IMPLEMENTACIÓN DE LABORATORIOS DE CIBERSEGURIDAD PARA LA SIMULACIÓN DE ATAQUES Y PRUEBAS DE INTRUSIÓN EN ENTORNOS AISLADOS

IMPLEMENTATION OF CYBERSECURITY LABORATORIES FOR ATTACK SIMULATION AND INTRUSION TESTING IN ISOLATED ENVIRONMENTS

Nicolás Alonzo Gutiérrez
Tecnológico Nacional de México / IT de Apizaco, México
nicolas.ag@apizaco.tecnm.mx

Lucía Muñoz Dávila
Tecnológico Nacional de México / IT de Apizaco, México
lucia.md@apizaco.tecnm.mx

Recepción: 28/noviembre/2025 **Aceptación:** 24/diciembre/2025

Resumen

En este trabajo se aborda el desafío de formar profesionales en ciberseguridad mediante la implementación de laboratorios accesibles y económicos para entornos académicos. Se presenta un enfoque para reducir costos sin comprometer la calidad educativa: contenedores Docker (Kali Linux, OWASP Juice Shop, Metasploitable). Estos laboratorios simulan ataques controlados en entornos aislados y seguros, permitiendo a los estudiantes practicar técnicas de reconocimiento, explotación y post-explotación sin riesgos para infraestructuras reales. Los resultados demuestran la efectividad de estas soluciones, destacando la escalabilidad de los contenedores, lo que optimiza significativamente el proceso de enseñanza-aprendizaje en instituciones con recursos limitados.

Palabras Clave: Ciberseguridad, contenedores, hacking ético, pruebas de vulnerabilidades.

Abstract

This paper addresses the challenge of training cybersecurity professionals by implementing accessible and affordable labs for academic environments. It presents

an approach to reducing costs without compromising educational quality: Docker containers (Kali Linux, OWASP Juice Shop, Metasploitable). These labs simulate controlled attacks in isolated and secure environments, allowing students to practice reconnaissance, exploitation, and post-exploitation techniques without risking real infrastructure. The results demonstrate the effectiveness of these solutions, highlighting the scalability of containers, which significantly optimizes the teaching-learning process in institutions with limited resources.

Keywords: *Cybersecurity, containers, ethical hacking, vulnerability testing.*

1. Introducción

La creciente complejidad de las ciber amenazas demanda profesionales con habilidades prácticas sólidas, capaces de enfrentar escenarios reales. En este contexto, los laboratorios de ciberseguridad se vuelven fundamentales en entornos académicos, ya que permiten a los estudiantes experimentar, equivocarse y aprender en un ambiente controlado, sin riesgo para infraestructuras reales en [Chaparro, 2024], [Santillan, 2024] se hace un análisis de infraestructura de ciberseguridad en ambientes académicos y [Švábenský, 2021] desarrolla habilidades de ciberseguridad con desafíos de captura de bandera. Sin embargo, la implementación de laboratorios tradicionales conlleva costos elevados en hardware, software y mantenimiento, limitando su accesibilidad a los estudiantes como en [García, 2024] que ofrece un entorno completo basado en nube privada. En [Martínez, 2024] se desarrolla un laboratorio basado en raspberry de bajas prestaciones, y se busca en una futura línea de investigación el uso de sistemas operativos y aplicaciones minimalistas para optimizar el rendimiento del sistema. [Fernández, 2019] propone también contenedores para pentesting. En [Hassan, 2022] se establece una infraestructura centralizada que prioriza la accesibilidad universal mediante navegador web y la seguridad mediante contenedores efímeros con limitación temporal y escalable, [Irvine, 2017] con Labtainers, representa un equilibrio entre portabilidad y comprensividad con un sistema automatizado de evaluación y laboratorios individualizados, pero impone requisitos técnicos como hipervisores que pueden limitar su accesibilidad inmediata. [Ahmed, 2023] ofrece

un entorno para ejercicios específicos basados en escenarios de vulnerabilidades concretas y privilegia la relación teoría y práctica, pero carece de escalabilidad. Cada solución constituye una respuesta válida a diferentes restricciones institucionales, demostrando que la efectividad pedagógica en ciberseguridad depende de la adecuación entre objetivos educativos, recursos tecnológicos y población estudiantil.

Con respecto a las plataformas comerciales en donde [Fahrnberger, 2025] propuso la creación de un ambiente para estudiar ramsomware. [Pratama, 2024] realiza un Asistente Inteligente de Pruebas de Penetración en Ciberseguridad para Investigadores Éticos. [Tashkov, 2025] realiza un análisis de distintas plataformas, tanto comerciales como universitarias incluyendo TryHackMe, Hack The Box, PicoCTF, CTFtime y cyber rango. [Karagiannis, 2020] hace una evaluación de plataformas como cybersecurity e-learning tools.

En este trabajo se analizan las posibilidades que ofrecen las tecnologías de virtualización, como contenedores Docker, y reducir drásticamente estos costos sin comprometer la calidad educativa. El objetivo es demostrar cómo estas herramientas facilitan la creación de entornos de prueba aislados, escalables y económicos, que replican vulnerabilidades y ataques complejos, optimizando así el proceso de enseñanza-aprendizaje en instituciones educativas con recursos limitados.

2. Métodos

Prerrequisitos

La implementación del laboratorio requiere una preparación técnica específica del sistema anfitrión. Para entornos con sistema operativo Windows, es esencial habilitar $WSL\ 2$ mediante PowerShell con permisos de administrador, seguido de la instalación del kernel de $WSL\ 2$ y la configuración de esta versión como predeterminada. En sistemas Linux, se recomienda una distribución basada en Debian. Posteriormente, se instala Docker Engine y Docker Compose, verificando su funcionamiento con contenedores de prueba. La integración entre $WSL\ 2$ y Docker Desktop en Windows es crucial para garantizar la compatibilidad. Se

asignan recursos mínimos de $4\ GB$ de RAM y $20\ GB$ de almacenamiento para operar múltiples contenedores simultáneamente.

Instalación de una distribución Linux en WSL

Se Accede a Microsoft Store y buscar la distribución deseada, como Debian. Proceder con la instalación y, una vez completada, iniciar la distribución desde el menú de inicio. O si ya conoce la distribución puede escribirla directamente. En la Tabla 1 se especifican los requisitos para WSL.

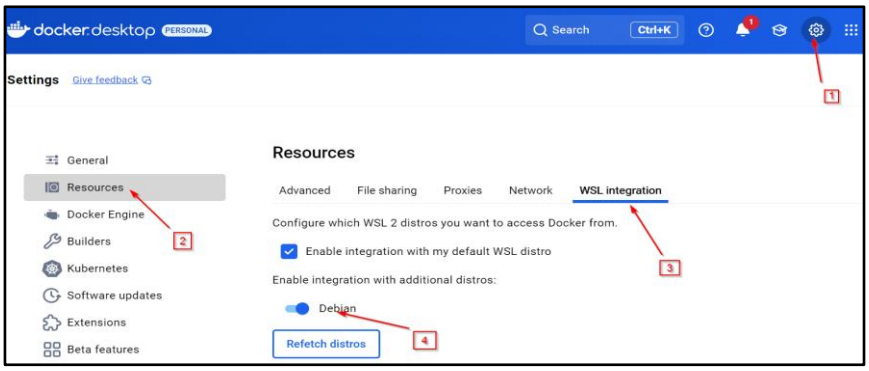
Tabla 1 Requisitos para WSL.

Paso	Acción	Comando/Verificación
1	Habilitar $WSL\ 2$ (Windows 10/11 Pro o Superior)	Ejecutar las instrucciones en PowerShell (Admin): • `dism.exe /online /enable-feature/featurename:Microsoft-Windows-Subsystem-Linux /all /norestart` • `dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart`
2	Reiniciar el sistema	-
3	Descargar e instalar el kernel de $WSL\ 2$	Descargar desde Microsoft y ejecutar el instalador. *https://aka.ms/wsl2kernel*
4	Establecer $WSL\ 2$ como versión predeterminada	Ejecutar la instrucción en PowerShell (Admin): `wsl --set-default-version 2`

Fuente: elaboración propia

Configuración inicial de Debian

Durante el primer inicio, se solicitará la creación de un usuario y contraseña. Estos datos son independientes de las credenciales de Windows y se utilizarán para gestionar la instancia de WSL. Obtener la versión estable de Docker Desktop desde el sitio oficial y ejecutar el instalador con las opciones predeterminadas, asegurándose de marcar la instalación de componentes adicionales requeridos para Docker [Martínez, 2022]. Es recomendable el uso de Docker desktop para facilitar la administración de los contenedores. En la referencia de Docker hay que descargarlo e instalarlo también en Debian. Es importante integrarlo con Debian. En la Figura 1, se describen los pasos para hacerlo. Por último, verificar la instalación con los comandos: *"docker –versión"* y *"Docker compose –versión".* Probar la ejecución de un contenedor con el siguiente comando: *"docker run hello-world".*

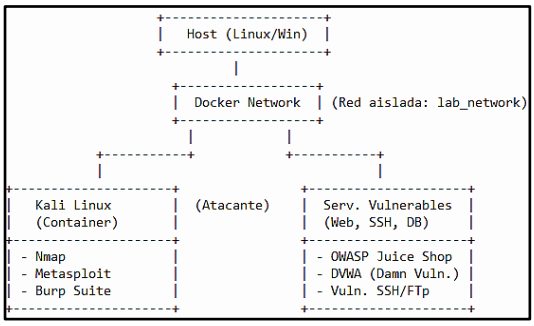


Fuente: elaboración propia

Figura 1 Integración de WSL2 con Docker Desktop.

Arquitectura del laboratorio de pruebas

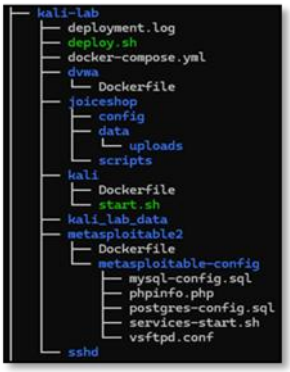
La arquitectura general de los laboratorios se presenta en la Figura 2, donde se muestra el contenedor de kali Linux que tiene las herramientas que se utilizan para realizar los ataques y están unidos por la red de Docker Network a los contenedores que tienen configurados los servicios vulnerables, todo ello en un solo host basado en Linux o Windows.



Fuente: elaboración propia

Figura 2 Arquitectura basada en contenedores.

La estructura del laboratorio se organiza en directorios modulares para facilitar la gestión y escalabilidad. En la Figura 3, se muestra como cada contenedor reside en su propia carpeta, conteniendo archivos de configuración específicos como Dockerfile y scripts. Esta organización permite la replicabilidad del entorno y la personalización de componentes individuales sin afectar el sistema general. La raíz del proyecto, que puede colocarse dentro de la carpeta de documentos del usuario, incluye el archivo *docker-compose.yml.*



Fuente: elaboración propia

Figura 3 Estructura de directorios recomendada.

Contenedores preconfigurados

El laboratorio utiliza imágenes especializadas para emular entornos vulnerables y herramientas de evaluación. El archivo que contiene todo el código de la red, los volúmenes, los servicios interconectados y la ubicación de los contenedores es *docker-compose.yml, é*ste y los archivos *Dockerfile* son sensibles a la identación. La Figura 4 muestra la estructura del archivo *docker-compose.yml*.

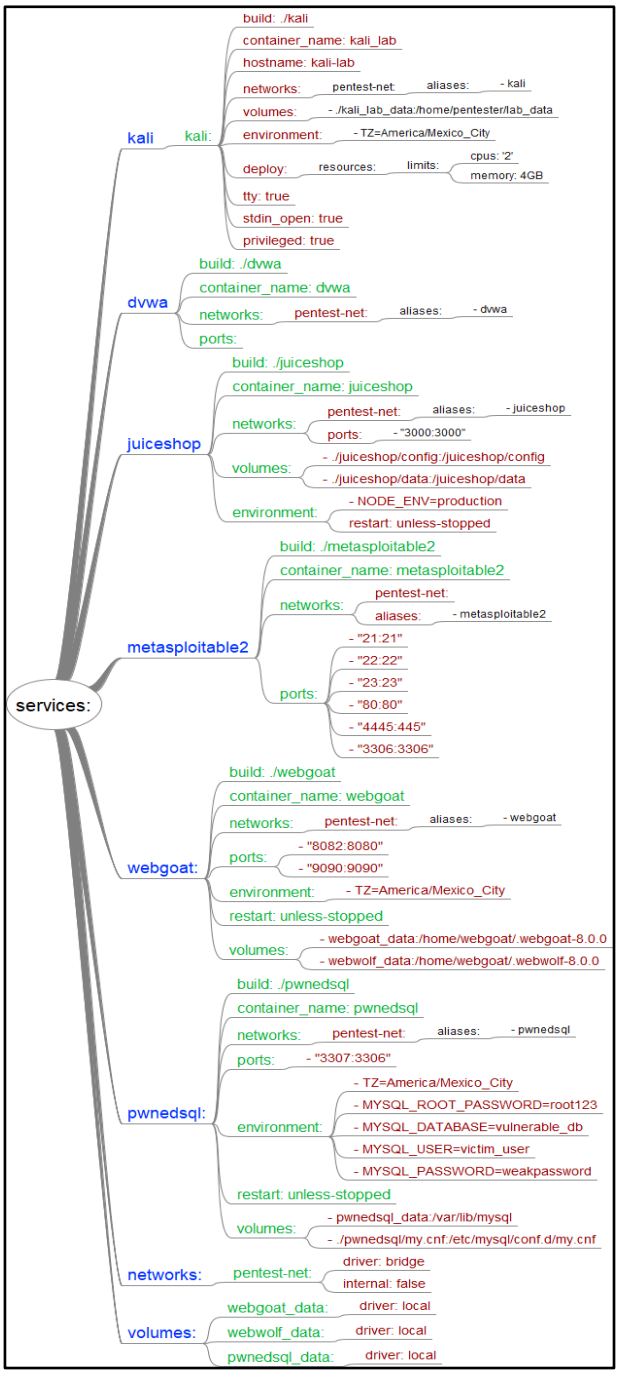
Características de la red aislada

La red personalizada pentest-net se configura como tipo bridge. Esta red permite comunicación exclusiva entre contenedores del laboratorio y opera en modo interno, bloqueando toda conectividad externa o acceso a Internet.

Es importante tomar en cuenta que la red networks debe estar conectada al exterior al momento de configurar y construir los contenedores: internal: false, posteriormente se debe deshabilitar cambiando a internal: true, con ello, se impide el acceso desde el exterior.

Despliegue del contenedor Kali Linux

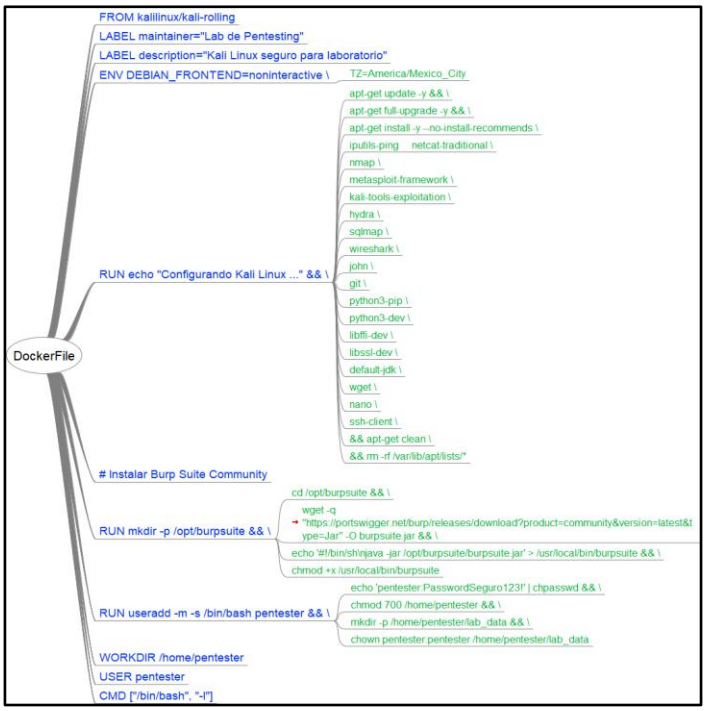
Kali se construye desde la imagen oficial kalilinux/kali-rolling, como un contenedor personalizado con herramientas esenciales como Metasploit, nmap y burp Suite entre otras. Se configura un usuario no privilegiado denominado pentester para prácticas seguras y se montan volúmenes persistentes para almacenar resultados de pruebas.



Fuente: elaboración propia

Figura 4 Estructura del archivo Docker-compose.yml.

El despliegue se ejecuta desde la línea de comandos mediante docker compose build y Docker compose up, integrando los contenedores a la red *pentest-net*. La estructura del archivo *Dockerfile* en la carpeta *kali* queda como se muestra en la Figura 5.



Fuente: elaboración propia

Figura 5 Estructura del Dockerfile Kali.

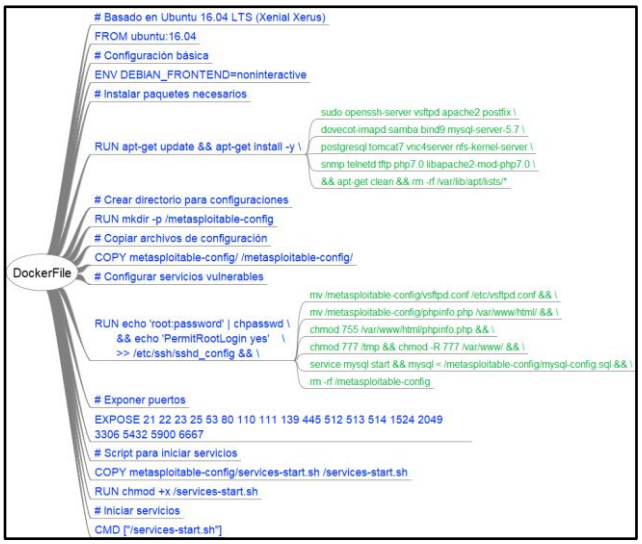
Despliegue de servicios vulnerables para entorno de pruebas

Un servicio vulnerable en un laboratorio de pruebas es un software, aplicación o sistema, intencionalmente configurado con fallos de seguridad conocidos o desconocidos, diseñado para ser explotado de manera controlada con fines educativos, de investigación o evaluación de seguridad.

Servicio SSH/FTP Metasploitable

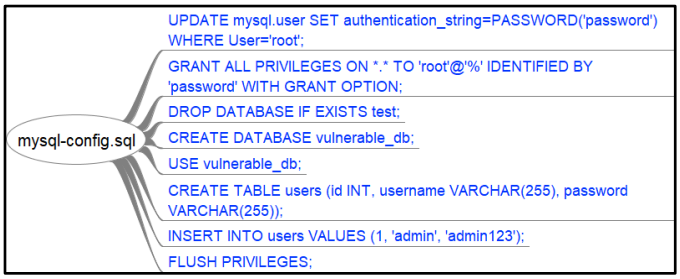
Basado en la imagen 'phocean/msf`, este contenedor emula un sistema Linux con servicios mal configurados. Los puertos 21 (FTP) y 22 (SSH) quedan expuestos intencionalmente con credenciales débiles y versiones de software vulnerables.

Este entorno es particularmente útil para practicar técnicas de escalamiento de privilegios y explotación de servicios de red. En la Figura 6 se muestra la estructura del archivo *Dockerfile* en la carpeta *metasploitable2.* El primer archivo de configuración es *mysql-config.sql* con la estructura que se muestra en la Figura 7. Nótese que se usa la palabra $password$ y usuario $root$ como autentificación. En la tabla users se agrega también el usuario $admin$ y clave $admin123$.



Fuente: elaboración propia

Figura 6 Estructura de metasploitable2.



Fuente: elaboración propia

Figura 7 Instrucciones de mysql-config.sql.

El código del archivo del archivo *phpinfo.php* es:

<?php

phpinfo();

?>

El código del archivo *postgres-config.sql* queda de la siguiente manera:

ALTER USER postgres WITH PASSWORD 'postgres';

CREATE DATABASE vulnerable_db;

\c vulnerable_db

CREATE TABLE users (id INT, username VARCHAR(255), password VARCHAR(255));

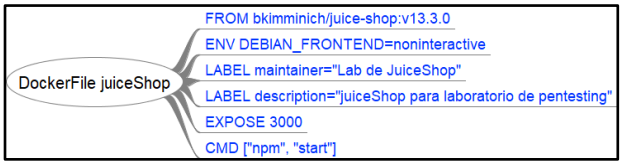
INSERT INTO users VALUES (1, 'admin', 'admin123');

Se usa como clave lo mismo que el nombre de usuario y se inserta el usuario admin y clave $admin123$. El archivo de configuración vfstpd.conf se modifica para permitir

la escritura lo que hace posible crear, modificar y eliminar archivos y directorios y también permite el acceso a usuarios anónimos sin clave. Como el usuario anonymous/ftp o cualquier correo como contraseña. Permite también subir archivos y conectarse en texto plano al servicio de ftp. Con xferlog_enable=YES, sólo se guardan las transferencias y no las órdenes completas para ello se necesitaría log_ftp_protocol=YES. Sin la opción chroot_local_user=YES, los usuarios pueden navegar por todo el sistema. Tampoco hay restricciones de IP ni tasa límite, lo que permite ataques de fuerza bruta. Se pueden agregar usuarios adicionales con la siguiente orden en *Dockerfile*: *RUN useradd -m ftpuser && echo "ftpuser:password123" | chpasswd*. Y crear también archivos sensibles: echo *"Secreto:credenciales=admin:password" > /var/ftp/confidencial.txt*.

OWASP Juice Shop

Se implementa utilizando la imagen oficial **bkimminich/juice-shop**, una aplicación web moderna diseñada específicamente para prácticas de seguridad. La orden de despliegue configura el servicio en el puerto 3000 tanto en el host como en el contenedor, integrado directamente a la red aislada del laboratorio. La aplicación ofrece más de 100 vulnerabilidades intencionales de OWASP Top 10, incluyendo inyecciones SQL, XSS, y fallos de autenticación [OWASP,2025]. La Figura 8 muestra el contenido del archivo *Dockerfile*. Se realiza una prueba de acceso desde el navegador con la siguiente url: http://localhost:3000



Fuente: elaboración propia

Figura 8 Estructura del archivo juiceshop.

Damn Vulnerable Web App (DVWA)

La imagen **vulnerables/web-dvwa** proporciona un entorno PHP/MySQL preconfigurado con vulnerabilidades clásicas. Al mapear el puerto 8080 del contenedor al puerto 80 en el host, se replica un entorno web realista. El sistema

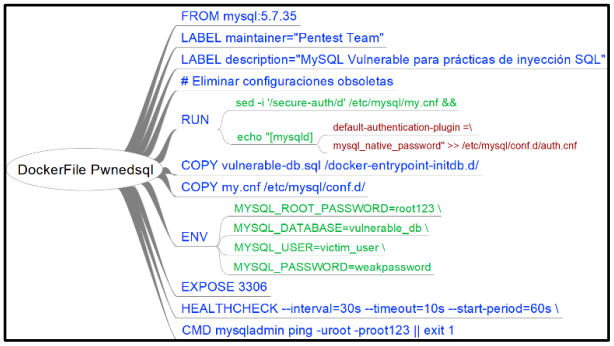
incluye un panel de configuración de seguridad ajustable que permite modificar el nivel de dificultad para las pruebas, desde bajo hasta imposible. En la Figura 9 se puede ver la estructura del archivo *Dockerfile*. Se realiza una prueba de acceso desde el navegador con la siguiente url: localhost:8080

Fuente: elaboración propia

Figura 9 Estructuctura de web-dvwa.

Base de Datos MySQL Vulnerable

La imagen **pwnedsql/mysql-vuln** despliega una instancia de MySQL con múltiples vulnerabilidades de inyección SQL preconfiguradas. El puerto 3306 queda accesible para permitir pruebas directas de explotación de bases de datos. Incluye conjuntos de datos de ejemplo con información sensible mal protegida, ideal para ejercicios de extracción de datos. La Figura 10 muestra la estructura en el archivo *Dockerfile* correspondiente. Se configuran dos archivos que deben de estar en la carpeta **pwnedsql**, el primero es *my.cnf* y el segundo es *vulnerable-db.sql*. En éste último es donde se crea una base de datos con vulnerabilidades conocidas.



Fuente: elaboración propia

Figura 10 Estructura de pwnedsql.

Verificación del despliegue

Para confirmar el correcto funcionamiento de todos los servicios:

- Ejecutar docker ps para verificar el estado de los contenedores

- Realizar pruebas de conectividad básica desde el contenedor Kali a cada uno de los contenedores con el comando: docker exec -it kali_lab ping metasploitable2
- Verificar los logs específicos de cada servicio, por ejemplo el de metasplotable2 con el comando: docker logs metasploitable2

Consideraciones de Seguridad

Cada servicio está diseñado con propósitos educativos y debe permanecer aislado en la red interna del laboratorio de pruebas. Después de configurar adecuadamente los contenedores se recomienda:

- Restringir el acceso físico a la red. Cambiar las instrucciones internal: false a internal:true.
- Monitorear el tráfico de red durante las pruebas.
- Eliminar los contenedores al finalizar las sesiones de práctica.
- Nunca exponer estos servicios en redes públicas o productivas.

3. Resultados

Se llevó a cabo una evaluación del rendimiento de la tecnología de contenedores Docker en un parque heterogéneo de 40 equipos informáticos, con el objetivo de determinar su viabilidad para la realización de prácticas didácticas de ciberseguridad. El proyecto completo puede descargarse desde: *https://github.com/nalonzo/ciberseguridad.*

La muestra incluyó equipos con distintos niveles de rendimiento, categorizados:

- Un primer grupo compuesto por equipos de bajo rendimiento, que incluían modelos con procesadores Intel Pentium Celeron y $4\ GB$ de RAM, adquiridos entre 2012 y 2016, con procesadores Intel Core i3 y $6\ GB$ de RAM del 2012.
- Un segundo grupo lo integraron equipos de rendimiento medio, equipados con procesadores Intel Core i5 y $8\ GB$ de RAM, fabricados en 2019.
- Finalmente, un tercer grupo consistió en equipos de alto rendimiento, con procesadores Intel $Core\ i7$ y $AMD\ Ryzen\ 7$, y $16\ GB$ de RAM.

Para la instalación del entorno Docker en los equipos de bajo rendimiento, fue necesario deshabilitar la verificación de compatibilidad de hardware, permitiendo así la instalación del sistema operativo Windows 11, el cual no es compatible de forma nativa con dichos equipos. Cabe destacar que un requisito fundamental para el correcto funcionamiento de Docker Engine es que el sistema operativo Windows 11 se encuentre completamente actualizado, garantizando así la instalación y operatividad estable de Windows Subsystem for Linux 2 ($WSL\ 2$), un componente esencial para Docker Desktop en esta plataforma.

El resultado más significativo de esta evaluación fue que, una vez superadas las barreras de instalación iniciales, el desempeño en la ejecución de los contenedores Docker fue notablemente similar en todos los grupos. Tanto los equipos de bajo rendimiento como los de alto rendimiento demostraron una capacidad equivalente para desplegar y ejecutar los entornos de laboratorio de ciberseguridad contenidos en Docker, sin que se observaran diferencias apreciables en los tiempos de carga o en la fluidez durante las prácticas. Esto valida la idoneidad de la tecnología de contenedores para estandarizar entornos prácticos en entornos educativos con infraestructura de hardware diversa, asegurando una experiencia de aprendizaje uniforme para todos los estudiantes.

La implementación del laboratorio basado en contenedores Docker demostró ser altamente efectiva para la creación de entornos de práctica en ciberseguridad. Se logró configurar exitosamente una infraestructura completa que incluye herramientas de pentesting (Kali Linux) y múltiples servicios vulnerables (OWASP Juice Shop, DVWA, Metasploitable) en un entorno aislado. La red personalizada pentest-net operó correctamente en modo interno, garantizando el aislamiento completo del exterior una vez configurada con internal: true.

Los contenedores mostraron un óptimo rendimiento con la asignación de recursos especificada ($4\ GB$ de RAM, 2 núcleos de CPU), permitiendo la ejecución simultánea de múltiples servicios sin afectar el sistema anfitrión. La verificación de conectividad mediante docker exec -it kali_lab ping metasploitable2 confirmó la correcta comunicación entre todos los componentes del laboratorio. El acceso a los servicios se validó satisfactoriamente a través de los puertos designados: Juice Shop en

puerto 3000, DVWA en puerto 8080, y los servicios de Metasploitable en sus puertos respectivos. La estructura modular de directorios facilitó la gestión y replicabilidad del ambiente, mientras que los volúmenes persistentes aseguraron la conservación de los resultados de las prácticas entre sesiones. El proceso de despliegue completo, desde la instalación de los prerrequisitos hasta la puesta en marcha de todos los contenedores, demostró ser reproducible y consistentemente exitoso en equipos con diferentes características de hardware con las configuraciones mínimas especificadas.

4. Discusión

Los resultados obtenidos validan la efectividad del enfoque basado en contenedores para la implementación de laboratorios de ciberseguridad en entornos académicos:

- La arquitectura propuesta aborda exitosamente el desafío de reducir costos sin comprometer la calidad educativa, eliminando la necesidad de hardware especializado y permitiendo la ejecución en equipos estándar.
- La escalabilidad de la solución con contenedores permite adaptar el laboratorio a diferentes necesidades educativas, desde ejercicios básicos hasta escenarios complejos de múltiples capas.
- El aislamiento de red efectivo garantiza que las prácticas de hacking ético se realicen en un ambiente completamente controlado, eliminando riesgos para infraestructuras externas y cumpliendo con los protocolos éticos requeridos.
- La diversidad de servicios vulnerables implementados (aplicaciones web, servicios de red, bases de datos) proporciona un espectro completo de vectores de ataque para prácticas educativas. Esto permite a los estudiantes desarrollar habilidades comprensivas que van desde el reconocimiento básico hasta técnicas avanzadas de explotación y post-explotación.

La principal limitación identificada reside en la familiaridad inicial con tecnologías de contenedores, aunque la documentación detallada mitiga esta barrera. El modelo demostró ser significativamente más económico que laboratorios físicos

tradicionales, con costos marginales por estudiante una vez establecida la infraestructura base. Los principios arquitectónicos y metodológicos documentados pueden extenderse a otros entornos de aprendizaje que requieran ambientes controlados para prácticas técnicas.

5. Conclusiones

La implementación de laboratorios de ciberseguridad mediante contenedores demuestra ser una solución eficaz y accesible para entornos académicos con recursos limitados:

- Los resultados confirman que esta aproximación tecnológica permite crear ambientes de práctica realistas y seguros, capaces de emular vulnerabilidades complejas y vectores de ataque diversos sin comprometer infraestructuras reales.
- El enfoque basado en contenedores ofrece ventajas significativas en términos de escalabilidad, portabilidad y eficiencia de recursos.
- La arquitectura modular facilita la replicabilidad del laboratorio y su adaptación a diferentes necesidades pedagógicas.
- La diversidad de servicios vulnerables implementados proporciona un espectro completo para el desarrollo de competencias técnicas.
- Los estudiantes desarrollan habilidades de pentesting en un ambiente controlado que fomenta experimentación y aprendizaje a prueba y error.

Este modelo representa una alternativa viable y de bajo costo frente a los laboratorios físicos tradicionales o laboratorios en línea, reduciendo barreras de acceso a la educación práctica en ciberseguridad. La documentación y metodología presentadas establecen un marco replicable para instituciones educativas que buscan implementar laboratorios especializados sin inversiones prohibitivas en hardware o infraestructura. Por lo que este trabajo contribuye a democratizar el acceso a la educación en ciberseguridad, proporcionando una base sólida para la formación de profesionales capaces de enfrentar los desafíos actuales del panorama de amenazas digitales.

6. Referencias y Bibliografía

[1] Ahmed, Z., (2023). Docker Technology for Small Scenario-Based Excercises in cybersecurity. Columbus State University, Theses and Dissertations. 503. https://csuepress.columbusstate.edu/theses_dissertations/503.

[2] Chaparro, J, Moreno, J., (2024). Creación de un laboratorio de pruebas de seguridad para fomentar la experiencia de aprendizaje en ciberseguridad. Universidad de San Buenaventura, Colombia. https://bibliotecadigital.usb.edu.co/entities/publication/84d5c6fd-bb16-4e70-af44-2cfa12094a00.

[3] Fahrnberger, G. Greiner, M., Hofbauer, S., Lechner, U., Seiler, A., Strussenberg, J., Wolf, P., (2025). Cybersecurity Awareness Education by Making Ransomware Tangible Securely. Innovations for Community Services. I4CS 2025. Communications in Computer and Information Science, vol 2513. Springer, Cham, pp. 386-414. https://doi.org/10.1007/978-3-031-94263-1_22.

[4] Fernández, M., (2019). Laboratorio de Pentesting basado en tecnología de contenedores. Universidad de Cantabria, España, G2450 Trabajos Académicos. http://hdl.handle.net/10902/16950.

[5] García, J., (2024). Desarrollo de un entorno de simulación como laboratorio de prácticas especializado en ciberseguridad, Universitat Oberta de Catalunya, España, Trabajo de grado. https://openaccess.uoc.edu/server/api/core/bitstreams/77971f44-41a0-457e-9e1c-9cc90a3c2c1d/content.

[6] Hassan, I., (2022). Levereging Apache Guacamole, Linux LXD and Docker Containers to Deliver a Secure Online Lab for a Large Cybersecurity Course, IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, pp. 1-9. https://doi.org/10.1109/FIE56618.2022.9962510.

[7] Irvine, C., Thompson, M., Khosalim, J., (2017). Labtainers: A Framework for Parameterized Cybersecurity Labs Using Containers. The University of Alabama in Huntsville, National Cyber Summit, 5. https://louis.uah.edu/cyber-summit/ncs2017/ncs2017papers/5.

[8] Karagiannis, S., Maragkos, E., Magkos, E., (2020). An Analysis and Evaluation of Open Source Capture the Flag Platforms as Cybersecurity e-Learning Tools, 13th IFIP World Conference on Information Security Education (WISE), Maribor, Slovenia, pp. 61-77. https://doi.org/10.1007/978-3-030-59291-2_5.

[9] Martínez-García, H. A., Camacho-Pérez, E., Chuc-Us, L. B., (2024). EthkLab: Laboratorio de bajo costo para aprendizaje práctico en temas de ciberseguridad. RIDE Revista Iberoamericana para la Investigación y el Desarrollo Educativo, Vol. 15, No. 29. https://doi.org/10.23913/ride.v15i29.2052.

[10] Martínez, S., F., H., (2022). Docker: A tool for creating images and launching multiple containers with ROS OS. Universidad Distrital Francisco José de Caldas, Colombia, Tekhnê, Vol. 19, No. 1, pp. 13–22. https://revistas.udistrital.edu.co/index.php/tekhne/article/view/20339.

[11] Pratama, D., Suryanto, N.,Adiputra, A.A., Le, T.-T.-H., Kadiptya, A.Y., Iqbal, M., Kim, H., (2024). CIPHER: Cybersecurity Intelligent Penetration-Testing Helper for Ethical Researcher. MDPI, Sensors 24. 6878. https://doi.org/10.3390/s24216878.

[12] Santillan, H., Arévalo- Satán, J. A., Wong , P., (2024). Un análisis integral de la infraestructura de ciberseguridad en ambientes académicos. Ingeniería, Revista de la Universidad de Costa Rica, Vol. 35, No. 1, pp. 11–23. https://doi.org/10.15517/ri.v35i1.60075.

[13] Švábenský, V., Čeleda, P., Vykopal, J., Brišáková, S., (2021). Cybersecurity knowledge and skills taught in capture the flag challenges. ScienceDirect, Computers & Security, Vol. 102. https://doi.org/10.1016/j.cose.2020.102154.

[14] Tashkov, D., Pavlova, E., Gagamova, V., Vasileva, V., (2025). Analysis of Cybersecurity Training Platforms and Simulation Environments and Opportunities for Their Integration Into Higher Education. International conference KNOWLEDGE-BASED ORGANIZATION, Nicolae Balcescu Land Forces Academy, Vol. 31. Issue 3, pp. 177-181. https://doi.org/10.2478/kbo-2025-0094.