https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 118 - 130

Integración de pruebas DAST en el desarrollo seguro de aplicaciones web

Integration of DAST testing into secure web application development

Información del reporte:

Para citar este reporte técnico:

Chavarría Fernández, R. (2026). Integración de pruebas DAST en el desarrollo seguro de aplicaciones web. *Cuadernos Técnicos Universitarios de la DGTIC, 4* (1), páginas (118 - 130). https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156

Ricardo Chavarría Fernández
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
rick@unam.mx
ORCID: 0009-0000-3817-6827

Resumen

Se llevó a cabo una serie de evaluaciones de seguridad sobre aplicaciones web institucionales en tiempo de ejecución, mediante la simulación de ataques reales utilizando el método de Pruebas Dinámicas de Seguridad de Aplicaciones (*Dynamic Application Security Testing*, DAST). El objetivo fue identificar vulnerabilidades durante la fase de pruebas del Ciclo de Vida del Desarrollo de Software (SDLC), antes de la implementación y de la realización de auditorías de seguridad. Para ello, se diseñó un entorno de evaluación controlado y se seleccionó y configuró la herramienta especializada OWASP Zed Attack Proxy (ZAP), con la que se realizaron escaneos automatizados y manuales para detectar vulnerabilidades comunes, tales como inyecciones de código malicioso (XSS) y configuraciones inseguras. Durante las pruebas, se identificaron hallazgos importantes, principalmente relacionados con errores en la validación de entradas y configuraciones por defecto. Los resultados se compararon con estándares actuales de seguridad web, lo que hizo posible establecer medidas de mitigación y prevención adecuadas. En conjunto, la integración de DAST durante la fase de pruebas del SDLC permitió fortalecer el aseguramiento continuo de la seguridad de los sistemas evaluados, al identificar vulnerabilidades en tiempo de ejecución antes de su despliegue en entornos productivos.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 119 - 130

Palabras clave: Análisis dinámico de seguridad, gestión de vulnerabilidades, OWASP-ZAP, seguridad web.

Abstract

A series of runtime security assessments were conducted on institutional web applications by simulating real-world attacks using Dynamic Application Security Testing (DAST). The objective was to identify vulnerabilities during the testing phase of the Software Development Life Cycle (SDLC), prior to implementation and security audits. To this end, a controlled testing environment was designed, and the specialized OWASP Zed Attack Proxy (ZAP) tool was selected and configured. Automated and manual scans were performed to detect common vulnerabilities, such as cross-site scripting (XSS) and insecure configurations. During the tests, significant findings were identified, primarily related to errors in input validation and default configurations. The results were compared with current web security standards, enabling the establishment of appropriate mitigation and prevention measures. Overall, the integration of DAST during the SDLC testing phase strengthened the ongoing security assurance of the evaluated systems by identifying runtime vulnerabilities before their deployment in production environments.

Keywords: *Dynamic security analysis, OWASP-ZAP, vulnerability management, web security.*

1. INTRODUCCIÓN

La proliferación y adopción de tecnologías en el desarrollo de software ha expandido significativamente los vectores de ataque en los entornos digitales, particularmente en aplicaciones web. Ante esta perspectiva, las organizaciones se han visto obligadas a adoptar estrategias de seguridad más sólidas para salvaguardar su información y asegurar los principios de confidencialidad, integridad y disponibilidad de la misma (International Organization for Standardization [ISO], 2022; National Institute of Standards and Technology [NIST], 2020).

En la práctica, las revisiones o auditorías de seguridad suelen realizarse al concluir el ciclo de desarrollo de software, lo que conduce al descubrimiento de hallazgos tardíos que pueden exponer a las organizaciones a riesgos innecesarios como la pérdida o robo de información, así como incrementar los costos de mantenimiento. Por esta razón, resulta prioritario adoptar enfoques —como DevSecOps o el Ciclo de Vida del Desarrollo de Software (SDLC) (NIST, 2022)— que integren la seguridad desde fases intermedias y continuas del ciclo de desarrollo, incentivando la evaluación periódica de riesgos y la aplicación proactiva de controles de seguridad desde la fase de diseño, desarrollo y pruebas, lo que contribuye a reducir la superficie de ataque y mejorar la resiliencia de las aplicaciones.

En el ámbito del aseguramiento continuo de los sistemas web, las Pruebas Dinámicas de Seguridad en Aplicaciones (DAST) (OWASP Foundation, 2023a) representan un proceso proactivo para identificar vulnerabilidades mientras las aplicaciones están en ejecución, interactuando directamente con las interfaces del sistema sin necesitar acceso al código fuente; además, permiten emular ataques reales desde la perspectiva de un usuario externo. Por esta razón, las DAST se han posicionado como un componente metodológico esencial dentro de las estrategias de seguridad de aplicaciones en línea (AppSec), particularmente en fases donde las aplicaciones ya cuentan con funcionalidad operativa,

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 120 - 130

complementando otras técnicas empleadas en el ámbito de la seguridad (Aydos & Baykara, 2022; Jennings, 2025; Dencheva, 2022).

En el entorno institucional, específicamente en la Dirección General de Cómputo y de Tecnologías de Información y Comunicación de la UNAM, ha sido necesario incorporar paulatinamente mecanismos de defensa que fortalezcan la seguridad de los sistemas de servicios web en la fase de pruebas. Por ello, entre 2023 y 2025, se realizaron evaluaciones dinámicas de seguridad en dos aplicaciones institucionales: el Sistema Institucional de Consultas Universitarias (SICU) y el Sistema de Votaciones Electrónicas (SVE), con el propósito de identificar vulnerabilidades y establecer acciones de mejora continua, sentando las bases para evaluaciones de seguridad más exhaustivas en etapas posteriores.

El objetivo principal consistió en implementar un proceso de Pruebas Dinámicas de Seguridad de Aplicaciones como parte del desarrollo seguro de sistemas web institucionales, mediante la ejecución de evaluaciones en entornos de pruebas controlados, con el fin de analizar el comportamiento de las aplicaciones en tiempo de ejecución, validar la eficacia de los controles de seguridad existentes y reducir riesgos antes de su despliegue en entornos productivos, sirviendo además como preparación para auditorías de seguridad internas y externas posteriores.

Adicionalmente, se buscó alinear dichas pruebas con estándares internacionales de seguridad, como OWASP Top 10 e ISO/IEC 27034 (OWASP Foundation, 2021; ISO, 2018); de esta manera, no sólo se fortaleció la seguridad de las aplicaciones, sino que también se impulsó la madurez del enfoque institucional hacia el desarrollo seguro de aplicaciones web en el equipo de trabajo.

2. DESARROLLO TÉCNICO

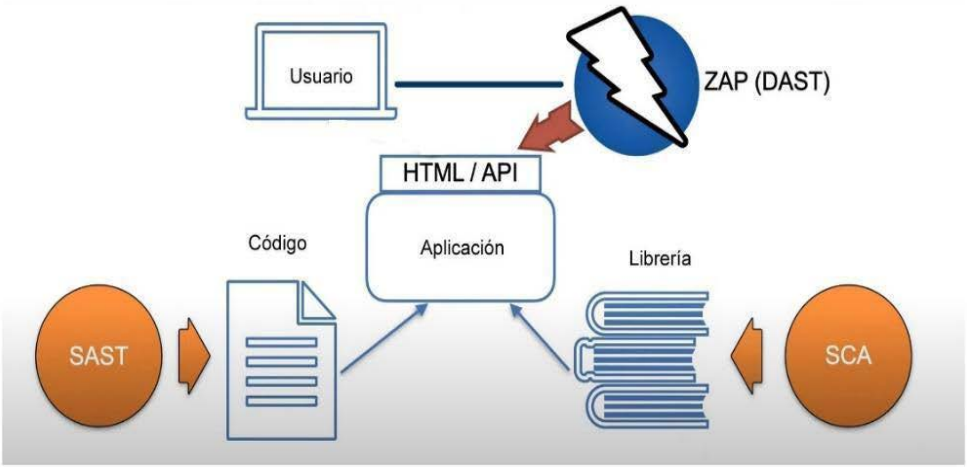
Para la planificación y ejecución de las pruebas, se empleó OWASP Zed Attack Proxy (OWASP ZAP), una herramienta gratuita de análisis dinámico reconocida por su capacidad para realizar escaneos activos, interceptar tráfico y detectar vulnerabilidades en tiempo de ejecución. Los análisis fueron configurados cuidadosamente para no afectar la disponibilidad del entorno: se inició con escaneos pasivos y, posteriormente, se ejecutaron pruebas activas autorizadas y pruebas manuales de comprobación (Chorell & Ekberg, 2024; Kondraciuk et al., 2022).

OWASP ZAP actúa como un proxy entre el navegador y la aplicación web, lo que permite interceptar e inspeccionar las solicitudes y respuestas para modificar su contenido y así reenviar los paquetes interceptados para descubrir vulnerabilidades y comportamientos inesperados; (véase la Figura 1).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 121 - 130

Figura 1

Intercepción e inspección de solicitudes y respuestas



Las evaluaciones automatizadas se complementaron con pruebas de penetración manuales (*pentesting*), las cuales simulan el comportamiento de un atacante externo con el fin de identificar fallos en los sistemas o en las comunicaciones, así como evaluar riesgos de filtración de datos, denegación de servicio o falsos positivos. Aunque las pruebas manuales pueden ser más lentas que los escaneos automáticos, proporcionan hallazgos de mayor precisión y ayudan a validar los mecanismos de defensa, los procedimientos de respuesta y el cumplimiento de políticas.

Después de identificar las vulnerabilidades, se priorizaron y aplicaron las correcciones necesarias, según su criticidad, mediante ajustes de configuración, mejoras en el código y refuerzo de controles. Posteriormente, se verificó la efectividad de estas correcciones con escaneos de verificación, empleando la misma herramienta de detección inicial. Para concluir, se documentaron los hallazgos y se proporcionaron recomendaciones adicionales para prevenir recurrencias y fortalecer la seguridad del sistema en desarrollo.

2.1 METODOLOGÍA

La metodología de pruebas dinámicas de seguridad se desarrolló siguiendo las guías y marcos establecidos por la OWASP Foundation, que representan estándares abiertos reconocidos en la industria para la seguridad de aplicaciones web. La estructura metodológica se organizó en seis etapas:

I. Planeación

En esta etapa, se definieron los objetivos del proceso de evaluación, el alcance del análisis dinámico y las restricciones operativas. Así mismo, se seleccionaron las pruebas a realizarse con la herramienta OWASP Zed Attack Proxy (ZAP).

Como marco de referencia para la identificación de vulnerabilidades, se adoptó el OWASP Top 10 2021, documento que establece las diez categorías de riesgos más críticos en aplicaciones web y que sirve como guía para priorizar esfuerzos de mitigación (OWASP Foundation, 2021); (véase la Figura 2).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 122 - 130

Figura 2

OWASP TOP 10 2021

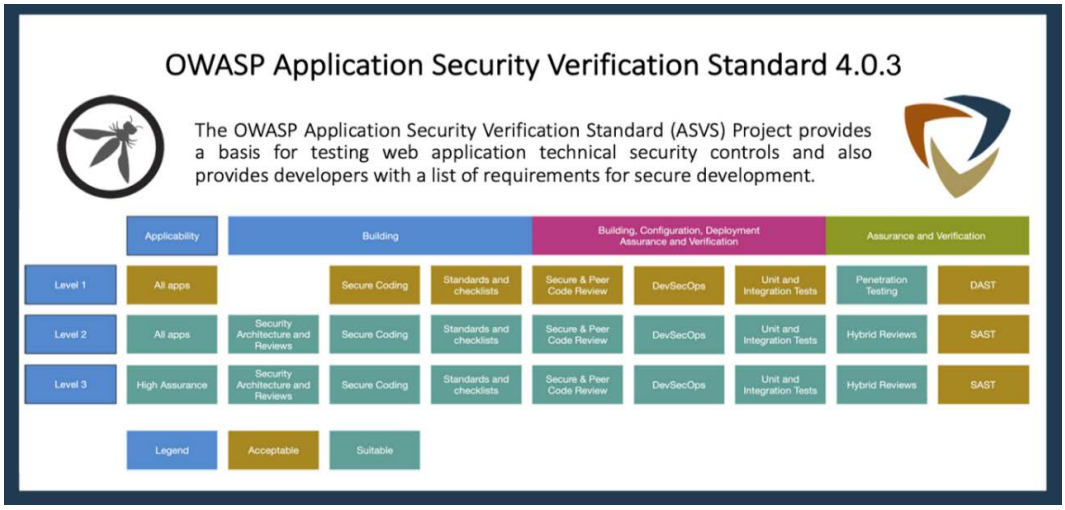
II. Preparación

La instalación de OWASP ZAP se realizó en un entorno aislado y controlado, utilizando máquinas virtuales para la aplicación y la base de datos a probar, lo que garantizó la reproducibilidad del análisis. Adicionalmente, se habilitaron extensiones de ZAP disponibles en su sitio oficial, lo que permitió ampliar la capacidad de exploración y generar reportes más específicos (OWASP Foundation, 2023a).

La selección de herramientas y la forma de despliegue se alinearon con el OWASP Application Security Verification Standard (ASVS) v4.0.3, que establece controles de seguridad requeridos para aplicaciones modernas y sugiere pruebas dinámicas en entornos dedicados (OWASP Foundation, 2023); véase la Figura 3.

Figura 3

OWASP Application Security Verification Standard v4.0.3



Nota. Tomado de https://blog.secureflag.com/2024/11/15/understanding-owasp-asvs/

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 123 - 130

III. Descubrimiento

El entorno de pruebas se diseñó para replicar condiciones reales de operación sin comprometer la disponibilidad del sistema. Se configuró OWASP ZAP con parámetros ajustados a las necesidades del aplicativo, incluyendo:

- Perfiles de escaneo pasivo y activo
- Credenciales de prueba con permisos controlados
- Exclusión de ciertos recursos sensibles para evitar interrupciones
- Límites de solicitudes y profundidad de análisis

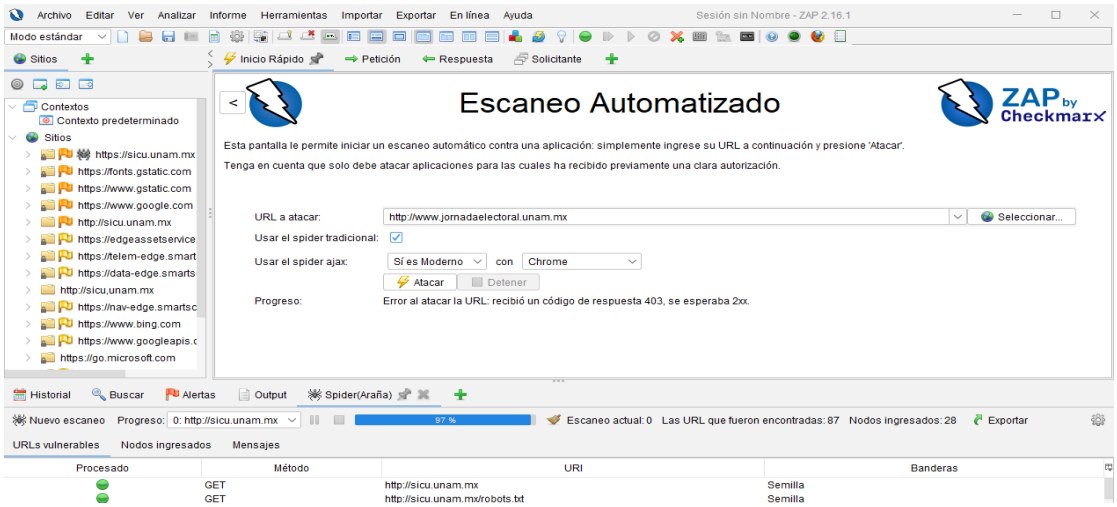
IV. Ejecución de pruebas

Las pruebas se ejecutaron en dos modalidades:

1. Escaneo activo, en el que se simularon ataques controlados para confirmar vulnerabilidades detectadas durante la etapa pasiva (véase la Figura 4).

Figura 4

Escaneo automatizado ZAP

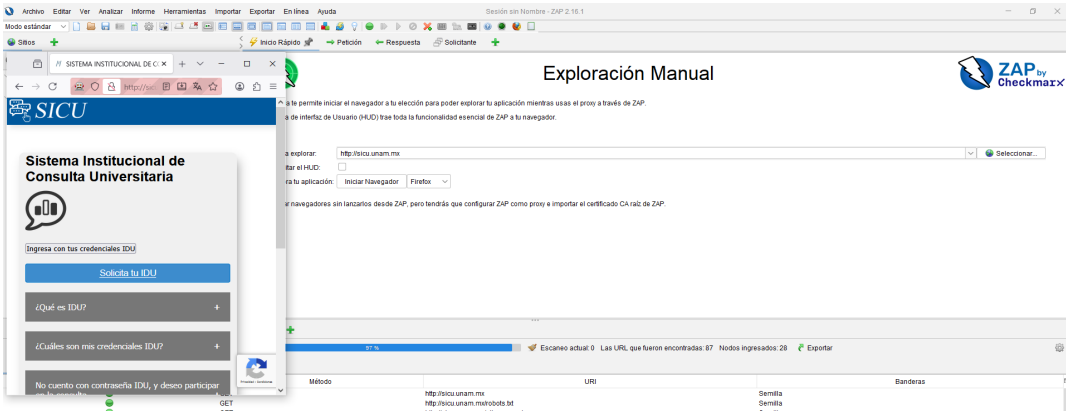


2. Escaneo manual, en el cual se registraron las solicitudes y respuestas sin modificar el tráfico, con el fin de encontrar vulnerabilidades que los escaneos automatizados pasan por alto, especialmente aquellas relacionadas con la lógica de negocio y el contexto específico de la aplicación (véase la Figura 5).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 124 - 130

Figura 5

Escaneo manual OWASP ZAP



V. Validación, análisis de impacto y priorización contextual de hallazgos

Para la priorización de los hallazgos se aplicó la OWASP Risk Rating Methodology (OWASP Foundation, 2023b), complementando la clasificación automática generada por OWASP ZAP mediante un proceso estructurado de dos etapas.

Etapa 1: Clasificación inicial automatizada

OWASP ZAP proporcionó una valoración técnica preliminar basada en:

- Patrones predefinidos de vulnerabilidad.
- Severidad estándar de la industria.
- Heurísticas automatizadas de detección.
- Resultado: Clasificación genérica, (véase Figura 6).

Figura 6

Reporte de alertas

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 125 - 130

Etapa 2: Evaluación contextual manual (Matriz OWASP)

Cada hallazgo fue reevaluado considerando el contexto institucional del sistema, aplicando los factores de valoración siguientes (véase Figura 7), así como la matriz de severidad de riesgo OWASP Risk; (véase Figura 8).

Figura 7

Reporte de alertas OWASP ZAP

Figura 8

Matriz de severidad, OWASP Risk Rating Methodology

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

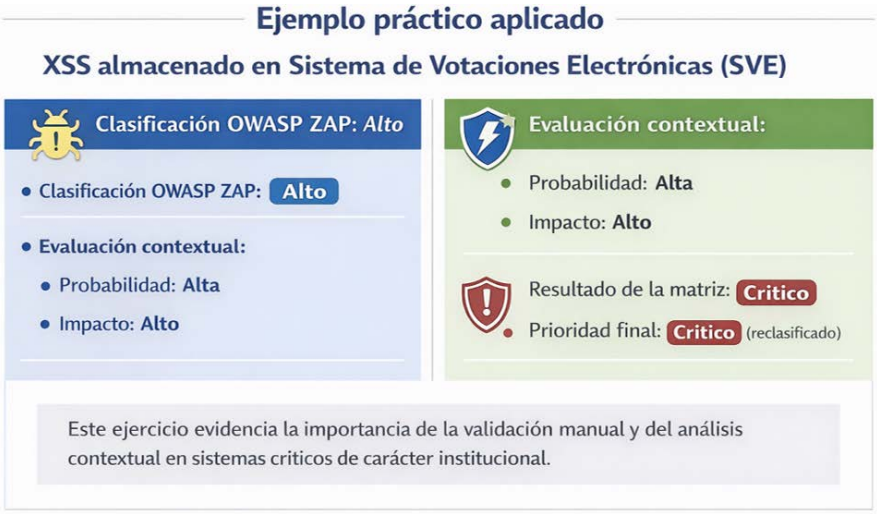
Nota. Riesgo = Probabilidad x Impacto = Likelihood x Impact. Recuperado de https://secumantra.com/owasp-top-ten-risk-rating/

A continuación, se muestra un ejemplo práctico de validación, análisis de impacto y priorización, (véase Figura 9).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 126 - 130

Figura 9

Evaluación de hallazgo cross-site scripting (XSS)



VI. Mitigación y revalidación

Los hallazgos confirmados fueron reportados mediante evidencias, nivel de riesgo y recomendaciones de mitigación. Posteriormente, se contempló la ejecución de re-escaneos para verificar la efectividad de las acciones correctivas aplicadas.

3. RESULTADOS

La integración de Pruebas Dinámicas de Seguridad de Aplicaciones, durante la fase de pruebas del SDLC, permitió identificar, corregir y mitigar vulnerabilidades en aproximadamente el 80% de los hallazgos identificados —principalmente aquellos clasificados con severidad media, alta y crítica— en la primera iteración de corrección. Esta práctica redujo significativamente los costos de corrección en comparación con la detección tardía durante auditorías posteriores al despliegue, ya que se realizaron en la fase de pruebas del SDLC con sistemas funcionales, pero aún en proceso de ajuste y mejora.

Los hallazgos obtenidos, además de proporcionar información relevante, retroalimentaron de manera inmediata las actividades de diseño y codificación en iteraciones subsecuentes, contribuyendo a la prevención de repeticiones y a la mejora progresiva de los controles de seguridad implementados. Asimismo, la adopción de estas pruebas dinámicas, alineadas con los estándares OWASP, fortaleció un ciclo robusto de mejora continua en la seguridad de aplicaciones web.

Como parte de una estrategia de mejora continua, los resultados de las pruebas dinámicas con OWASP ZAP en los sistemas de Consultas Universitarias y de Voto Electrónico revelaron hallazgos relacionados con aspectos de seguridad que no fueron considerados durante el diseño y desarrollo inicial, tales como configuraciones inseguras por defecto, cabeceras de seguridad faltantes (Política de Seguridad de Contenidos y Seguridad Estricta de Transporte HTTP), y controles insuficientes en mecanismos de autenticación, mismos que fueron subsanados en cada iteración (véase la Figura 10 y 11).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 127 - 130

Figura 10

Figura extraída de la auditoría externa realizada al Sistema de Votaciones Electrónicas 2024

NIVEL DE SEGURIDAD EN EL SISTEMA DE VOTACIONES.

ENTENDIMIENTO DE RESULTADOS

LOS RESULTADOS DETERMINAN UN NIVEL **ALTO** DE SEGURIDAD EN EL SISTEMA DE VOTACIONES ELECTRÓNICAS EN SU VERSIÓN 5, CON BASE EN EL HECHO DE QUE FUERON **DETECTADAS 0 (CERO)** "VULNERABILIDADES" DE ACUERDO CON EL MARCO OWASP.

LO ANTERIOR NO EXIME DE MANTENER CONTROLES DE SEGURIDAD Y EVALUAR CONTINUAMENTE EL SISTEMA, DADO QUE, PARA UN INTERNO CON CONOCIMIENTO, RESULTARÍA MÁS SENCILLO CAUSAR ALGÚN DAÑO A LOS ACTIVOS DE INFORMACIÓN Y A LA INFORMACIÓN MISMA DE LA UNIVERSIDAD Y SUS VOTANTES

Figura 11

Figura extraída de la auditoría interna realizada al Sistema Institucional de Consultas Universitarias 2024

DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN
COORDINACIÓN DE SEGURIDAD DE LA INFORMACIÓN

INFORME DE PRUEBAS DE SEGURIDAD A SITIO WEB SICU.UNAM.MX

DGTIC

RESUMEN EJECUTIVO

En el presente informe se documentan los resultados de las pruebas y, con base en ellos, se emiten recomendaciones que incluyen acciones correctivas y preventivas orientadas a fortalecer el esquema de seguridad actual.

Como resultado de las pruebas se identificaron 6 hallazgos, 1 medio, 2 bajos y 3 sin impacto, según la calificación establecida a través del *Common Vulnerability Scoring System* Versión 3.1 (Anexo C).

La siguiente tabla muestra la cantidad de hallazgos identificados por activo.

HALLAZGOS IDENTIFICADOS DURANTE LA REVISIÓN						
ID ACTIVO	SITIO	CRITICOS	ALTO	MEDIO	BAJO	SIN IMPACTO
ACT01	sicu.unam.mx	0	0	1	2	3

Como resultado, las auditorías de seguridad internas y externas, realizadas posteriormente a los sistemas, reflejaron una madurez institucional, evidenciada por una superficie de ataque reducida y una gestión eficaz de los hallazgos residuales de acuerdo con los informes presentados por las auditorías respectivas.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 128 - 130

4. CONCLUSIONES

Las Pruebas Dinámicas de Seguridad de Aplicaciones (DAST), aplicadas a los sistemas institucionales, permitieron identificar vulnerabilidades críticas que no siempre pueden ser detectadas mediante otros tipos de análisis, tales como inyecciones SQL, *cross-site scripting* (XSS) y debilidades en los mecanismos de autenticación, lo cual se reflejó en la corrección de los hallazgos identificados. Al evaluar la aplicación en ejecución y bajo condiciones reales, las DAST proporcionan una medición más precisa de su nivel de seguridad, con una tasa significativamente menor de falsos positivos en comparación con técnicas que no analizan aplicaciones en funcionamiento (Putra et al., 2024; Qadir et al., 2025).

No obstante, desde una perspectiva metodológica, las DAST presentan ciertas limitaciones, ya que, al enfocarse exclusivamente en componentes accesibles desde el exterior, no permiten examinar en profundidad elementos internos como la lógica de negocio o procesos no expuestos. Asimismo, al requerir que el sistema se encuentre operativo y generalmente en entornos de producción, su aplicación en fases tardías del ciclo de desarrollo puede incrementar el esfuerzo y los costos de corrección (Abdulghaffar et al., 2023).

En el presente contexto, estas limitaciones fueron reducidas al integrar pruebas dinámicas antes del despliegue en entornos productivos, permitiendo que los hallazgos se atendieran durante etapas aún controladas en la fase de pruebas, aunque se reconoce que una incorporación de pruebas unitarias por parte de los desarrolladores, complementada con Pruebas Estáticas de Seguridad de Aplicaciones (SAST), podría optimizar aún más los resultados.

Por estas razones, se recomienda utilizar DAST en conjunto con pruebas SAST, ya que esta combinación permite analizar el código fuente desde la fase de desarrollo y ofrece una visión más integral del estado de seguridad de una aplicación (Koman & Janiszewski, 2025).

Para que las pruebas DAST sean realmente eficaces, es indispensable configurar la herramienta de acuerdo con las características específicas del sistema a evaluar, prestando atención a la autenticación, gestión de sesiones y exposición de información sensible del aplicativo o sistema. En este sentido, se aconseja implementar validaciones robustas, mecanismos de autenticación más estrictos y políticas que limiten la divulgación de información operativa del software.

Asimismo, la incorporación de DAST como parte del proceso de desarrollo favorece una cultura de seguridad continua en las organizaciones, al complementar otras técnicas de análisis y contribuir a una mejor preparación de los sistemas ante amenazas de seguridad en sus entornos productivos.

REFERENCIAS

Abdulghaffar, K., Elmrabit, N., & Yousefi, M. (2023). Enhancing web application security through automated penetration testing with multiple vulnerability scanners. *Computers,* 12(11), 235. https://www.mdpi.com/2073-431X/12/11/235

Aydos, M., & Baykara, M. (2022). Security testing of web applications: A systematic mapping. *Journal of Information Security and Applications*, 63, 103005. https://www.sciencedirect.com/science/article/pii/S131915782100269X?via%3Dihub

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 129 - 130

Chorell, I., & Ekberg, C. (2024). *A comparative analysis of open source dynamic application security testing tools* [Tesis de maestría, Linköping University, Departamento de Ciencias de la Computación e Información]. Linköping University.

https://www.diva-portal.org/smash/get/diva2:1868722/FULLTEXT01.pdf

Dencheva, L. (2022). *Comparative analysis of static application security testing (SAST) and dynamic application security testing (DAST).*

https://norma.ncirl.ie/5956/1/lyubkadencheva.pdf

International Organization for Standardization. (2018). *Information technology — Security techniques — Application security — Part 3: Application security management process* (ISO/IEC Standard No. 27034-3:2018*)*. https://www.iso.org/standard/55583.html

International Organization for Standardization. (2022). *ISO/IEC 27001:2022 Information technology—Security techniques—Information security management systems—Requirements*. ISO. – Requirements. ISO. https://www.iso.org/standard/27001

Jennings, T. (2025, 5 de mayo). *Dynamic Application Security Testing: DAST Basics*. Mend.io. https://www.mend.io/blog/dast-dynamic-application-security-testing/

Koman, J., & Janiszewski, M. (2025). SCAnME – scanner comparative analysis and metrics for evaluation. *International Journal of Information Security*, 24, Article 147. https://doi.org/10.1007/s10207-025-01054-8

Kondraciuk, A., Bartos, A., & Pańczyk, B. (2022). Comparative analysis of the effectiveness of OWASP ZAP, Burp Suite, Nikto and Skipfish in testing the security of web applications. *Journal of Computer Sciences Institute*, 24, 176-180. https://doi.org/10.35784/jcsi.2929

National Institute of Standards and Technology. (2020). *Security and privacy controls for information systems and organizations* (NIST Special Publication 800-53, Revision 5). https://doi.org/10.6028/NIST.SP.800-53r5

National Institute of Standards and Technology. (2022). *Secure software development framework (SSDF) version 1.1: Recommendations for mitigating the risk of software vulnerabilities* (NIST Special Publication 800-218). U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-218

OWASP Foundation. (2021). *OWASP Top Ten:2021*. https://owasp.org/Top10/

OWASP Foundation. (2023). *Application Security Verification Standard (ASVS)* v4.0.3. https://owasp.org/ASVS/

OWASP Foundation. (2023a). *OWASP Zed Attack Proxy (ZAP)*. https://www.zaproxy.org/

OWASP Foundation. (2023b). *OWASP Risk Rating Methodology*. OWASP Foundation. https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

Putra, F. P. E., Ubaidi, U., Hamzah, A., Pramadi, W. A., & Nuraini, A. (2024). Systematic Literature Review: Security Gap Detection On Websites Using OWASP ZAP. *Brilliance: Research of Artificial Intelligence,* 4(1), 348-355. https://doi.org/10.47709/brilliance.v4i1.4227

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 130 - 130

Qadir, S., Waheed, E., Khanum, A., & Jehan, S. (2025). Comparative evaluation of approaches & tools for effective security testing of Web applications. *PeerJ Computer Science*, 11, e2821. https://doi.org/10.7717/peerj-cs.2821