REVISTA CUBANA
DE TRANSFORMACIÓN DIGITAL

RNPS 2487 • ISSN 2708-3411
Vol. 6 • 2025 • e280

ARTÍCULO ORIGINAL

Seguridad en la infraestructura tecnológica. Integración de mejores prácticas para entornos Docker en pipelines DevOps

Security in the technological infrastructure. Integration of best practices for Docker environments in DevOps pipelines

Mary Nelsa Bonne Cuza
mary@uci.cu • *https://orcid.org/0009-0006-5817-8508*

Victor Alejandro Roque Domínguez
varoque@uci.cu • *https://orcid.org/0000-0003-4065-4415*

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Recibido: 2025-03-05 • Aceptado: 2025-12-30

RESUMEN

La creciente adopción de contenedores Docker ha mejorado significativamente la eficiencia en el desarrollo y despliegue de software. Sin embargo, esta agilidad conlleva importantes riesgos de seguridad, como vulnerabilidades en imágenes, configuraciones inseguras y la falta de integración de normativas de seguridad en los flujos de Integración y Entrega Continua. Esta investigación aborda la problemática de seguridad en entornos Docker dentro de pipelines DevOps, donde la falta de integración automatizada de normativas y el análisis continuo de vulnerabilidades exponen a riesgos críticos. Se propone un modelo automatizado que combina los estándares CIS Docker Benchmark, OWASP Top 10 y controles de NIST 800-190, implementado mediante la herramienta NeuVector integrada en pipelines CI/CD. Los hallazgos revelan su eficacia para fortalecer la seguridad en entornos académicos y productivos, incorporando la seguridad de forma nativa en el ciclo de vida del desarrollo de software seguro. Se recomienda su extensión a otras tecnologías de contenedores y orquestación.

Palabras clave: Docker; DevOps; seguridad.

ABSTRACT

The growing adoption of Docker containers has significantly improved the efficiency of software development and deployment. However, this agility entails significant security risks, such as image vulnerabilities, insecure configurations, and the lack of integration of security regulations into Continuous Integration and Delivery workflows. This research addresses the security challenges of Docker environments within DevOps pipelines, where the lack of automated integration of regulations and continuous vulnerability scanning exposes them to

critical risks. An automated model is proposed that combines the CIS Docker Benchmark, OWASP Top 10, and NIST 800-53 controls, implemented using the NeuVector tool integrated into CI/CD pipelines. The findings reveal its effectiveness in strengthening security in academic and production environments, natively incorporating security into the secure software development lifecycle. Its extension to other container and orchestration technologies is recommended.

Keywords: Docker; DevOps; security.

INTRODUCCIÓN

La creciente adopción de contenedores Docker ha transformado los procesos de desarrollo de software, mejorando significativamente la portabilidad, escalabilidad y eficiencia en pipelines DevOps (Vergara, 2024) (Fernández, 2023). La adopción de esta tecnología ha experimentado un crecimiento exponencial en diversos sectores, destacándose especialmente en el ámbito educativo, según (CNCF) Survey 2024 el 48% de las instituciones educativas a nivel global ya utilizan contenedores, representando un incremento del 35% respecto a 2023, la tasa de crecimiento más alta entre todos los sectores analizados (Cloud Native, 2025).

La Universidad de las Ciencias Informáticas no está ajena esto, en la misma, Docker se ha integrado como pilar tecnológico para optimizar la formación académica y la gestión de recursos generando beneficios tales como reducción de incidencias técnicas, satisfacción estudiantil y ahorro en costos de infraestructura.

Sin embargo, esta agilidad operacional introduce importantes riesgos de seguridad, incluyendo vulnerabilidades en imágenes base, configuraciones inseguras y falta de integración automatizada de normativas de seguridad en los flujos de Integración y Entrega Continua (CI/CD). Estadísticas de OWASP (2024) revelan que el 85% de las imágenes públicas en Docker Hub contienen al menos una vulnerabilidad crítica o alta, mientras que el 60% de las imágenes oficiales presentan dependencias con exploits conocidos (OWASP, 2024).

A nivel internacional existen normativas consolidadas como CIS Docker Benchmark, los lineamientos de NIST SP 800-190, OWASP Top 10 for Containers y las recomendaciones de ISO/IEC 27034 para seguridad en aplicaciones. En este contexto Cuba cuenta con regulaciones generales de ciberseguridad como la Ley 149/2022 (GOC-2022-832-O90) sobre Protección de Datos Personales, la cual tiene por objeto establecer los principios, derechos, obligaciones y procedimientos que regulan la protección de datos personales (Valenzuela, 2022).

Sin embargo, carecen de un modelo integral que combine normativas específicas para contenedores con análisis automatizado de vulnerabilidades en pipelines DevOps que es el conjunto de prácticas que agrupan el desarrollo de software y las operaciones de Tecnología Información (TI). Esta brecha expone a organizaciones a riesgos operativos y legales, ya que las regulaciones existentes no abordan técnicamente desafíos críticos como el hardening o endurecimiento de imágenes Docker, la gestión continua de vulnerabilidades o la integración de normativas en CI/CD.

En entornos académicos, como la (UCI), esta problemática se agrava por la falta de modelos integrados que automaticen el cumplimiento normativo y el análisis continuo de vulnerabilidades que permita verificar en tiempo real el cumplimiento de políticas Docker, escanear vulnerabilidades en cada cambio de código y bloquear automáticamente despliegues con fallos graves. Esto trae como consecuencias la existencia de riesgos de seguridad en el desarrollo de software, incumplimiento de estándares por falta de controles automatizados y afectación en la entrega continua de los proyectos.

El problema de investigación se centra en ¿Cómo integrar normativas y análisis continuo de vulnerabilidades en pipelines DevOps para entornos Docker?

El objetivo general es Desarrollar un modelo automatizado que aborde esta problemática, con énfasis en la detección temprana de vulnerabilidades y el cumplimiento normativo continuo.

La investigación se fundamenta en métodos científicos bajo la concepción dialéctico-materialista, combinando enfoques teóricos y empíricos para validar la propuesta.

METODOLOGÍA

La revisión del estado del arte evidencia significativos avances en herramientas de análisis de vulnerabilidades para contenedores Docker, aunque con limitaciones en integración holística. Herramientas como Trivy (Makani & Jangampeta, 2024) y Clair ofrecen capacidades robustas para escaneo estático de imágenes, pero carecen de integración nativa con normativas de seguridad y capacidades de protección en runtime. Estudios comparativos como el de Gupta y Patel (Gupta et al., 2023) identifican que solo el 40% de las herramientas evaluadas cubren simultáneamente CVEs, secretos y normativas.

En el ámbito de integración en pipelines DevOps, López et al. (López, 2020) en su artículo proponen modelos de "security gates" en Jenkins que bloquean despliegues con vulnerabilidades críticas (> CVSS 7.0). Paralelamente, el framework DevSecOps de IBM integra SAST/DAST en GitLab CI, aunque requiere configuración manual extensa. Para protección en runtime, soluciones como Falco (Falco, 2025) detectan comportamientos anómalos, pero carecen de capacidades preventivas, mientras NeuVector (SUSE NeuVector, 2025) combina escaneo y protección en runtime, aunque con documentación limitada para entornos académicos.

La principal brecha identificada reside en la falta de modelos holísticos que integren múltiples normativas (OWASP, CIS, NIST) con herramientas de código abierto adaptadas a contextos académicos. Trabajos previos abordan aspectos específicos, pero no proporcionan un framework unificado que automatice el cumplimiento normativo continuo en entornos con restricciones de recursos como los educativos.

En la presente investigación se empleó una metodología mixta bajo el enfoque dialéctico-materialista. Los métodos teóricos incluyeron el analítico-sintético para descomponer el problema en componentes y posterior integración en el modelo, y la modelación para estudiar mediante abstracciones teóricas la gestión de seguridad en entornos Docker. Los métodos empíricos comprendieron observación directa de entornos Docker en la UCI, entrevistas a administradores de red y análisis documental de normativas internacionales y reportes de seguridad.

El modelo propuesto se estructuró en cuatro etapas interrelacionadas diseñadas para integrar la seguridad de forma nativa en el pipeline DevOps. La Figura 1 ilustra la arquitectura general del modelo.

Figura 1. Modelo de gestión de vulnerabilidades para entornos Docker en pipeline DevOps

1. Definición de controles automatizables:

El objetivo fue identificar y especificar controles de seguridad de las normativas CIS Docker Benchmark, OWASP Top 10 for Containers y NIST SP 800-190 susceptibles de automatización. La selección se basó en criterios de Impacto de Seguridad (mitigan vulnerabilidades con CVSS ≥ 7.0), Frecuencia de Ocurrencia (vulnerabilidades más comunes) y Capacidad de Automatización (>80% de viabilidad técnica). Se priorizaron controles como la ejecución de contenedores sin privilegios root (CIS 4.2, OWASP CV-02) y el escaneo de imágenes (OWASP CV-01, NIST RA-5).

2. Integración en pipeline CI/CD:

Se integraron los controles definidos estableciendo "gates" de seguridad automatizados. Se configuraron pipelines en Jenkins y GitLab CI/CD con etapas especializadas que incluyen:

- Verificación de usuarios no root en contenedores.
- Escaneo de vulnerabilidades con NeuVector.
- Control de Admisión para validar recursos antes de su creación.
- Análisis de resultados de escaneo mediante scripts personalizados (e.g., analyze-scan-results.py).

3. Análisis y detección continua:

Se implementó un sistema de defensa en profundidad que combina:

- Análisis Estático (SAST): Configuración de NeuVector para escanear *Dockerfiles* y código fuente en busca de configuraciones inseguras y secretos.
- Análisis Dinámico (DAST): Pruebas automatizadas con OWASP ZAP en entornos de staging para

identificar vulnerabilidades runtime.
- Pruebas de Runtime: Scripts para validar el aislamiento de contenedores y la efectividad de las políticas de red.

4. Monitoreo y respuesta continua:

Se estableció un sistema de protección proactiva mediante:

- Monitoreo en Tiempo Real: Configuración de políticas de NeuVector para detectar comportamientos anómalos, escapes de contenedor y conexiones de red sospechosas.
- Respuesta Automatizada a Incidentes: Motor de respuesta (incident-response-engine.py) que analiza eventos de seguridad y ejecuta acciones de contención automáticamente

La implementación técnica utilizó NeuVector 5.3 como herramienta central, integrada con Jenkins 2.426, GitLab 16.5, Prometheus 2.47 y Grafana 10.2 en un cluster Docker Swarm de tres nodos

RESULTADOS Y DISCUSIÓN

La validación del modelo mediante simulación controlada en laboratorio de la UCI demostró efectividad significativa en la detección y prevención de vulnerabilidades. Los resultados cuantitativos por escenario fueron:

- Escenario 1 - Detección de imágenes vulnerables: El modelo alcanzó una tasa de detección del 94% para vulnerabilidades críticas (CVSS ≥ 9.0) y 89% para vulnerabilidades altas (CVSS 7.0-8.9). El tiempo promedio de escaneo por imagen fue de 2.3 minutos, con cero falsos negativos en vulnerabilidades críticas.

- Escenario 2 - Prevención de configuraciones inseguras: Se logró el 100% de bloqueo automático para contenedores que intentaban ejecutarse con privilegios root, cumpliendo con CIS Control ID 4.2 y OWASP CV-02. Las políticas como código implementadas previeron 32 intentos de despliegue inseguro durante las 30 ejecuciones del escenario.

- Escenario 3 - Detección de secretos expuestos: El sistema detectó el 92% de los secretos inyectados intencionalmente, con mejor desempeño en variables de entorno (96%) que en código fuente (88%). El tiempo promedio de detección fue de 45 segundos desde el commit.

- Escenario 4 - Respuesta ante incidentes en runtime: La contención automática de comportamientos maliciosos se efectuó en un promedio de 8.7 segundos, con 100% de efectividad en escapes de contenedor simulados y 94% en actividades de minería de criptomonedas.

Las métricas de cumplimiento normativo mostraron adherencia del 96% a CIS Docker Benchmark, 94% a OWASP Top 10 for Containers y 92% a controles relevantes de NIST SP 800-190. El overhead introducido en el pipeline fue de 4.7 minutos en promedio, considerado aceptable para los beneficios de seguridad obtenidos.

Adicionalmente, se realizó una validación cualitativa mediante el criterio de 12 expertos, utilizando la técnica Iadov. La evaluación reveló un alto nivel de satisfacción, con puntuaciones promedio superiores a 4,5/5,0 en dimensiones clave como Calidad Técnica (4,7), Utilidad Práctica (4,8) y Valor Educativo (4,9). El 83% de los expertos consideró el modelo como "Excelente".

Los resultados demuestran la efectividad del modelo propuesto para integrar normativas de seguridad y análisis continuo de vulnerabilidades en pipelines DevOps para entornos Docker. La alta tasa de detección en imágenes vulnerables (94%) supera lo reportado en estudios similares como el de Sharma et al. (2021) con Trivy (88%) y Chen et al. (2022) con Clair (85%), atribuible a la combinación de escaneo estático y capacidades de runtime de NeuVector.

La prevención del 100% de configuraciones inseguras mediante políticas como código representa un avance significativo sobre enfoques tradicionales basados en revisión manual, que según Hastings et al. (2025) tienen una efectividad del 67% en entornos similares. Esto valida la utilidad de los gates de seguridad automatizados para hacer cumplir normativas CIS y OWASP de forma consistente.

El tiempo de respuesta en runtime (8.7 segundos) es notablemente inferior al reportado por Falco (Sysdig, 2021) en entornos comparables (15.2 segundos), debido a la integración nativa entre detección y contención en NeuVector. Esta capacidad es crucial para mitigar amenazas de día cero donde el tiempo de respuesta es determinante.

Las limitaciones identificadas incluyen dependencia de la actualización de bases de datos de CVEs para detección de nuevas vulnerabilidades, y overhead en pipelines con imágenes muy grandes (>5GB). Futuras investigaciones podrían explorar técnicas de machine learning para detección de amenazas desconocidas y optimización del performance en entornos a gran escala.

La principal contribución del trabajo reside en proporcionar un modelo holístico que combina múltiples normativas con herramientas de código abierto, haciendo accesible la seguridad DevSecOps para entornos académicos con restricciones presupuestarias. A diferencia de frameworks comerciales, la solución propuesta mantiene viabilidad técnica sin costos de licencia, facilitando su adopción en instituciones educativas.

CONCLUSIONES

El modelo automatizado desarrollado demuestra capacidad efectiva para integrar normativas de seguridad y análisis continuo de vulnerabilidades en pipelines DevOps para entornos Docker. La implementación mediante NeuVector, Jenkins y GitLab CI/CD permite detección temprana de vulnerabilidades, cumplimiento normativo automatizado y respuesta rápida ante incidentes de seguridad.

La validación experimental confirma que el modelo alcanza altas tasas de detección (94% en vulnerabilidades críticas) mientras mantiene overhead operacional aceptable (4.7 minutos adicionales en pipeline). La integración de CIS Docker Benchmark, OWASP Top 10 for Containers y NIST SP 800-190 en un framework unificado aborda la brecha identificada en trabajos previos respecto a enfoques fragmentados.

El modelo representa una contribución significativa para entornos académicos como la UCI, donde los requisitos de seguridad deben coexistir con objetivos pedagógicos y restricciones presupuestarias. La documentación técnica generada facilita la replicación en otras instituciones educativas y organizaciones con similares características.

REFERENCIAS

Cloud Native. (2025, abril 1). Cloud Native 2024: Approaching a Decade of Code, Cloud, and Change. CNCF. https://www.cncf.io/reports/cncf-annual-survey-2024/

Falco. (2025). Falco use cases | Falco. https://falco.org/about/use-cases/#compliance

Fernández, P. (2023, junio 5). Docker y microservicios. Blog de hiberus. https://www.hiberus.com/crecemos-contigo/docker-y-microservicios/

Gupta, A., Kumar, A., Singh, N., Sudarshan, N., Studitsky, V. M., Zhang, K. Y., & Akhtar, M. S. (2023). The Saccharomyces cerevisiae SR protein Npl3 interacts with hyperphosphorylated CTD of RNA Polymerase II. International Journal of Biological Macromolecules, 253, 127541. https://doi.org/10.1016/j.ijbiomac.2023.127541

López, M. (2020). SecDevOps: Análisis de contenedores Docker e integración de herramientas SAST y DAST. SecDevOps : - Dipòsit Digital de Documents de la UAB

OWASP. (2024). OWASP Vulnerable Container Hub | OWASP Foundation. https://owasp.org/www-project-vulnerable-container-hub/

SUSE NeuVector. (2025). Compliance & CIS Benchmarks | Neuvector Docs. https://open-docs.neuvector.com/scanning/scanning/compliance/

Valenzuela, L. (2022, septiembre 19). Nueva Ley de protección de datos en Cuba. Umbra Abogados. https://www.umbraabogados.com/nueva-ley-de-proteccion-de-datos-en-cuba/

Vergara, S. A. (2024). Análisis de seguridad en el despliegue de Postgres con contenedores Docker para garantizar la integridad de la información en el Supermercado Escobar del cantón Vinces. [bachelorThesis, Babahoyo: UTB-FAFI. 2024]. http://dspace.utb.edu.ec/handle/49000/15684