revisión sistemática de la literatura, estructurada bajo la técnica PICO, se realizaron búsquedas en bases de datos como Scopus, obteniendo 1,279 documentos iniciales. A través de un diagrama de flujo PRISMA y considerando los criterios de inclusión y exclusión, se seleccionaron 53 estudios finales para su análisis. Los resultados resaltan la necesidad de desarrollar criterios estandarizados que faciliten la elección de metodologías más efectivas para garantizar la seguridad de las aplicaciones web. No obstante, se identificó una falta de consenso sobre los enfoques óptimos, lo que representa un desafío significativo para los profesionales de la seguridad. En conclusión, aunque existen herramientas y métodos prometedores, la diversidad y ausencia de estandarización limitan su implementación práctica, evidenciando la importancia de nuevas investigaciones para cerrar estas brechas y avanzar hacia entornos web más seguros.

Palabras Clave-- ciberataques, vulnerabilidades, inyeccion SQL, Cross-Site Scripting, metodologías.

I. INTRODUCCIÓN

En la era digital actual, las aplicaciones web han pasado a ser componentes esenciales para las operaciones de muchas organizaciones, lo que le convierte en objetivos prioritarios para los ciberdelincuentes. Estas aplicaciones, que dependen en gran medida de bases de datos, son particularmente vulnerables a ataques como la inyección SQL y el cross-site scripting, los cuales permiten la inyección de código malicioso, comprometiendo la integridad de los datos y la funcionalidad de las aplicaciones. Ante la creciente sofisticación de estos ataques, las metodologías tradicionales de seguridad, como las pruebas de penetración y los escaneos automatizados, si bien útiles para identificar vulnerabilidades, resultan ser insuficiente al enfrentarse a amenazas complejas que requieren de un análisis más exhaustivo.

En respuesta a este panorama, se han implementado técnicas avanzadas, como el aprendizaje profundo y los sistemas de detección de intrusiones (IDS), así como firewalls de aplicaciones web (WAF), que analiza patrones de tráfico en tiempo real para bloquear ataques potenciales [1]. Sin embargo, estas soluciones tecnológicas no han eliminado la necesidad de contar con metodologías más completas y efectivas que aborden la seguridad desde un enfoque holístico, capaz de mitigar tanto las amenazas conocidas como las emergentes. La creciente adopción de estándares internacionales, como la norma ISO/IEC 27001, subraya la importancia de establecer un-Sistema de Gestión de Seguridad de la Información (SGSI), garantizando la confidencialidad, integridad y disponibilidad de los datos [2]. Sin embargo, el reto radica en como implementar estas normativas de manera efectiva en un entorno de amenazas en constante evolución, lo que resalta la necesidad de analizar y evaluar nuevas metodologías de seguridad. Ante este contexto surge la necesidad de realizar una Revisión Sistemática de la Literatura (RSL) que permita identificar las brechas existentes en las metodologías actuales de seguridad para aplicaciones web.

Este estudio tiene como objetivo evaluar como las metodologías de análisis estático y dinámico, junto con las herramientas de seguridad automatizadas y manuales, influyen en la eficacia para identificar y mitigar vulnerabilidades. Además, se pretende señalar las principales deficiencias en las metodologías actuales, con el fin de proponer enfoques innovadores que respondan a las crecientes demandas de ciberseguridad.

El artículo se organiza en cuatro secciones principales: la metodología(Sección II), donde se emplea la técnica PICO para formular preguntas de investigación y el protocolo PRISMA para la selección de artículos, detallando criterios de inclusión y exclusión; los resultados (Sección III), que sintetizan los hallazgos de una revisión sistemática de literatura, respondiendo a las preguntas de investigación basadas en el enfoque PICO; la discusión (Sección IV), que compara los hallazgos con estudios previos, analiza sus implicaciones en la seguridad en aplicaciones web y examina las fortalezas y limitaciones de las metodologías evaluadas; y las conclusiones (Sección V), donde se resumen los principales resultados y se ofrecen recomendaciones para investigaciones futuras. Finalmente, en la sección V presenta las conclusiones y recomendaciones resaltando las metodologías y herramientas que han mostrado mayor efectividad y proponiendo posibles direcciones para investigaciones futuras en este campo.

II. METODOLOGÍA

La Revisión Sistemática de Literatura (RSL) es una metodologia que permite realizar una búsqueda exhaustive y un análisis estructurado de información cientifica, basada en diversas metodologías que aseguran un alto nivel de calidad y objetividad en los datos obtenidos. Esta revision se centra en identificar, evaluar y combinar información de artículos científicos para responder a una pregunta de investigación específica. En este contexto, se aplicó la técnica PICO, la cual estructura la pregunta de investigación en componentes específicos: Problema (Patinet), Intervención (Intervention), Comparación (Comparation) y Resultado (Outcome), lo que permite una evaluación más detallada y organizada de la literature, ver tablas 1 y 2.

TABLA I
PREGUNTAS ORIENTADORA PICO Y SUBPREGUNTAS

RQ: ¿Cómo influyen las metodologías y herramientas de seguridad, comparando análisis estático y dinámico, así como herramientas automatizadas y manuales, en la eficacia para identificar y mitigar vulnerabilidades en aplicaciones web?
RQ1: ¿Cuáles son las principales vulnerabilidades de seguridad que se presentan en aplicaciones web desarrolladas en entornos de software?
RQ2: ¿De qué manera las metodologías de seguridad como el análisis estático y dinámico influyen en la detección de vulnerabilidades?
RQ3: ¿Cómo se comparan las herramientas automatizadas y manuales en términos de precisión para identificar y mitigar vulnerabilidades de seguridad?
RQ4: ¿Cuál es el impacto de las metodologías y herramientas de seguridad en la eficacia de la identificación y mitigación de vulnerabilidades en aplicaciones web?

TABLA II
COMPONENTES PICO Y PALABRAS CLAVE

P	Identificación de vulnerabilidades de seguridad en aplicaciones web desarrolladas en entornos de software.	"Vulnerabilites" OR "Cybersecurity" OR "Security failures"
I	Metodologías y herramientas de seguridad.	“Static Analysis” OR “Testing”.
C	Efectividad del análisis estático y dinámico.	“Detect vulnerabilities” OR “Automation” OR “Tools” OR “Accuracy of automated systems”
O	Optimización de la corrección de vulnerabilidades de seguridad.	“Mitigating risks” OR “Impact” OR “Minimizing vulnerabilidades” OR “Detecting failures”

A. Estrategia de Búsqueda

Partiendo de la tabla 2, se ha construido las ecuaciones de búsqueda, para SCOPUS, por lo cual la ecuación fue: (TITLE-ABS-KEY ("Vulnerabilities" OR "Cybersecurity" OR "Security failures") AND TITLE-ABS-KEY ("Methodologies" OR "Dynamic Analysis" OR "Static Analysis" OR "Testing") AND TITLE-ABS-KEY ("Detect vulnerabilities" OR "Automation" OR "Tools" OR "Accuracy of automated systems") AND TITLE-ABS-KEY ("Mitigating risks" OR "impact" OR "Minimizing vulnerabilities" OR "Detecting failures")).

B. Criterios de elegibilidad

Dentro del objetivo central del estudio, se aplicaron los siguientes criterios de exclusión e inclusión:

_ Criterios de inclusión:

- _ CI1:
  Se incluyeron estudios que identifiquen y analicen las principales vulnerabilidades de seguridad que se presentan en aplicaciones web desarrolladas en diferentes entornos de software (Java, PHP, Python, entre otros).
- _ CI2:
  Se incluyeron estudios que evalúen la efectividad de las metodologías de seguridad como el análisis estático y dinámico en la detención de las vulnerabilidades de aplicaciones web.
- _ CI3:
  Se consideraron estudios que comparen herramientas automatizadas y manuales en términos de precisión para identificar y mitigar vulnerabilidades de seguridad en aplicaciones web.
- _ CI4:
  Se consideraron artículos que evalúen el impacto de metodologías y herramientas de seguridad en la eficacia de identificación y mitigación de vulnerabilidades en aplicaciones web.

_ Criterios de Exclusión:

- _ CE1:
  No se considerarón investigaciones publicados antes del año 2020.
- _ CE2:
  Todos los artículos no redactados en inglés o español no fueron considerados.
- _ CE3:
  Se omitieron todos los documentos que no fueran artículos originales.

C. Selección de estudios

Inicialmente, se extrajeron 1279 artículos de la base de datos de SCOPUS. Tras aplicar los criterios de elegibilidad y seguir el proceso de selección guiado por la metodología PRISMA, la muestra final se redujo a 53 artículos (ver figura 1).

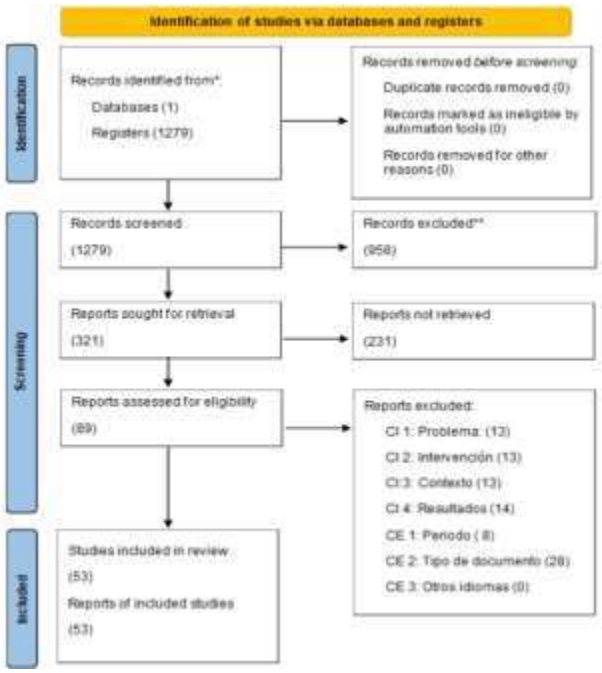

Fig 1. Diagrama de flujo PRISMA

III. RESULTADO

Una vez definidas las preguntas PICO para la búsqueda de literatura, se llevó a cabo el proceso de cribado según los criterios establecidos por la declaración PRISMA. Se seleccionaron 53 artículos relevantes de la base de datos, los cuales cumplían con los criterios de inclusión para el análisis de metodologías de seguridad en aplicaciones web. Estos artículos permiten realizar una evaluación detallada de las metodologías utilizadas para identificar y mitigar vulnerabilidades en entornos de software, comparando métodos de análisis estático y dinámico, y considerando tanto herramientas automáticas como manuales. Además, el proceso de selección facilitó la recopilación de datos clave para evaluar la efectividad, relevancia y viabilidad de estas metodologías en la mejora de la seguridad de aplicaciones web, alineando los resultados con los objetivos de esta revisión sistemática.

A. Resultados bibliométricos

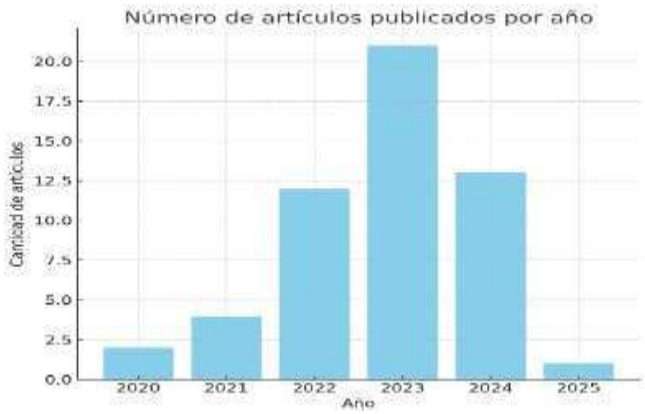

Fig. 2 Cantidad artículos publicados por año

Como se puede observar en la figura 2, el número de publicaciones muestra un aumento significativo desde 2020, alcanzando su punto máximo en 2023 con más de 20 artículos. Esto refleja un notable interés en el tema de seguridad en aplicaciones web y tecnologías emergentes durante ese año. En 2024, aunque la cantidad de publicaciones sigue siendo alta, presenta una ligera disminución, lo que sugiere una continuidad en la investigación, aunque con una ligera desaceleración. Por otro lado, los años 2020 y 2021 presentan un número reducido de artículos, lo que podría indicar que el tema no era tan popular en ese período.

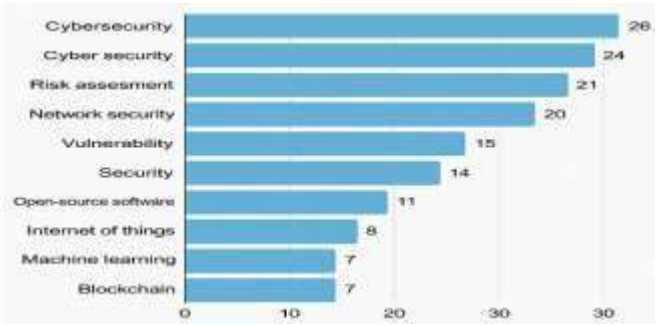

Fig. 3. Frecuencia de palabras clave de artículos

En la figura 3 se observan las palabras clave más mencionadas. "Cybersecurity" y "cyber security" destacan con 26 y 24 menciones, respectivamente, subrayando la importancia de la seguridad informática en la revisión. A continuación, "risk assessment" y "network security" aparecen con 21 y 20 menciones, destacando la relevancia de la evaluación de riesgos y la protección de redes. Otros conceptos relevantes incluyen "vulnerabilidad" y "seguridad", con 15 y 14 menciones, lo que refleja el interés en identificar y mitigar riesgos. Asimismo, términos como "automation" y "open- source software" (con 11 y 9 menciones) evidencian el creciente interés en métodos automatizados y plataformas de código abierto. Finalmente, términos como "internet of things", "machine learning" y "blockchain" resaltan la influencia de estas tecnologías en la protección de aplicaciones web.

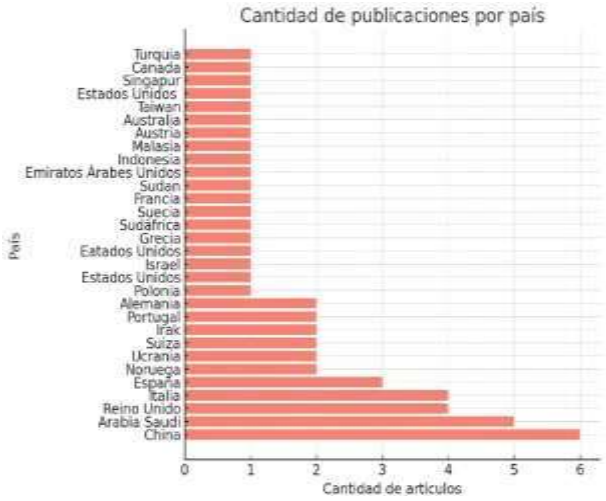

Fig. 4. Países con mayores artículos publicados

Como se observa en la figura 4, China y Arabia Saudita destacan por liderar el número de publicaciones, con 6 y 5 artículos respectivamente, lo que resalta su relevancia en la investigación sobre seguridad en aplicaciones web y tecnologías emergentes. A continuación, el Reino Unido e Italia, con 4 artículos cada uno, demuestran su significativa contribución en este campo. Otros países de interés, como España, Noruega y Ucrania, tienen 3 publicaciones cada uno, lo que indica un creciente interés en el tema. Además, Alemania, Portugal e Irak aportan 2 artículos cada uno, destacando sus implicaciones en el estudio. Finalmente, países como Estados Unidos, Sudáfrica y Suecia contribuyen con un artículo cada uno, lo que refleja un interés más amplio y diverso a nivel mundial en la investigación sobre seguridad en aplicaciones web. En total, se han revisado 53 artículos, proporcionando una visión global de las contribuciones internacionales en este ámbito.

B. Resultados académicos

RQ1: ¿Cuáles son las principales vulnerabilidades de seguridad que se presentan en aplicaciones web desarrolladas en entornos de software?

RQ 1.1 ¿Qué tipos de vulnerabilidades de seguridad se han identificado comúnmente en aplicaciones web desarrolladas en entornos de software?

En la Figura 5 se muestran las vulnerabilidades de seguridad más frecuentes en aplicaciones web, destacando varias amenazas críticas. Entre las más prevalentes se encuentran las inyecciones de código, como la inyección SQL, Cross-Site Scripting (XSS), falsificación de solicitudes entre sitios (CSRF), y la exposición de datos sensibles. También se identifican configuraciones de seguridad incorrectas, falta de control de acceso y problemas relacionados con la gestión de autenticación y sesiones [2]-[4], [20], [29], [39].Un tema destacado en la revisión de la literatura son las diversas amenazas de seguridad que afectan tanto a las aplicaciones web como a los sistemas de TI, entre las cuales se incluyen ataques de malware (como Stuxnet), ransomware (como WannaCry), inyecciones de datos falsos, errores de configuración y riesgos de phishing, así como la exposición a amenazas cibernéticas en dispositivos y redes [5]-[8], [27], [42], [43], [47]-[49]. Estas amenazas subrayan la gravedad de las vulnerabilidades más comunes en aplicaciones web, tales como la exposición de datos sensibles, configuraciones de seguridad incorrectas, fallas en la red y en la lógica de seguridad, así como deficiencias en los mecanismos de autenticación y autorización [12]-[15], [32]. Como resultado, las implicaciones de estas vulnerabilidades afectan tres aspectos fundamentales de la seguridad de la información: la disponibilidad, la integridad y la confidencialidad de los datos [33]-[35].

Fig. 5. Tipos de vulnerabilidades

RQ 1.2. ¿Qué factores contribuyen a la aparición de estas vulnerabilidades en el desarrollo de aplicaciones web en entornos de software?

Los factores que contribuyen a las vulnerabilidades en el desarrollo de aplicaciones web son diversos, e incluyen desde la falta de conocimientos en seguridad y la adopción de prácticas de codificación inseguras, hasta el uso de bibliotecas obsoletas y configuraciones inadecuadas. Además, la presión por cumplir plazos ajustados y la escasa inversión en ciberseguridad agravan aún más el problema [4], [14], [22], [26], [27], [31], [34], [53]. La falta de formación continua y la conciencia insuficiente en ciberseguridad entre desarrolladores y administradores también son factores cruciales [3], [13], [19], [25], [26], [30], [33], [43], [49], [50]. La prisa por cumplir con los plazos a menudo compromete la seguridad, afectando las pruebas, revisiones de código y auditorías [24], [28], [31], [39], [40]. Además, la creciente complejidad de las aplicaciones modernas y la dependencia de componentes de terceros incrementan el riesgo de vulnerabilidades, ya que pueden introducir fallos de seguridad imprevistos [4], [10], [14], [15], [27], [36].

Las configuraciones incorrectas o predeterminadas, sumadas a la falta de gestión de parches, dejan al software vulnerable ante amenazas [23], [29], [38], [44], [46]. Asimismo, la ausencia de un diseño seguro desde el inicio, que incluya protocolos modernos, control de acceso, autenticación y arquitecturas seguras, aumenta los riesgos [7], [16], [41], [45]. Esto se ve reflejado en la utilización de comunicaciones inseguras, como redes inalámbricas sin cifrado, y la vulnerabilidad en redes integradas de TI y OT, lo que amplía la superficie de ataque [6], [34], [42], [52]. La falta de datos actualizados, estrategias de protección robustas y controles adecuados en sistemas de inteligencia artificial, así como la insuficiencia de pruebas de seguridad, comprometen aún más la seguridad [1], [5], [8], [22], [37], [47]. Finalmente, una gestión deficiente de datos sensibles, control de acceso inadecuado y problemas en la autenticación permiten accesos no autorizados, aumentando el riesgo de explotación [9], [18], [21], [32], [51].

Fig. 6. Factores de la aparición de vulnerabilidades

RQ2. ¿De qué manera las metodologías de seguridad como el análisis estático y dinámico influyen en la detección de vulnerabilidades?

RQ 2.1. ¿Qué efectividad tienen las metodologías de análisis estático y dinámico en la detección de vulnerabilidades en aplicaciones web?

Los datos disponibles no abordan la efectividad de las metodologías de análisis estático y dinámico en la detección de vulnerabilidades en aplicaciones web, lo que limita su aplicabilidad en la evaluación de su rendimiento en este ámbito [1], [4], [8], [14], [27], [34], [41], [42], [44], [45], [48], [49]. La figura 7 ilustra que el análisis estático es eficaz para identificar vulnerabilidades en el código fuente en etapas tempranas, sin necesidad de ejecutar la aplicación; sin embargo, puede generar falsos positivos y no detecta problemas que solo se manifiestan en tiempo de ejecución. En contraste, el análisis dinámico es útil para detectar vulnerabilidades en entornos de ejecución reales, permitiendo la identificación de problemas que solo emergen mientras la aplicación está en funcionamiento.

La combinación de ambos enfoques mejora la cobertura en la detección de vulnerabilidades en aplicaciones web [2], [11], [12], [16], [18], [20], [21], [23], [26], [28], [29], [31]-[33], [36]-[38], [40], [43], [47], [52], [53]. Las metodologías de análisis estático y dinámico son complementarias y mejoran significativamente la detección de vulnerabilidades. Ambas son eficaces antes del despliegue final, siendo el análisis dinámico especialmente destacado para la identificación de ataques en entornos específicos. Aunque no siempre proporcionan una evaluación exhaustiva por separado, su uso combinado refuerza la seguridad al abordar diferentes tipos de vulnerabilidades, garantizando un enfoque integral en la protección del software [3], [10], [13], [15], [17], [19], [22], [30], [35], [39], [46], [50].

Fig. 7. Detección de vulnerabilidades

RQ 2.2. ¿Cuáles son las principales ventajas y desventajas del uso de análisis estático y dinámico en la detección de vulnerabilidades en comparación con otros enfoques?

La Figura 8 ilustra que el análisis estático y dinámico son enfoques que se complementan en la detección de vulnerabilidades en aplicaciones web, pero con características diferenciadas. El análisis estático destaca por su rapidez y capacidad de detectar vulnerabilidades sin necesidad de ejecutar el código, permitiendo una identificación temprana de errores estructurales y algoritmos defectuosos [2], [4], [5]. Sin embargo, su precisión se ve limitada por una alta tasa de falsos positivos [11] y su incapacidad para detectar problemas que emergen solo en tiempo de ejecución [12], [15], [53].

Por su parte, el análisis dinámico, aunque más lento y demandante en recursos, permite la detección precisa de vulnerabilidades que solo se manifiestan durante la ejecución real del software [25], [30], [32]. Ofrece una visión más realista de las amenazas, simulando escenarios de ataque efectivos [30], aunque depende de entornos de prueba bien configurados y puede generar falsos negativos si las condiciones no son óptimas [36], [40], [41], [47].

La combinación de ambos enfoques no solo amplía la cobertura de la detección de vulnerabilidades, sino que también compensa las limitaciones individuales de cada método [3], [10], [13], [30], [35]. No obstante, la revisión evidenció una falta de estrategias estandarizadas para integrar eficazmente ambos métodos, lo que representa una oportunidad significativa para futuras investigaciones en optimización de marcos de análisis híbrido.

Fig. 8. Análisis estático y dinámico

RQ 3. ¿Cómo se comparan las herramientas automatizadas y manuales en términos de precisión para identificar y mitigar vulnerabilidades de seguridad?

RQ 3.1. ¿Qué nivel de precisión ofrecen las herramientas automatizadas en comparación con las manuales para identificar vulnerabilidades de seguridad?

Los resultados de esta revisión sistemática de literatura destacan que las herramientas automatizadas son rápidas y eficientes para la detección de vulnerabilidades comunes en sistemas de software, proporcionando una amplia cobertura y análisis constante. Estas herramientas son particularmente útiles en evaluaciones a gran escala, ya que permiten identificar patrones conocidos y ejecutar pruebas regulares de manera repetitiva y eficaz. Sin embargo, los estudios revisados indican que presentan limitaciones significativas en términos de precisión, generando falsos positivos y negativos, y carecen de la capacidad para identificar vulnerabilidades complejas o específicas que requieren un análisis más detallado. Esta falta de profundidad se ha señalado como un desafío importante, particularmente en escenarios donde problemas sutiles podrían ser identificados únicamente mediante enfoques manuales [1], [5], [9], [18], [19], [28], [39], [44].

Por otro lado, la literatura analizada subraya que las herramientas manuales, aunque más lentas y costosas, son superiores en la identificación de vulnerabilidades complejas o contextuales. Los estudios destacan que estas herramientas son efectivas para abordar problemas específicos del sistema o relacionados con la lógica de negocio, aspectos que las herramientas automatizadas no logran cubrir de manera adecuada.

En síntesis, los resultados de esta revisión indican que las herramientas automatizadas son ideales para detectar vulnerabilidades comunes de manera rápida y eficaz en análisis a gran escala, mientras que las herramientas manuales ofrecen una mayor precisión en la identificación de problemas complejos. Los hallazgos sugieren que un enfoque combinado que integre ambas metodologías es la estrategia más efectiva para lograr una detección exhaustiva de vulnerabilidades, aprovechando la velocidad de las herramientas automatizadas y la precisión de las manuales.

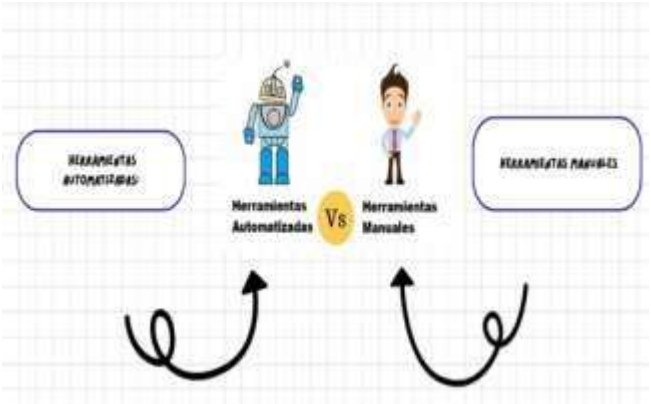

Fig. 9. Herramientas automatizadas y manuales

RQ 3.2. ¿Cuáles son las principales ventajas y desventajas de usar herramientas automatizadas frente a herramientas manuales en la mitigación de vulnerabilidades de seguridad?

La Figura 10 presenta una comparación clara entre las herramientas automatizadas y manuales en la mitigación de vulnerabilidades. Las herramientas automatizadas son valoradas por su rapidez, escalabilidad y capacidad para realizar escaneos masivos de código, permitiendo un monitoreo continuo eficiente en grandes proyectos o entornos dinámicos [5], [7], [8], [9], [17], [19], [20], [30], [38], [42], [43], [44]. Sin embargo, su eficiencia se ve afectada por una elevada tasa de falsos positivos y su limitada habilidad para identificar vulnerabilidades contextuales o específicas [3], [9], [13], [36], [50].

En contraste, las herramientas manuales, a pesar de su mayor coste y tiempo de implementación, permiten un análisis más profundo y contextualizado de las vulnerabilidades, ofreciendo alta precisión para detectar fallos complejos y específicos del entorno [3], [9], [20], [29], [39], [50]. Sin embargo, su dependencia de la pericia del analista introduce un riesgo adicional de errores humanos, aumentando los costos y tiempos en auditorías extensas [5], [19], [31], [33], [47].

Este contraste sugiere que ninguna estrategia es suficiente por sí sola. La revisión sistemática destaca la necesidad de enfoques híbridos que integren las capacidades de escaneo rápido de las herramientas automatizadas con el análisis detallado de las técnicas manuales para mejorar tanto la cobertura como la precisión en la mitigación de vulnerabilidades.

Además, se identificó una brecha importante en la literatura: existen pocos modelos que optimicen de manera conjunta el uso balanceado de herramientas automáticas y manuales en función del tipo de vulnerabilidad detectada, lo cual representa una oportunidad de investigación futura.

Fig. 10. Ventajas y desventajas

RQ 4. ¿Cuál es el impacto de las metodologías y herramientas de seguridad en la eficacia de la identificación y mitigación de vulnerabilidades en aplicaciones web?

RQ 4.1. ¿Qué metodologías de seguridad han demostrado ser más efectivas en la identificación de vulnerabilidades en aplicaciones web?

La figura 11 resume las metodologías de seguridad identificadas en esta revisión para detectar vulnerabilidades en aplicaciones web, agrupándolas según su enfoque y herramientas. El análisis estático (SAST), dinámico (DAST) y fuzzing son técnicas eficaces para identificar vulnerabilidades en código fuente y entornos en ejecución [9], [17], [29], [30], [32], [36], [40], [44], [46], [47], [51], [52]. Sin embargo, presentan limitaciones frente a vulnerabilidades complejas. Las pruebas de penetración (Pentesting) destacan al simular ataques reales, siendo efectivas para detectar fallos complejos que otras técnicas no abordan [3], [6], [15], [20], [21], [24], [41], [43], [53]. Los enfoques híbridos, que combinan análisis estático, dinámico, pentesting y revisiones manuales, ofrecen evaluaciones más completas y precisas [18], [30], [31], [34], [44], [53].

Modelos como **SDLC** y **DevSecOps** integran seguridad desde el inicio del desarrollo, previniendo vulnerabilidades y fortaleciendo cada fase del proceso [16], [20], [25], [39], [53]. Asimismo, marcos como **OWASP**, **STRIDE** y **ATT&CK de MITRE** estructuran la identificación y mitigación de riesgos, aportando estrategias claras [3], [10], [16], [33], [39].

Finalmente, tecnologías como inteligencia artificial y aprendizaje automático potencian estas metodologías al detectar patrones maliciosos y amenazas complejas en tiempo real, mejorando la precisión y eficiencia [2], [5], [8], [19], [35]. En conjunto, estas estrategias combinan prevención, análisis profundo y tecnologías avanzadas para optimizar la seguridad en aplicaciones web.

Fig. 11. Metodologías de seguridad

RQ 4.2. ¿Cómo han influido las herramientas de seguridad en la mitigación de vulnerabilidades en aplicaciones web en términos de eficiencia y efectividad?

La figura 12 muestra cómo la automatización en la detección y corrección de vulnerabilidades incrementa la eficiencia, permitiendo una identificación y mitigación rápida de riesgos. Estas herramientas facilitan el monitoreo continuo, pruebas más frecuentes y completas, y permiten a los equipos concentrarse en problemas complejos [2], [3], [5], [11], [15], [24], [25], [33], [39], [40], [51], [53]. Además, tecnologías de predicción y detección temprana identifican riesgos antes de que se conviertan en amenazas críticas, optimizando la prevención y minimizando impactos [17], [18], [21], [28], [29], [32], [41], [44].

Herramientas avanzadas, como simuladores y verificadores formales, refuerzan la resiliencia mediante análisis predictivo y simulaciones de fallos [7], [36], [52]. Otras soluciones automatizan pruebas, optimizan rutas críticas y mejoran la visibilidad, facilitando el cumplimiento normativo y respaldando decisiones estratégicas [9], [10], [14], [19], [20], [23], [35], [38], [51]. Sin embargo, la intervención humana sigue siendo clave para resolver vulnerabilidades avanzadas que exceden las capacidades de la automatización [39], [47].

Fig. 12. Herramientas de seguridad

VI. DISCUSIÓN

RQ1: La discusión sobre las vulnerabilidades en aplicaciones web destaca una serie de factores críticos que afectan la seguridad de estas plataformas. Las principales amenazas incluyen inyecciones de código como SQL Injection, Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF), así como la exposición de datos sensibles, configuraciones inseguras y fallos en los mecanismos de autenticación, autorización y control de acceso [2], [4], [20], [29], [39]. Además, factores como los plazos de entrega ajustados, prácticas de codificación inseguras, bibliotecas desactualizadas, la ausencia de protocolos modernos de seguridad y una gestión deficiente de parches contribuyen al incremento de estas vulnerabilidades [4], [14], [22], [27], [31], [34], [53]. La creciente complejidad de las aplicaciones modernas y la integración de componentes de terceros amplifican la superficie de ataque, lo que aumenta la exposición a amenazas [10], [14], [27], [36]. Además, la falta de capacitación continua y de pruebas de seguridad rigurosas

contribuye al mantenimiento de estas vulnerabilidades en los sistemas [5], [8], [22], [33], [35], [37]. Es esencial considerar también la interdependencia entre diferentes servicios y tecnologías, lo cual presenta nuevos desafíos para mitigar estas vulnerabilidades, especialmente con el creciente uso de arquitecturas basadas en microservicios y la nube.

RQ2: El análisis estático y dinámico, cuando se usan de forma complementaria, resultan ser dos enfoques eficaces para la detección de vulnerabilidades en aplicaciones web. El análisis estático permite identificar vulnerabilidades en el código fuente de manera temprana, sin necesidad de ejecutar la aplicación, lo cual es ventajoso para detectar problemas en fases iniciales del ciclo de desarrollo. Sin embargo, este enfoque puede generar falsos positivos y no es capaz de detectar vulnerabilidades que solo se manifiestan durante la ejecución de la aplicación [2], [4], [11], [15]. Por otro lado, el análisis dinámico es más adecuado para detectar vulnerabilidades en un entorno de ejecución real, como problemas que dependen del comportamiento de la aplicación en condiciones operativas reales. Aunque este método tiene la ventaja de detectar fallos que el análisis estático no puede, requiere de mayores recursos y de entornos de prueba más complejos [25], [30], [35], [40]. La combinación de ambos métodos es fundamental para proporcionar una cobertura más completa, ya que aborda tanto los errores estructurales del diseño como las vulnerabilidades que surgen solo durante la ejecución real [3], [10], [13], [30], [35]. Este enfoque híbrido es especialmente importante en la identificación de vulnerabilidades en aplicaciones de gran complejidad, donde se deben abordar tanto los problemas de diseño como los que dependen de la interacción con los usuarios.

RQ3: Las herramientas automatizadas y manuales ofrecen ventajas complementarias en la detección de vulnerabilidades. Las herramientas automatizadas, debido a su velocidad y capacidad para analizar grandes volúmenes de código rápidamente, son especialmente útiles en proyectos de gran escala y para realizar monitoreos continuos de seguridad. Sin embargo, estas herramientas pueden generar falsos positivos y tienen limitaciones en la detección de vulnerabilidades complejas, que requieren un análisis más detallado [1], [9], [19], [28], [39], [44]. Por otro lado, las herramientas manuales, aunque más lentas y costosas, ofrecen una mayor precisión en la identificación de vulnerabilidades complejas y de bajo nivel, lo que ayuda a reducir los falsos positivos. No obstante, estas requieren un mayor esfuerzo y recursos, así como un nivel de experiencia técnica significativo para su implementación [3], [20], [32], [39], [50]. Un enfoque combinado que integre herramientas automatizadas con la intervención manual optimiza tanto la velocidad como la precisión de la detección de vulnerabilidades, permitiendo cubrir un rango más amplio de problemas sin comprometer la efectividad [6], [15], [30], [50]. La integración de ambas metodologías resulta ser la mejor estrategia para abordar las vulnerabilidades de forma eficiente y precisa, especialmente cuando las amenazas son complejas y requieren una respuesta rápida y bien fundamentada.

RQ4: Las metodologías y herramientas de seguridad son fundamentales para identificar y mitigar vulnerabilidades en aplicaciones web, y su efectividad se maximiza cuando se combinan de forma estratégica. Técnicas como el análisis estático de código (SAST), el análisis dinámico (DAST), el fuzzing y las pruebas de penetración son esenciales para detectar vulnerabilidades tanto en el código fuente como en los entornos de ejecución [9], [30], [17], [44], [53]. Las metodologías híbridas, como el ciclo de vida del desarrollo de software (SDLC) y DevSecOps, integran prácticas de seguridad en todas las etapas del ciclo de desarrollo, garantizando que los aspectos de seguridad se aborden desde el inicio del proyecto [16], [20], [25], [39]. Además, tecnologías emergentes como la inteligencia artificial y el aprendizaje automático están mejorando significativamente la capacidad de detección y mitigación de amenazas, permitiendo análisis más rápidos y precisos [2], [5], [19], [35]. La automatización, por su parte, facilita la identificación rápida de vulnerabilidades y el monitoreo continuo, mientras que las herramientas predictivas ayudan a anticipar y prevenir ataques antes de que ocurran [7], [36], [52]. Sin embargo, la intervención manual sigue siendo crucial para abordar vulnerabilidades complejas y específicas que las herramientas automatizadas no pueden detectar con precisión, lo que demuestra la importancia de una estrategia que combine herramientas automatizadas y la experiencia humana para maximizar la efectividad de las soluciones de seguridad [39], [47].

IV. Conclusiones

Esta revisión sistemática examinó metodologías y herramientas utilizadas para garantizar la seguridad en aplicaciones web, con un enfoque particular en el análisis estático y dinámico, así como en la comparación entre herramientas manuales y automatizadas. El análisis estático es eficaz para identificar vulnerabilidades en el código antes del despliegue, como errores de sintaxis y configuraciones inseguras. Sin embargo, este método no aborda los problemas que pueden surgir durante la ejecución de la aplicación. Por otro lado, el análisis dinámico resulta más efectivo para identificar vulnerabilidades en tiempo real, como inyecciones SQL y ataques XSS, aunque requiere que la aplicación esté completamente operativa.

Las herramientas automatizadas destacan por su rapidez y capacidad para procesar grandes volúmenes de código, lo que las convierte en una opción excelente para fases tempranas del desarrollo o para monitoreo continuo. Sin embargo, las revisiones manuales y las pruebas de penetración siguen siendo esenciales para abordar vulnerabilidades más complejas y específicas, que no siempre pueden ser detectadas por herramientas automatizadas. Así, un enfoque híbrido que combine ambas metodologías, automatizadas y manuales, se muestra como el más efectivo para reducir los riesgos de seguridad de manera integral.

Las vulnerabilidades más comunes identificadas coinciden con las del OWASP Top 10, como la inyección de código y la autenticación rota. Además, la falta de marcos de desarrollo seguro desde las primeras etapas agrava estas debilidades. La incorporación de nuevas tecnologías, como APIs y microservicios, introduce nuevos desafíos que requieren estrategias de seguridad específicas y adaptadas a sus características.

Un enfoque integral que combine el análisis estático y dinámico, el uso de herramientas automatizadas y revisiones manuales, junto con la implementación de marcos de desarrollo seguro, optimiza la detección de vulnerabilidades, fortalece la mitigación de riesgos y mejora la resistencia frente a amenazas emergentes. Para futuras investigaciones, se recomienda explorar la integración de inteligencia artificial y aprendizaje automático para mejorar la precisión y eficiencia de los análisis de seguridad, así como el estudio de nuevas arquitecturas y tecnologías emergentes, como las relacionadas con la computación en la nube y la inteligencia distribuida, que podrían redefinir los enfoques de seguridad en aplicaciones web.

REFERENCIAS

[1] Tadhani, Jaydeep R; Vekariya, Vipul; Sorathiya, Vishal Alshathri, Samah; El-Shafai, Walid,"Securing web applications against XSS and SQLi attacks using a novel deep learning approach", [online]Reporte cientifico., Vol 14, Issue 1 December 2024.

[2] I. Coronel Suárez y D. Quirumbay Yagual, "Seguridad informática, metodologías, estándares y marco de gestión en un enfoque hacia las aplicaciones web", RCTU, vol. 9, n.º 2, pp. 97-108, dic. 2022.

[3] M. J. Page et al., "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," BMJ, vol. 372, n. 71, Mar. 2021. [Online]. Available: https://www.bmj.com/content/372/bmj.n71.

[4] Mulero-Palencia S.; Monzon Baeza V, "Detection of Vulnerabilities in Smart Buildings Using the Shodan Tool", 2023 España, Volume 12, Article number 4815, doi: 10.3390/electronics12234815.

[5] Medeiros I.; Neves N.; Correia M, "Statically Detecting Vulnerabilities by Processing Programming Languages as Natural Languages", Open Access, Volume 71, Issue 2, Pages 1033 -10561, 2022 Portugal, doi: 10.1109/TR.2021.3137314.

[6] Gao C.; Yang W.; Ye J.; Xue Y.; Sun J,"SGuard+: Machine Learning Guided Rule-Based Automated Vulnerability Repair on Smart Contracts", Open Access, Volume 33, Issue 54, 2024 China, Article number 114, doi: 10.1145/3641846.

[7] Dann A.; Plate H.; Hermann B.; Ponta S.E.; Bodden E, "Identifying Challenges for OSS Vulnerability Scanners-A Study & Test Suite ", IEEE Transactions on Software Engineering, Volume 48, Issue 9, Pages 3613 - 36251, 2022 Alemania, doi: 10.1109/TSE.2021.3101739.

[8] Samia N.; Saha S.; Haque A, "Predicting and mitigating cyber threats through data mining and machine learning ", Computer Communications, Volume 2281, Article number 107949,2024 Canada, doi: 10.1016/j.comcom.2024.107949.

[9] Amro A.; Oruc A.; Gkioulos V.; Katsikas S, "Navigation Data Anomaly Analysis and Detection", Volume 13, Issue, Article number 104,2022 Noruega , doi: 10.3390/info13030104.

[10] Roomi M.M.; Ong W.S.; Hussain S.M.S.; Mashima D, "IEC 61850 Compatible OpenPLC for Cyber Attack Case Studies on Smart Substation Systems", IEEE Access, Open Access, Volume 10, Pages 9164 - 9173, 2022 Singapur, doi: 10.1109/ACCESS.2022.3144027.

[11] Shepita P.; Tupychak L.; Shepita J, "Analysis of Cyber Security Threats of the Printing Enterprise", Journal of Cyber Security and Mobility, Open Access, Volume 12, Issue 3, Pages 415 - 434, 2023 Ucrania, doi: 10.13052/jcsm2245-1439.123.8.

[12] Adhikari A.; Kulkarni P, "Survey of techniques to detect common weaknesses in program binaries", Cyber Security and Applications, Open Access, Volume 3, Article number 100061, 2025 EE. UU, doi: 10.1016/j.csa.2024.100061.

[13] Jbair M.; Ahmad B.; Maple C.; Harrison R, "Threat modelling for industrial cyber physical systems in the era of smart manufacturing", Computers in Industry, Open Access, Volume 137, Article number 103611, 2022 Reino Unido, doi: 10.1016/j.compind.2022.103611.

[14] Hsiang H.; Chen Y.-Y, "Development of an Effective Corruption-Related Scenario-Based Testing Approach for Robustness Verification and Enhancement of Perception Systems in Autonomous Driving",Open Access Volume 24, Issue 1, Article number 301, 2024 Taiwán, doi: 10.3390/s24010301.

[15] Pugnetti C.; Björck A.; Schönauer R.; Casián C, "Towards Diagnosing and Mitigating Behavioral Cyber Risks", RisksOpen Access, Volume 12, Issue 7, Article number 116,2024 Suiza, doi: 10.3390/risks12070116.

[16] Patreliuk D.; Svoboda I.; Kalancha I.; Filashkin V.; Kachmar B,"Effective Use of Electronic Systems for International Exchange of Evidence in Criminal Investigations", Pakistan Journal of Life and Social Sciences, Volume 22, Issue 2, Pages 4811 - 4820, 2024 Ucrania, doi:10.57239/PJLSS-2024-22.2.00356.

[17] Enoch S.Y.; Huang Z.; Moon C.Y.; Lee D.; Ahn M.K.; Kim D.S, "HARMer: Cyber-Attacks Automation and Evaluation ", IEEE Access, Open Access, Volume 8, Pages 129397 - 129414, Article number 9142179, 2020 Australia, doi: 10.1109/ACCESS.2020.3009748.

[18] Erdődi L.; Zennaro F.M, "The Agent Web Model: modeling web hacking for reinforcement learning ", International Journal of Information Security, Open Access, Volume 21, Issue 2, Pages 293-309,2022 Noruega, doi: 10.1007/s10207-021-005547.

[19] Maia E.; Sousa N.; Oliveira N.; Wannous S.; Sousa O.; Praça I,"SMS-I: Intelligent Security for Cyber–Physical Systems ", Open Access, Volume 13, Issue 9, Article number 403 ,2022 Portugal, doi: 10.3390/info13090403.

[20] He Z.; Jia P.; Fang Y.; Liu Y.; Luo H, "SwitchFuzz: Switch Short-Term Goals in Directed Grey-Box Fuzzing", Applied Sciences, Open AccessVolume 12, Issue 21, Article number 11097, 2022 China, doi: 10.3390/app122111097.

[21] Altameem A.; Al-Ma'aitah M.; Kovtun V.; Altameem T, "A computationally efficient method for assessing the impact of an active viral cyber threat on a high-availability cluster ", Egyptian Informatics Journal, Open Access, Volume 24, Issue 1, Pages 61 - 69, 2023 Arabia Saudi, doi: 10.1016/j.eij.2022.11.002.

[22] Kareem M.I.; Jasim M.N, "Fast and accurate classifying model for denial-of-service attacks by using machine learning ", Bulletin of Electrical Engineering and Informatics, Open Access, Volume 11, Issue 3, Pages 1742 - 1751,2022 Irak, doi:10.11591/eei. v11i3.3688.

[23] Zhang F.; Fan L.; Chen S.; Cai M.; Xu S.; Zhao L, "Does Vulnerability Threaten Our Projects? Automated Vulnerable API Detection for Third-Party Libraries", IEEE Transactions on Software Engineering ,2022 China, doi: 10.1109/TSE.2024.3454960.

[24] Faisal Fadlalla F.; Elshoush H.T,"Input Validation Vulnerabilities in Web Applications: Systematic Review, Classification, and Analysis of the Current State-of-the-Art ", IEEE Access, Open Access, Volume 11, Pages 40128 - 401612023, 2023 Sudan, doi:10.1109/ACCESS.2023.3266385.

[25] Leithner M.; Garn B.; Simos D.E,"HYDRA: Feedback-driven black-box exploitation of injection vulnerabilities ", Information and Software Technology, Volume 140, Article number 106703, 2021 Austria, doi: 10.1016/j.infsof.2021.106703.

[26] Al Humaid Alneyadi M.R.M.; Normalini M.K, "FACTORS INFLUENCING USER'S INTENTION TO ADOPT AI-BASEDCYBERSECURITY SYSTEMS IN THE UAE ", Interdisciplinary Journal of Information, Knowledge, and Management, Open Access, Volume 18, Pages 459 - 486, 2023 Malasia, doi:10.28945/5166.

[27] Sánchez-Zas C.; Larriva-Novo X.; Villagrá V.A.; Rivera D.; Marín-Lopez A, "A methodology for ontology-based interoperability of dynamic risk assessment frameworks in IoT environments", Internet of Things, Open

Access, Volume 27, Article number 101267, 2024 España, doi: 10.1016/j.iot.2024.101267.
[28] Rizal R.; Selamat S.R.; Mas'ud M.Z.; Rahmatulloh A,"AResNet Model Using Deep Learning Approach for Enhancing the Internet of Things (IoT) Forensic Readiness Framework",International Journal of Intelligent Engineering and Systems, Volume 17, Issue 5, Pages 952 - 965, 2024 Indonesia, doi: 10.22266/ijies2024.1031.71.
[29] Abughali A; Alansari M.; Al-Sumaiti A.S.,"Deep Learning Strategies for Detecting and Mitigating Cyber-Attacks Targeting Water-Energy Nexus", IEEE Access, Open Access, Volume 12, Pages 129690 - 129704, 2024 Emiratos Árabes Unidos, doi:10.1109/ACCESS.2024.3458788.
[30] Mahmood B, "Prioritizing CWE/SANS and OWASP vulnerabilities: A network-based model", International Journal of Computing and Digital Systems, Volume 10, Issue 1, Pages 361 - 372 ,2021 Irak, doi:10.12785/ijcds/100137.
[31] Bitton R.; Maman N.; Singh I.; Momiyama S.; Elovici Y.; Shabtai A, "Evaluating the Cybersecurity Risk of Real-world, Machine Learning Production Systems", ACM Computing Surveys, Open Access, Volume 55, Issue 9, Article number 183, 2023 Israel, doi: 10.1145/3559104.
[32] Yaacoub J.-P.A.; Noura H.N.; Salman O.; Chehab A, "Ethical hacking for IoT: Security issues, challenges, solutions and recommendations", Internet of Things and Cyber-Physical Systems, Open Access, Volume 3, Pages 280 - 308, 2023 Francia, doi: 10.1016/j.iotcps.2023.04.002.
[33] Argyridou E.; Nifakos S.; Laoudias C.; Panda S.; Panaousis E.; Chandramouli K.; Navarro-Llobet D.; Zamorano J.M.; Papachristou P.; Bonacina S, "Cyber Hygiene Methodology for Raising Cybersecurity and Data Privacy Awareness in Health Care Organizations: Concept Study",Journal of Medical Internet Research, Open Access, Volume 25, Article number e41294, 2023 Suecia, doi: 10.2196/41294.
[34] Longo G.; Orlich A.; Musante S.; Merlo A.; Russo E, "MaCySTe: A virtual testbed for maritime cybersecurity",SoftwareX, Open Access, Volume,Article number 101426, 2023 Italia, doi: 10.1016/j.softx.2023.101426.
[35] Katsantonis M.N.; Manikas A.; Mavridis I.; Gritzalis D, "Cyber range design framework for cyber security education and training ", International Journal of Information Security, Open Access, Volume 22, Issue 4, Pages 1005 - 1027, 2023 Grecia, doi:10.1007/s10207-023-00680-4.
[36] Granata D.; Rak M.; Salzillo G.; Di Guida G.; Petrillo S, "Automated threat modelling and risk analysis in e-Government using BPMN",Connection Science, Open Access, Volume 35, Issue 1, Article number 2284645, 2023 Italia, doi:10.1080/09540091.2023.2284645.
[37] Iaiani M.; Tugnoli A.; Cozzani V,"Identification of cyber-risks for the control and safety instrumented systems: a synergic framework for the process industry", Process Safety and Environmental Protection, Open Access, Volume 172, Pages 69 - 82, 2023 Italia, doi: 10.1016/j.psep.2023.01.078.
[38] Su Y.; Xiong D.; Wan Y.; Shi C.; Zeng Q,"LinFuzz: Program-Sensitive Seed Scheduling Greybox Fuzzing Based on LinUCB Algorithm", IEEE Access, Open Access, Volume 12, Pages 74843 - 74860, 2024 China, doi:10.1109/ACCESS.2024.3404918.
[39] Simone F.; Akel A.J.N.; Gravio G.D.; Patriarca R, "Thinking in Systems, Sifting Through Simulations: A Way Ahead for Cyber Resilience Assessment",IEEE Access, Open Access, Volume 11, Pages 11430 - 11450, 2023 Italia, doi: 10.1109/ACCESS.2023.3241552.
[40] Hussain S.; Iqbal A.; Hussain S.M.S.; Zanero S.; Shikfa A.; Ragaini E.; Khan I.; Alammari R,"A novel hybrid methodology to secure GOOSE messages against cyberattacks in smart grids", Scientific Reports, Open Access, Volume 13, Issue 1, Article number 1857, 2023 Arabia Saudi, doi: 10.1038/s41598-022-27157-z.
[41] Piskachev G.; Becker M.; Bodden E, "Can the configuration of static analyses make resolving security vulnerabilities more effective? - A user study", Empirical Software Engineering, Open Access, Volume 28, Issue 5, Article number 118, 2023 Alemania, doi: 10.1007/s10664-023-10354-3.
[42] Zadeh A.; Lavine B.; Zolbanin H.; Hopkins D, "A cybersecurity risk quantification and classification framework for informed risk mitigation decisions", Decision Analytics Journal, Open Access, Volume 9, Article number 100328, 2023 Estados Unidos, doi: 10.1016/j.dajour.2023.100328.
[43] Ghanem M.C.; Chen T.M.; Ferrag M.A.; Kettouche M.E, "ESASCF: Expertise Extraction, Generalization and Reply Framework for Optimized Automation of Network Security Compliance", IEEE Access, Open Access, Volume 11, Pages 129840 - 129853, 2023 Reino Unido, doi: 10.1109/ACCESS.2023.3332834.
[44] Albalawi N.; Alamrani N.; Aloufi R.; Albalawi M.; Aljaedi A.; Alharbi A.R, "The Reality of Internet Infrastructure and Services Defacement: A Second Look at Characterizing Web-Based Vulnerabilities", Electronics, Open Access, Volume 12, Issue 12, Article number 2664, 2023 Arabia Saudi, doi: 10.3390/electronics12122664.
[45] Hussain S.M.S.; Aftab M.A.; Farooq S.M.; Ali I.; Ustun T.S.; Konstantinou C, "An Effective Security Scheme for Attacks on Sample Value Messages in IEC 61850 Automated Substations", IEEE Open Access Journal of Power and Energy, Open Access, Volume 10, Pages 304 - 315, 2023 Arabia Saudi, doi: 10.1109/OAJPE.2023.3255790.
[46] Gonzalez-Granadillo G.; Menesidou S.A.; Papamartzivanos D.; Romeu R.; Navarro-Llobet D.; Okoh C.; Nifakos S.; Xenakis C.; Panaousis E, "Automated cyber and privacy risk management toolkit", Sensors, Open Access, Volume 21, Issue 162, Article number 5493, 2021 España, doi: 10.3390/s21165493.
[47] Eyeleko A.H.; Feng T, "A Critical Overview of Industrial Internet of Things Security and Privacy Issues Using a Layer-Based Hacking Scenario",IEEE Internet of Things Journal, Open Access, Volume 10, Issue 24, Pages 21917 - 21941, 2023 China, doi: 10.1109/JIOT.2023.3308195.
[48] Lachkov P.; Tawalbeh L.; Bhatt S, "Vulnerability Assessment for Applications Security Through Penetration Simulation and Testing", Journal of Web Engineering, Volume 21, Issue 7, Pages 2187 - 2208, 2022, Estados Unidos, doi: 10.13052/jwe1540-9589.2178.
[49] Wang E.; Wang B.; Xie W.; Wang Z.; Luo Z.; Yue T, "EWVHunter: Grey-box fuzzing with knowledge guide on embedded web front ends", Applied Sciences, Open Access, Volume 10, Issue 11, Article number 4015, 2020 China, doi: 10.3390/app10114015.
[50] Filus K.; Domańska J, "Software vulnerabilities in TensorFlow-based deep learning applications", Computers and Security, Open Access, Volume 124, Article number 102948, 2023 Polonia, doi: 10.1016/j.cose.2022.102948.
[51] Alqahtani M.A, "Cybersecurity Awareness Based on Software and E-mail Security with Statistical Analysis", Computational Intelligence and Neuroscience, Open Access, Volume 2022, Article number 6775980, 2022 Arabia Saudi, doi: 10.1155/2022/6775980.
[52] Oriola O.; Adeyemo A.B.; Papadaki M.; Kotzé E, "A collaborative approach for national cybersecurity incident management", Information and Computer Security, Volume 29, Issue 3, Pages 457 - 484, 2021 Sudáfrica, doi: 10.1108/ICS-02-2020-0027.
[53] Kioskli K.; Dellagiacoma D.; Fotis T.; Mouratidis H,"The supply chain of a Living Lab: Modelling security, privacy, and vulnerability issues alongside with their impact and potential mitigation strategies", Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, Volume 13, Issue 2, Pages 147 - 182, 2022 Reino Unido, doi: 10.22667/JOWUA.2022.06.30.147.
[54] Sutter T.; Kehrer T.; Rennhard M.; Tellenbach B.; Klein J, "Dynamic Security Analysis on Android: A Systematic Literature Review" , IEEE Access, Open Access, Volume 12, Pages 57261 - 57287, 2024 Suiza, doi:10.1109/ACCESS.2024.3390612.
[55] Farrell M.; Bradbury M.; Cardoso R.C.; Fisher M.; Dennis L.A.; Dixon C.; Sheik A.T.; Yuan H.; Maple C, "Security-Minded Verification of Cooperative Awareness Messages", IEEE Transactions on Dependable and Secure Computing, Open Access, Volume 21, Issue 4, Pages 4048 - 4065, 2024 Reino Unido , doi: 10.1109/TDSC.2023.3345543.
[56] Balcioğlu Y.S.; Artar M, "The evolutıon of dıgıtal leadershıp: content and sentıment analysıs of the New York Tımes coverage ", Current Psychology, Open Access, Volume 43, Issue 28, Pages 23953 - 23970, 2024 Turquía, doi: 10.1007/s12144-024-06149-4.