Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Seguridad Informática

Entorno de seguridad web portable y preconfigurado con aplicación en el ámbito docente

Trabajo fin de estudio presentado por:	Javier Cleva Millor
Tipo de trabajo:	Desarrollo de Software
Director/a:	Rafael Alejandro Rodríguez Gómez
Fecha:	22 de septiembre de 2021

Resumen

El objetivo de este trabajo es dar una solución a la necesidad que tienen de manera incremental las personas que quieren aprender o potenciar sus conocimientos en seguridad informática facilitando y reduciendo el tiempo en tareas de búsquedas de herramientas y de configuración de entornos de trabajo donde poder realizar ejercicios prácticos para su aprendizaje.

Para ello, se ha diseñado e implementado un entorno de seguridad de la información multiplataforma, portable, ligero y conformado con aplicaciones de entrenamiento, herramientas esenciales de ciberseguridad y casos de uso con aplicación docente, acotado todo ello a las vulnerabilidades relevantes en el ámbito web.

Para la implementación se utiliza tecnología de contenedores Docker dado que sus características estudiadas en este trabajo se consideran las más adecuadas

Finalmente, se ha realizado la comprobación del correcto funcionamiento del entorno ejecutando cada uno de los casos de uso que se han diseñado.

Palabras clave: Docker, vulnerabilidades web, contenedores, retos.

Abstract

The purpose of this work is to provide a solution for people who want to learn or enhance their knowledge in computer security, facilitating and reducing the time in tasks of searching for tools and configuring work environments to be able to carry out practical exercises for their learning.

A multiplatform, portable, lightweight information security environment has been designed and implemented, including training applications, essential cybersecurity tools and use cases with teaching applications, all limited to relevant web vulnerabilities.

For the implementation, Docker container technology is used since its characteristics that are studied in this work are considered the most appropriate.

Finally, the correct operation of the environment has been verified by executing each of the use cases that have been designed.

Keywords: Docker, container, CTF, web vulnerabilities.

Índice de contenidos

Índice de figuras

Índice de tablas

1. Introducción

1.1. Motivación:

La ciberseguridad y los cibercriminales se han convertido en algo cotidiano, real y que impacta en nuestro día a día de una manera muy tangible. Muestra de ello es que cada día se encuentran en la prensa más y más noticias relacionadas con la seguridad informática y su impacto en las empresas:

- Servicios públicos que sufren ciberataques como los ataques sufridos por el SEPE durante 2021 (Aguiar, 2021).
- Posibles intromisiones en campañas políticas («La campaña de Hillary Clinton, afectada por el ciberataque al Partido Demócrata», 2016).
- Ataques de cifrado de información como los que están sufriendo en el sector sanitario, que ha visto duplicado su cifra ciberataques en España (Herrador, 2021) y que paralizan empresas y un largo etcétera.

Así, en el informe Ciberamenazas y Tendencias del CCN-CERT (Centro Criptológico Nacional, 2020), se aprecia que, en 2019, gestionó 42.997 ciberincidentes –más de un 11 % con respecto al año anterior–, de los cuales casi un 7,5 % fueron de peligrosidad muy alta o crítica. Por tanto, las empresas tienen la necesidad importante y en algunos casos urgente, de invertir en seguridad informática para poder hacer frente a estas nuevas amenazas.

Este escenario queda patente en los estudios al respecto, donde se observa que la cantidad invertida en esta área es cada vez mayor tanto en contratación de personal especializado como en formación y herramientas para combatir el cibercrimen. En la Digital Trust Survey (PricewaterhouseCoopers, 2021) se puede ver que:

- El 55% de las empresas va a aumentar su presupuesto en ciberseguridad.
- El 51% va a incrementar sus equipos.
- El 50% afirma que la ciberseguridad está presente en cada decisión de negocio.
- El 72% espera fortalecer el área de ciberseguridad.

Además, en el informe Cybersecurity Workforce Study ((ISC)², 2018) se pone de relevancia que la demanda de expertos en ciberseguridad aumenta considerablemente pero que se quedan sin cubrir aproximadamente 3 millones de puestos en todo el mundo y tan solo un

28% de las empresas consideran que tienen la cantidad adecuada de personal dedicado y especializado en ciberseguridad.

Figura 1. *Gap de profesionales en ciberseguridad a nivel mundial*

Fuente: Cybersecurity Workforce Study ((ISC)², 2018)

En base a esta situación, una de las formaciones más demandadas es la relacionada con la seguridad informática ya sea mediante talleres como los que impulsa INCIBE, másteres o especializaciones, además de considerarse como una de las profesiones del futuro, siendo muy posible que incluso llegue a convertirse en una titulación universitaria propia.

Sin embargo, este tipo de formación que, si bien tiene una base eminentemente teórica en conocimientos de ciberseguridad, no se entiende sin una práctica que lo acompañe para su compresión y aplicación en casos lo más reales posibles. Además, se requiere un aprendizaje constante dado que la tecnología cambia a un ritmo elevado y los métodos que los ciberdelincuentes utilizan para encontrar y explotar vulnerabilidades también.

Es por ello por lo que se hace fundamental disponer de entornos donde poder aprender seguridad informática ya sea para la enseñanza o para cualquier profesional que necesita y debe seguir aprendiendo sobre nuevos aspectos prácticos relacionados con la seguridad.

Un entorno informático en general y en particular de seguridad informática, requiere de un conjunto de herramientas relacionadas con la funcionalidad que se persigue cubrir y un hardware donde desplegarlas, configurarlas y usarlas. Esta tarea si bien es sencilla de describir, es compleja de ejecutar puesto que es necesario conocer cuáles son las herramientas más adecuadas, realizar un proceso de instalación y configuración del entorno y finalmente configurar escenarios donde poder aprender aspectos de la ciberseguridad, teniendo que invertir para ello tiempo y dinero antes de comenzar a usarlos y, por tanto, comenzar a aprender.

De esta manera, se concluye que nos encontramos con un problema al que deben hacer frente todo este nuevo conjunto de personas y empresas que se acercan al mundo de la ciberseguridad pero que no tienen por qué conocer sobre estos aspectos técnicos en primera instancia de configuraciones y parametrizaciones y para los que sería de gran utilidad disponer de un entorno preconfigurado, portable y flexible así como de casos de usos docentes más relevantes que incorporen teoría y práctica sobre ejercicios a realizar en relación a la seguridad informática para que la curva de aprendizaje sea efectiva y eficiente.

1.2. Planteamiento del trabajo

Lo que se pretende en este trabajo es dar una solución a la necesidad que tienen de manera incremental las personas y empresas que quieren aprender o potenciar sus conocimientos en seguridad informática facilitando y reduciendo el tiempo en tareas de búsquedas de herramientas y de configuración de entornos de trabajo donde poder realizar ejercicios prácticos para su uso y aprendizaje.

Así, el objetivo consiste en definir y poner a disposición de los usuarios un entorno de seguridad que sea portable, con bajas necesidades de recursos tecnológicos para su ejecución y preconfigurado con las herramientas esenciales y escenarios docentes para aprender y practicar aspectos fundamentales relacionados con la ciberseguridad.

De esa manera, el planteamiento que se define para la consecución de este objetivo consiste en:

- Conocer los principales tipos de ejercicios de ciberseguridad existentes, así como las herramientas esenciales que se utilizan para su resolución.
- Evaluar, seleccionar y aplicar el uso de una tecnología para el montaje del entorno de seguridad informática que dé cobertura al objetivo fijado como meta.
- Determinar y generar casos de uso docentes más relevantes para aprender mediante la práctica los aspectos teóricos fundamentales sobre los que se sustentan dichos casos de uso y que se ejecutarán sobre el entorno de seguridad informática.
- Acotar el alcance en base al desarrollo de casos de uso docentes y selección de herramientas relacionadas con uno de los aspectos que genera mayor número de vulnerabilidades de seguridad como son las aplicaciones web.

- Implementar y probar los casos de uso docentes generados en el entorno de seguridad para garantizar y verificar que se cubre y cumple el objetivo definido.

1.3. Estructura del trabajo

El presente trabajo se encuentra desarrollado mediante los siguientes capítulos:

- Introducción: describe el contexto en el que se enmarca el objetivo que se desarrolla en el presente trabajo, identificando la motivación y el plan de trabajo.
- Estado del arte: describe el estudio de contexto realizado en relación con la necesidad de aprender y practicar conocimientos de seguridad de la información. Para ello, se estudian los elementos que conforman este proceso comenzando con la identificación de retos existentes, aplicaciones donde ejecutarlos utilizando herramientas de seguridad de la información instaladas en un ordenador personal, usando máquinas virtuales o tecnología de contenedores. Finalmente se realiza el análisis de documentos científicos al respecto de las aproximaciones existentes para generar un entorno de seguridad portable, flexible y con casos de usos docentes para mejorar la curva de aprendizaje de los usuarios, detallando en las conclusiones con este trabajo a la comunidad científica.
- Objetivos y metodología: en él se concretan los objetivos generales y particulares que se pretende plantear y evaluar, realizando una descripción de la metodología implementada para la ejecución del trabajo.
- Identificación de requisitos: donde se aborda el proceso de análisis y selección de las herramientas y aplicaciones que conformarán el entorno de seguridad en base a la relevancia de éstas y al criterio que se ha establecido para determinar el alcance de la primera versión.
- Descripción de entorno de seguridad informática: se describe el diseño técnico, implementación y puesta en marcha del entorno junto con las herramientas y aplicaciones que lo conforman, así como el diseño de los casos de uso docentes.
- Evaluación del entorno de seguridad informática: se presentan los resultados de testear el entorno mediante la ejecución de los casos de uso docentes diseñados como parte del alcance del presente trabajo.

- Conclusiones y líneas de trabajo futuro: finalmente se abordan las principales conclusiones obtenidas en el presente trabajo y se establece una línea de próximos pasos para potenciar el entorno de seguridad informático mediante evoluciones en sus funcionalidades, herramientas, aplicaciones, casos de uso docentes y para aplicar el entorno de seguridad en empresas o sectores de formación como puedan ser universidades para obtener feedback al respecto de su uso y que permita seguir evolucionándolo en base a datos reales obtenidos.

2. Estado del arte

Este apartado describe el contexto del trabajo, así como el detalle del estudio realizado sobre referencias que hayan abordado estudios similares para la resolución del problema identificado en relación con la necesidad de mejorar el proceso de aprendizaje de ciberseguridad.

De esta manera, se persigue que las personas puedan mejorar su curva de aprendizaje en conceptos relacionados con la ciberseguridad para hacerla eficaz y eficiente a través del uso de entornos preconfigurados, portables, flexibles y con casos de usos docentes más significativos para su puesta en práctica de manera sencilla y orientada.

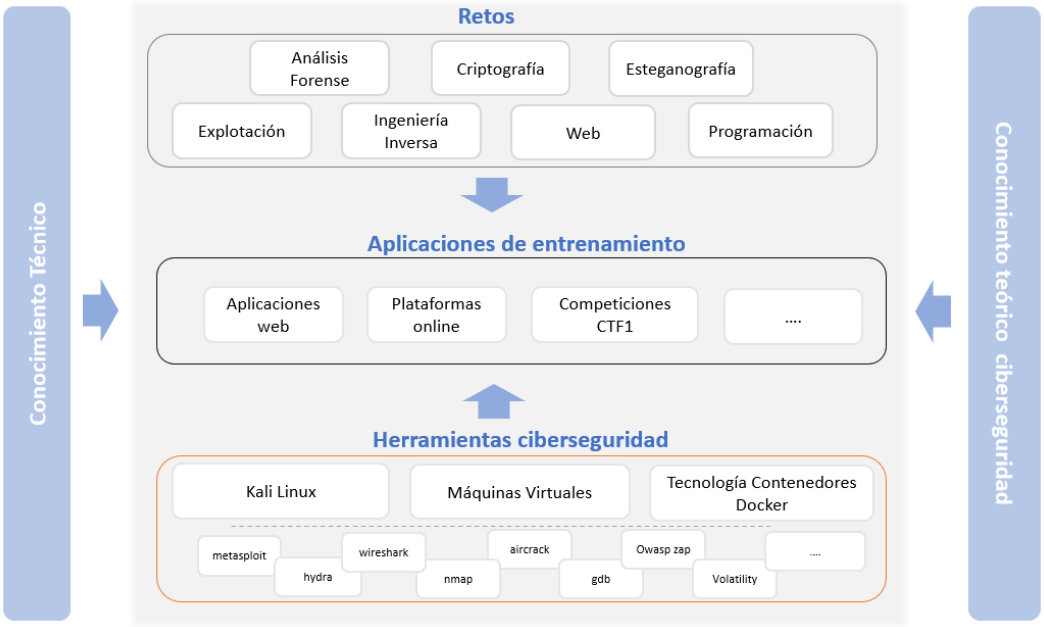
Así, las personas no requieren invertir tiempo en configuraciones ni parametrizaciones técnicas para poder focalizarse en realizar ejercicios de seguridad de la información.

2.1. Contexto

Como se ha descrito en la introducción del documento, cualquier persona que tiene conocimientos sobre la seguridad informática, necesita poder practicarlos y continuar aprendiendo desde un punto de vista práctico, ya sea a nivel básico o avanzado.

Para ello, y como se refleja en el siguiente esquema, las personas deberán utilizar, conocer y hacer frente a los siguientes elementos de contexto:

Figura 2. *Elementos del contexto de aprendizaje en seguridad de la información*



2.1.1. Retos de seguridad de la información

Ante la necesidad de las personas de disponer de ejercicios para aprender de manera práctica sobre seguridad informática ya existen hoy en día una serie de retos, desafíos y ejercicios que cubren esta necesidad, mediante unos eventos denominamos *Capture The Flag*.

Capture The Flag o CTF, que es como se denominan habitualmente, son unos entrenamientos de seguridad a través de unos retos o desafíos informáticos enfocados a la seguridad informática.

Esta definición y la descripción de estos entrenamientos la encontramos en el artículo de INCIBE (*Pablos*, 2014), reseñando que:

- Se trata de eventos donde se resuelven tareas con un grado de dificultad que se va incrementando en cada una de ellas.
- Son eventos que se suelen abordar como competiciones en equipos y en algunos casos con una duración de varios días.
- Los retos se superan al capturar la bandera que suele ser una clave que se debe obtener.
- En algunos de estos eventos se obtienen premios, puntos y prestigio.
- Tras la finalización de un CTF, se publica un *Writeup* donde se describe la solución de manera que todos aprendan al respecto con independencia de haber participado en el reto o a posteriori.

Como denominador común, los CTF, como describe el artículo de INCIBE, se suelen dividir en unas categorías específicas agrupadas por tipos de conocimientos o funcionalidades relacionadas con la seguridad informática.

De esta manera, los usuarios pueden especializarse o seleccionar los bloques funcionales en los que quieran profundizar, aprender o evolucionar:

- Análisis Forense: ejercicios de análisis de imágenes de memoria, discos duros, etc. para encontrar anomalías provocadas por un tercero o atacante.
- Criptografía: relacionados con ejercicios de información cifrada mediante algún sistema o algoritmo concreto.
- Esteganografía: búsqueda de información que está oculta en archivos multimedia como puedan ser en ficheros de tipo imagen, sonido o videos.

- Explotación: relacionados con vulnerabilidades en servidores.
- Ingeniería Inversa: partiendo de un código binario de una aplicación, ser capaz de analizar y conocer su funcionamiento.
- Programación: ejercicios en los que es necesario disponer de conocimientos de programación dado que se deberá desarrollar algún programa de código o algún script para su resolución.
- Web: problemas relacionados con vulnerabilidades de una aplicación web.
- Reconocimiento: ejercicios que consisten en buscar unas marcas o banderas navegando por internet y para ello se deben buscar pistas aplicando por ejemplo conceptos de *footprinting*.
- Trivial: ejercicios más teóricos consistentes en responder correctamente a preguntas de seguridad de la información.
- Misceláneo: modo aleatorio de realización de ejercicios de seguridad que por tanto no está centralizado en una categoría en concreto de las mencionadas, sino que se van realizando de manera aleatoria los ejercicios de cada categoría.

Para poder abordar los retos, los usuarios deberán disponer del conocimiento funcional en ciberseguridad o, en caso de no tenerlo, buscarlo por cuenta propia ya sea en búsquedas, apoyo en terceras personas si se realizan en equipos o a nivel docente si es en un ámbito educativo, siendo deseable y más efectivo si los retos pudieran proveer de la información teórica a los usuarios para facilitar la curva de aprendizaje.

Finalmente, los retos en si mismos son descriptivos sobre los ejercicios propuestos a los usuarios pero que para su desarrollo requieren de aplicaciones y herramientas para su consecución.

2.1.2. Aplicaciones de entrenamiento

Existen numerosos sitios web que gestionan, ofrecen eventos CTF y *writeups* con sus resoluciones, muchos de ellos para hacerlos en equipo como CTFTime (https://ctftime.org) pero también de manera individual como HackPlayers (https://www.hackplayers.com)

En líneas generales, los usuarios para poder realizarlos deben disponer de sus propios equipos maquetados y configurados con las herramientas de seguridad que puedan ser más

convenientes para realizar los retos, requiriendo para ello de un esfuerzo previo de conocimiento de herramientas y de instalación y configuración en sus equipos.

Existen otras plataformas que ofrecen retos de seguridad con máquinas virtuales preconfiguradas para poder realizarlos como son:

- HackTheBox (https://www.hackthebox.eu): Plataforma que pone a disposición de los usuarios máquinas virtuales accesibles por VPN para realizar retos de tipo web, de tipo explotación, criptográficos, etc.
- VulnHub (https://www.vulnhub.com): Plataforma que pone a disposición de los usuarios máquinas virtuales para que los usuarios las descarguen en local siendo importante que el equipo de usuario cumpla con los requisitos de Hardware y sistema operativo para que funcione, así como tener un hipervisor. Dispone de un catálogo bastante completo de retos y también segmentados por nivel de dificultad.
- TryhackMe (https://tryhackme.com): Plataforma muy enfocada a la enseñanza de ciberseguridad que ofrece diferentes cursos, apuntes, videos y máquinas virtuales en cloud para hacer los ejercicios propuestos. Tienen un sistema de gamificación por el que te guía a lo largo de los CTF propuestos de manera didáctica, obteniendo puntos e insignias por los retos conseguidos.

También existen otras aplicaciones dedicadas al aprendizaje de conocimientos de seguridad de la información en el ámbito web y que se caracterizan por ser páginas web que contienen vulnerabilidades o errores de diseño y desarrollo que los usuarios deben encontrar utilizando para ello sus propias herramientas que puedan ser necesarias, así como disponer de manera autónoma del conocimiento para poder hacerlo. Los referentes de este tipo de aplicaciones son:

- DVWA (https://dvwa.co.uk): *Damn Vulnerable Web App* (DVWA) es una aplicación web desarrollada con PHP y MySQL con un amplio espectro de vulnerabilidades de seguridad de la información en el ámbito web más relevantes para aprender a explotarlas de manera autónoma.
- OWASP Juice Shop [20] (*Juice Shop - Insecure Web Application for Training | OWASP*, s. f.): aplicación web desarrollada por la empresa OWASP (https://owasp.org) que es una fundación sin ánimo de lucro que trabaja para mejorar la seguridad del software.

La aplicación dispone de más de 100 vulnerabilidades para su explotación por parte de los usuarios.

De esta manera, en líneas generales, se puede apreciar que existe un amplio espectro de aplicaciones para el aprendizaje y que el usuario tiene dos opciones para afrontar los retos en dichas aplicaciones:

- Disponer de un equipo personal en donde tenga las herramientas instaladas y configuradas que considere necesarias.
- Utilizar las herramientas que se ponen a disposición de los usuarios en algunas de las plataformas, como, por ejemplo, las que se han descrito a través de máquinas virtuales.

En la primera opción, se requiere que el usuario conozca cuales son las herramientas óptimas que necesita, que disponga de un equipo que cumpla con los requisitos técnicos para realizar la instalación de cada una de ellas y su ejecución, así como saber o aprender a configurarlas convenientemente para uso.

En la segunda opción, el problema identificado es que, o bien debes disponer de un equipo que cumpla los requisitos técnicos para ejecutar las máquinas virtuales o si se usan elementos en red que ofrecen las plataformas, los usuarios no podrán practicar de manera independiente salvo conectándose a dichas plataformas y sus recursos.

2.1.3. Herramientas de ciberseguridad

En relación con cómo puede preparar un usuario su equipo personal para poder abordar los retos descritos anteriormente, existen varias opciones al respecto entre las más utilizadas:

- La primera de ellas es Kali Linux («Kali Linux», 2021), que es un sistema operativo basado en Debian GNU/Linux diseñada principalmente para dar cobertura a las necesidades de funcionalidades de la auditoría y de la seguridad de la información.

  Kali Linux incorpora preinstalados más de 600 programas relacionados (*Kali Linux Tools Listing*, s. f.) y dispone además de paquetes de herramientas agrupados por funcionalidades para facilitar su instalación en función del uso que se pueda necesitar (*Kali Linux Metapackages | Kali Linux Blog*, 2014): top10 de herramientas esenciales, herramientas web, forenses, wireless, etc

Kali permite varias opciones de instalación: como sistema operativo, en máquina virtual a partir de la imagen que proporciona para ello o usado en Live CD, live-usb.

Su uso requiere de conocimientos de Linux básicos para poder instalarlo, así como para configurar los paquetes o herramientas que vienen preinstalados, y conocimientos algo más avanzados para mantener el entorno actualizado, si bien es cierto que dispone de interfaz visual para facilitar algunas de estas necesidades.

- Otra opción es la de utilizar máquinas virtuales en las que ir desplegando las herramientas de seguridad que se necesiten.

  Una máquina virtual es un software que permite utilizar los recursos físicos de un sistema para emular el funcionamiento de un sistema operativo como si se tratase de un ordenador dentro de otro y de manera aislada el uno del otro.

  De esta manera, es posible utilizar varios sistemas operativos en un ordenador y configurar cada máquina virtual con las herramientas y aplicaciones que puedan necesitarse y que además puedan tener alguna restricción de estar solo disponibles para algún sistema operativo específico.

  Por ejemplo, y como se comentaba con anterioridad, un usuario con un ordenador con Windows instalado, podría generar una máquina virtual para instalar y usar Kali Linux.

  Las máquinas virtuales se pueden exportar por lo que podrían dar respuesta a parte del problema, pero, sin embargo, requieren una cantidad de recursos para su uso dado que se va segmentando la capacidad del ordenador matriz entre cada máquina virtual que se incorpore al sistema, lo cual puede ser un problema para personas que no dispongan de equipos potentes además de tener que conocer el uso de las máquinas virtuales, cómo exportarlas, etc.

- Finalmente, también existe la opción del uso de contenedores siendo Docker la más representativa de esta tecnología.

  Los contenedores (*Rodríguez*, 2019) se ejecutan sobre el mismo sistema operativo anfitrión de forma aislada al igual que las máquinas virtuales, pero en este caso no necesitan un sistema operativo propio sino que utilizan la misma infraestructura del ordenador lo que los hace mucho más ligeros.

De esta manera, un contenedor de Docker ocupa mucho menos que una máquina virtual, al no tener que emular todo un sistema operativo lo cual representa un importante ahorro en tiempo y coste.

Básicamente, los contenedores se basan en namespace y en CGroups para aislar cada contenedor dentro del sistema operativo de manera que cada contenedor con su respectiva aplicación solo tiene visibilidad sobre su propio sistema virtual de ficheros, proceso, etc, pudiendo limitar los recursos del ordenador que se le permite utilizar o consumir.

El uso de esta tecnología también requiere de un proceso de instalación de Docker por parte del usuario en su sistema. Este proceso es sencillo siendo la parte más compleja en cuanto a necesidades de conocimiento técnico el conocer cómo usar la herramienta Docker para instalar y arrancar los contenedores para su ejecución.

Por tanto, todas las opciones son funcionalmente válidas para utilizarlas en la creación de un entorno de seguridad de la información y en todos los casos se requiere un conocimiento técnico inicial para instalar el software base en los que se sustentan las tres tipologías. Sin embargo, se selecciona Docker para su uso debido a que:

- Requiere menos infraestructura para la instalación y ejecución de las herramientas.
- Es posible exportar e importar contenedores en cualquier momento y en cualquier sistema de manera que facilita la portabilidad del entorno que se pueda configurar un usuario.
- Es compatible su uso tanto con sistema operativo Windows como Linux y en ambos sistemas operativos se usa de igual manera.
- Si bien es recomendable que los usuarios conozcan los comandos para usar Docker, existe una versión Desktop con interfaz gráfica que facilita su uso.
- Existe la opción de publicar los contenedores Docker que genera un usuario para que otros terceros también puedan hacer uso de ellos.
- Existe un catálogo importante de herramientas y aplicaciones de seguridad de la información incorporadas a Docker para su uso por terceros.

- Se puede ir arrancando solo los contenedores que contienen las herramientas que se necesiten en cada momento por lo que resulta más eficiente desde el punto de vista de uso de recursos del sistema.
- Desde el punto de vista docente permite poder enseñar a los alumnos tan solo instalándose los contenedores preparados para la docencia y no requiriendo así que los alumnos instalen multitud de software en sus equipos.
- En contraposición, usuarios más profesionales preferirán montar sus entornos de seguridad ad-hoc en infraestructuras dedicadas donde tendría más sentido usar alguna de las otras dos alternativas.
- Este trabajo en particular se centrará en usos docentes o de usuarios que no dispongan o quieran invertir en generar una infraestructura dedicada en la que poder montar el entorno de herramientas y aplicaciones.

Dado que Docker ha sido la tecnología que se ha seleccionado para la realización del entorno de seguridad de la información en este trabajo, en los anexos A y B del documento se profundiza más sobre esta herramienta, sus funcionalidades, funcionamiento, componentes, comandos, etc.

2.1.4. Conocimientos técnicos y conocimientos teóricos en ciberseguridad

De manera transversal al contexto de retos, aplicaciones y herramientas de seguridad de la información, los usuarios requieren disponer de conocimientos técnicos para realizar instalaciones, configuraciones y uso de las aplicaciones y herramientas, así como conocimiento teórico en ciberseguridad para poder hacer frente a los retos.

La curva de aprendizaje de los usuarios dependerá en gran medida de estos dos factores que por tanto deben ser tenidos muy en cuenta para lograr que el proceso sea efectivo y eficiente:

- No todos los usuarios tienen un perfil técnico que les permita enfrentarse con garantías al proceso de instalación y configuración de herramientas.
- Esto puede suponer una barrera a la hora de aprender o evolucionar e incluso provocar rechazo por parte de los usuarios por lo que es fundamental poder dar una solución en la que estas actividades técnicas sean las menores posibles.
- La utilización de herramientas y aplicaciones requieren de unas características de infraestructura para su funcionamiento que en ocasiones los usuarios no disponen en

sus ordenadores para hacer frente. Este problema también debe ser tenido en cuenta para buscar soluciones ligeras que puedan ser utilizadas en infraestructuras estándar que pueda tener un usuario medio.

Con respecto a conocimientos teóricos en ciberseguridad, se diferencian entre dos tipos de usuarios para entender su contexto y problemática a la que hacer frente:

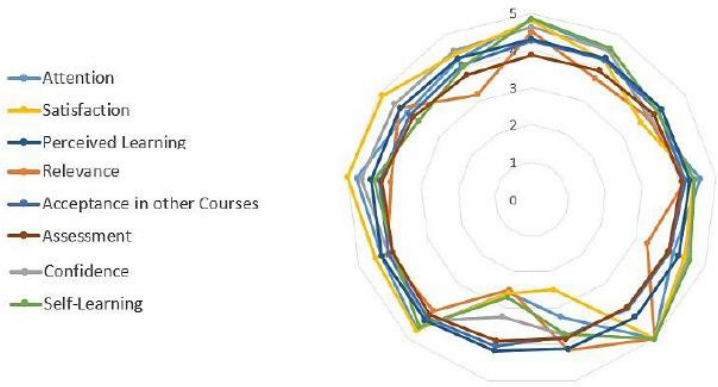
- Un usuario autodidacta que tiene la necesidad de aprender o reforzar conocimiento, cuando se enfrenta a un reto es posible que no tenga todo el conocimiento teórico que pueda necesitar para poder resolverlos por lo que es importante que un entorno de aprendizaje ponga a disposición de los usuarios una parte teórica inicial asociada al reto para vencer estas dificultades.
- En el ámbito docente, los alumnos recibirán la información teórica para su aprendizaje dentro de las dinámicas y materiales de las asignaturas de ciberseguridad. Sin embargo, si bien la teoría es la base fundamental, poder acompañarla con ejercicios prácticos durante su explicación hace que la curva de aprendizaje sea más efectiva.

  Poder poner a disposición de docentes y de alumnos de un entorno ligero en el que poder enseñar de manera práctica los conceptos fundamentales de la ciberseguridad es un factor de éxito en el marco de la enseñanza donde los propios alumnos pueden seguir las explicaciones directamente en sus entornos y asentar así mejor los conocimientos al ver el funcionamiento real de la teoría.

  En este sentido, existen trabajos de investigación para aplicar esta metodología de aprendizaje a nivel universitario donde se aprecian los beneficios anteriormente mencionados.

  - En la Universidad de Ionian (Karagiannis y Magkos, 2020) han aplicado el uso de retos CTFs en el ámbito docente durante el año 2020 obteniendo como resultados que los alumnos se comprometen más en el aprendizaje y adquieren unos conocimientos más asentados.

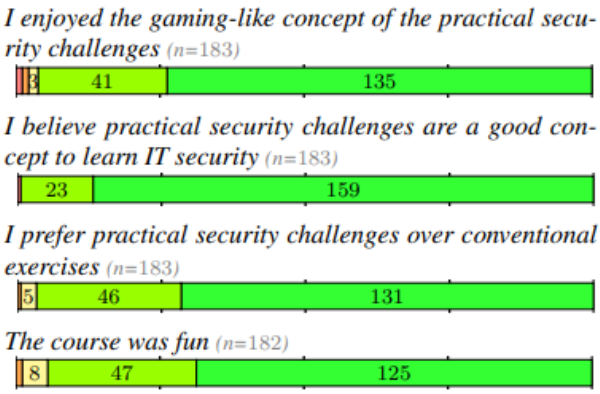
Figura 3. *Resultados obtenidos al aplicar retos en el ámbito docente en la Universidad de Ionian*



Fuente: (Karagiannis y Magkos, 2020)

- En la Universidad de Viena (Dabrowski et al., s. f.) llevan aplicando varios años este tipo de enfoque de aprendizaje basado en retos sobre un alcance acotado de 183 alumnos, obteniendo unos resultados más que satisfactorios en el compromiso, satisfacción y calidad del aprendizaje del alumnado.

Figura 4. *Resultados encuestas al aplicar aprendizaje basado en retos en Universidad Viena*



Fuente: (Dabrowski et al., s. f.)

2.2. Análisis de estudios publicados

En este apartado se expone el resultado de la búsqueda de literatura académica realizada en Google Scholar.

El objetivo que se persigue es el de encontrar y analizar trabajos ya existentes que estén relacionados con la resolución del problema que se pretende solventar para así alcanzar unas

conclusiones que nos permitan definir el aporte que realiza el presente trabajo y sentar las bases para fijar los objetivos.

Tras realizar búsquedas de otros trabajos de entornos de seguridad y la aplicación de Docker se han encontrado los siguientes trabajos como los más significativos para su estudio:

2.2.1. Plataforma de gestión de escenarios de ciberseguridad para aprendizaje y entrenamiento

Este trabajo publicado por el Departamento de Ingeniería de Sistemas Telemáticos de la Universidad Politécnica de Madrid (Barea et al., 2018), parte de la identificación de un problema muy similar en cuanto a la demanda creciente de personal cualificado para el sector de la seguridad informática.

También identifican la necesidad de disponer de laboratorios que sean capaces de optimizar y hacer escalable la realización de ejercicios de seguridad en un ámbito docente ya que, replicar los laboratorios existentes utilizados en empresas especializadas en certificaciones profesionales tiene un alto coste.

Para ello, proponen, diseñan y desarrollan una plataforma para realizar prácticas docentes con una infraestructura dedicada al alumno para que las pueda realizar.

Cada escenario práctico es uno o varios contenedores de Docker configurados con lo necesario para su ejecución.

Además, disponen de un módulo funcional que permite diseñar a los docentes nuevos ejercicios, realizar las puntuaciones y seguimiento de las resoluciones que realizan los alumnos.

Las categorías de ejercicios disponibles en la plataforma son las de:

- Análisis web.
- Análisis forense.
- Criptografía.
- Ingeniería Inversa.
- *Exploiting*.

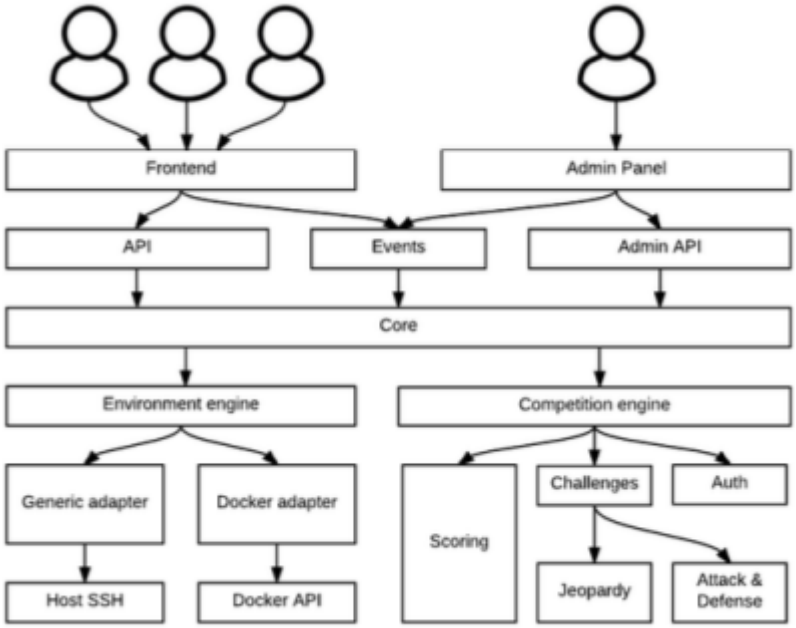
Dentro de la publicación se encuentra descrita la arquitectura de la plataforma con el uso de Docker y también casos de uso realizados para testear su funcionamiento.

Las tecnologías utilizadas en la plataforma son las siguientes:

Figura 5. *Arquitectura técnica de la Plataforma de gestión de escenarios de ciberseguridad para aprendizaje y entrenamiento*

Fuente: (Barea et al., 2018)

Figura 6. *Arquitectura funcional de la Plataforma de gestión de escenarios de ciberseguridad para aprendizaje y entrenamiento*



Fuente: (Barea et al., 2018)

Sobre los retos es interesante el sistema de pistas que se incorporan donde los alumnos si las quieren utilizar deben comprarlas con un coste de puntos que deben haber adquirido tras la realización de otros ejercicios.

2.2.2. Itinerario de retos para la formación de profesionales

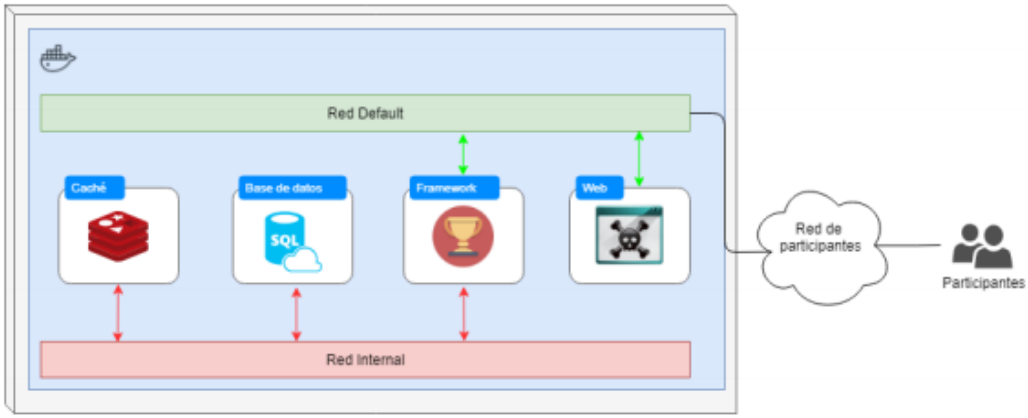
Trabajo de Fin de Master publicado en la Universidad UOC (Montes, 2018) que pretende solventar el problema y la necesidad de formación básica y continuada que necesitan las personas con conocimientos de seguridad informática.

Para ello, se realiza un análisis de los tipos de retos de seguridad y plataformas de CTFs que existen, segmentándolas en las siguientes tipologías:

- Frameworks abiertos: CTFd, HackThe Arch, etc.
- CTFs en línea: CTF Time, Backdoor, etc.
- Fuentes públicas de retos: OWASP Juice Shop.

A partir de este análisis se diseña una arquitectura con Docker en la que poder implementar un evento CTF de la siguiente manera:

Figura 7. *Arquitectura del trabajo de Itinerario de retos para la formación de profesionales*



Fuente: (Montes, 2018)

- Framework para la configuración y gestión del evento CTF por parte de administradores y acceso de los participantes para encontrar los recursos de los retos. En concreto se utiliza el framework CTFd que además ya existe en Docker Hub para su utilización.
- Base de datos MariaDB que es una de las recomendadas por CTFd para su correcto funcionamiento.

- Web vulnerable: Se ha seleccionado la aplicación OWASP Juice Shop que también existe en Docker Hub para su utilización.

Finalmente se ha generado un Docker Compose disponible en el repositorio de Github para su utilización: https://github.com/diegolopmon/CTF-demo/blob/master/docker-compose.yml y que arranca en Docker los contenedores necesarios para hacer funcionar el CTF: Ctfd, MariaDB, Redis y Juice Shop

Es importante reseñar que las herramientas que puedan ser necesarias para resolver los retos no se proporcionan dentro de este entorno de evento CTF.

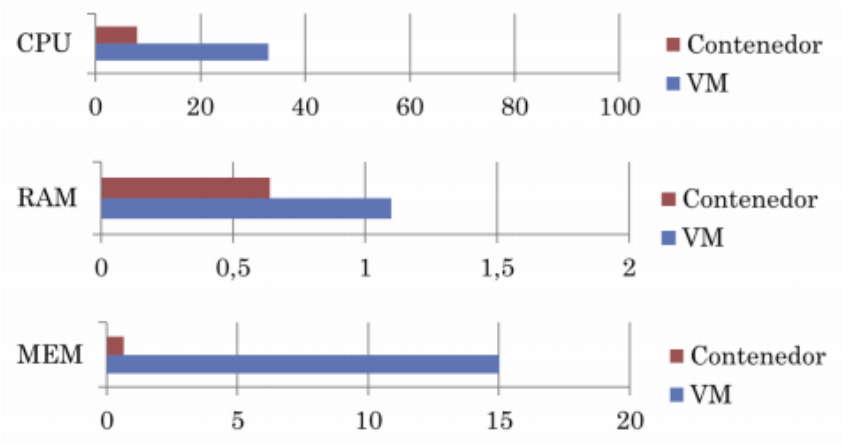
2.2.3. Laboratorio de Pentesting basado en tecnología de contenedores

Trabajo de Fin de Máster publicado en la Universidad de Cantabria (*Fernández*, 2019) que tiene como objetivo la resolución del problema de los recursos crecientes que necesitan las aplicaciones y servicios en general y en particular los relacionados para el uso en la seguridad informática.

Para ello, se realiza un estudio práctico comparativo de uso de recursos entre Máquinas Virtuales vs los contenedores Dockers, desarrollando para ello un caso práctico de reto de seguridad en ambas tecnologías.

En los resultados obtenidos se concluye que la tecnología de contenedores es más eficiente y menos demandante de recursos (memoria, CPU y RAM).

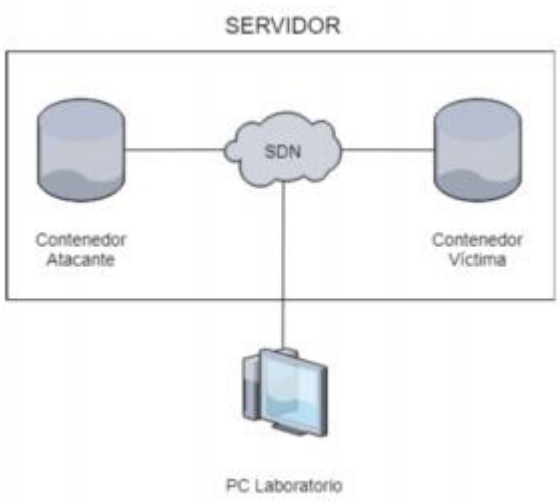
Figura 8. *Consumo de recursos de Máquinas Virtuales vs Contenedores*



Fuente: (*Fernández*, 2019.)

El entorno desarrollado para las pruebas y que se denomina laboratorio de pentesting puede ser utilizado en entornos académicos y en este caso consta de dos contenedores, uno del atacante que replica el sistema Kali y la víctima que replica el sistema Metasploitable.

Figura 9. *Arquitectura laboratorio de pentesting con contenedores*



Fuente (*Fernández*, 2019)

Es relevante remarcar que en el trabajo se realiza también un análisis muy exhaustivo de herramientas de virtualización, herramientas de contenederores con foco en Docker y de las herramientas de Seguridad que existen.

2.2.4. Utilización de paquetes de software vulnerable en el ámbito de la enseñanza

En este trabajo desarrollado en la Universidad Novi Sad de ciencias y tecnología en Serbia (Luburić et al., 2019), realizan un trabajo que parte de la necesidad de disponer de un conocimiento no solo teórico sino también práctico en ciberseguridad para poder hacer frente a todos los ataques cada vez más crecientes.

Para ello, determinan que la utilización en el ámbito de la enseñanza de paquetes de software vulnerable es fundamental para que el alumno pueda asentar los conocimientos de manera correcta y precisa y así, tras terminar la formación, estar realmente preparados para su aplicación.

Plantean generar el siguiente framework de laboratorio para la enseñanza a partir de dicho software vulnerable, ejecutándolo con un ejemplo de aprendizaje de vulnerabilidades de inyección y utilizando la aplicación web vulnerable que proporciona OWASP y que es *The Juice Shop*:

Figura 10. *Framework retos utilizando aplicación web vulnerable preexistente*

Fuente: [48] (Luburić et al., 2019)

En este framework se aprecia la incorporación docente de la preparación de casos de uso teórico-prácticos que se ponen a disposición de los alumnos para que puedan completar los retos que ya se encuentran disponibles en OWASP *Juice Shop* y finalizar con un resumen de lo aprendido tras su finalización.

Es interesante remarcar que, con este laboratorio, no se facilitan dentro del mismo herramientas para la realización de los retos ni se utiliza tecnología de contenedores para desplegar OWASP *Juice Shop*, sino que se utiliza la infraestructura y herramientas que se facilitan en servidores a los alumnos y docentes.

Como conclusiones del trabajo se obtiene que la utilización de aplicaciones vulnerables ya existentes, reducen considerablemente el esfuerzo de tener que crear o simular dichas

vulnerabilidades en un entorno propio docente dentro de la propia universidad, así como una mejora en el aprendizaje a través de la gamificación incorporada en *The Juice Shop* a la hora de ir resolviendo los retos.

2.3. Conclusión

Se ha analizado el contexto del funcionamiento de los retos de seguridad informática que existen, como pueden afrontarlos los usuarios desde el punto de vista de las necesidades que tienen de conocimiento, instalación y configuración de herramientas, así como de la utilización de aplicaciones de entrenamiento y como los retos favorecen el aprendizaje de los aspectos teóricos más relevantes de la seguridad de la información.

Como resultado, se ha ratificado que existe un problema y necesidad no cubierta actualmente para que un usuario disponga de un entorno en el que, con unos requisitos razonables de hardware, pueda ir usando y activando de manera sencilla herramientas y aplicaciones de entrenamiento, sin perder tiempo en búsquedas y sus configuraciones, que sea fácilmente portable a aquellos equipos donde quisiera utilizarlo, con un uso docente teórico-práctico de manera que todo este conjunto de características permita que la curva de aprendizaje de los usuarios sea más eficiente y efectiva.

También se ha analizado el estado del arte evaluando trabajos científicos que están relacionados con el problema que deseamos solventar:

- Se parte de la identificación de un problema que es denominador común de todos los trabajos, la necesidad creciente de formación y de disponer de entornos razonables desde el punto de vista de consumo de recursos.
- La manera de resolverlo consiste en generar una plataforma para crear retos a medida basada en Docker dada su eficiencia desde el punto de vista de uso de recursos frente a otros sistemas como máquinas virtuales o plataformas de seguridad.
- Otras maneras de resolverlos consisten en desplegar en Docker aplicaciones ya existentes con vulnerabilidades para su uso docente.
- En todos los casos se pone de manifiesto que la combinación docente de explicaciones teóricas con retos en plataformas para que los vayan realizando los alumnos, así como la gamificación, permiten obtener mejores resultados en el proceso de aprendizaje.

- En los trabajos analizados, las soluciones son independientes, es decir, el laboratorio o consta de retos hechos a medida y no accesibles por terceros o consta de aplicaciones vulnerables públicas ya existentes. No existe una unificación de todo el contexto de retos – herramientas – aplicaciones – casos de uso dentro de un laboratorio de seguridad de la información.
- Tampoco se encuentran casos de usos docentes publicados para su aplicación con el laboratorio, sino que son generados ad-hoc y para uso interno.
- Las herramientas que puedan necesitarse para resolver los retos en estos trabajos no forman parte de laboratorio que se ha diseñado por lo que el usuario debe o bien tenerlas en sus ordenadores o estar disponibles en alguna infraestructura a la que puedan acceder para así poder realizar los retos.

Finalmente podemos concluir en base a estos análisis realizados que con el presente trabajo vamos a poder aportar una nueva solución que:

- Unifique bajo el mismo entorno de seguridad de la información que se pone a disposición de los usuarios, aplicaciones de entrenamiento, herramientas esenciales de ciberseguridad para su uso con las aplicaciones de entrenamientos u otros fines y casos de uso sobre los aspectos de ciberseguridad más relevantes para su utilización en un ámbito docente.
- Permita a un usuario usar su propio entorno de seguridad informático activando y desactivando de manera autónoma las herramientas o aplicaciones que desee entrenar en cada momento.
- Facilita las aproximaciones iniciales de usuarios con conocimientos teóricos en ciberseguridad para su aprendizaje dado que en el entorno encontrará por dónde empezar al estar formado por herramientas, aplicaciones de entrenamientos y casos de uso esenciales.
- Al utilizar la tecnología de contenedores, es exportable y requiere una infraestructura razonable que por tanto es muy adecuada para un uso docente o para usuarios que no disponen de infraestructuras dedicadas solo para trabajar con ciberseguridad.
- Permite aplicaciones docentes dado que los alumnos podrían acceder de manera sencilla y poco costosa en recursos a las herramientas que habitualmente se utilizan

en la docencia para poder verlas, usarlas y aprenderlas dentro de las propias clases o prácticas de laboratorio.

- La incorporación de casos de uso docentes permite el acercamiento a personas con un conocimiento más básico para saber por dónde empezar y aprender de manera práctica aspectos de seguridad de la información.

3. Objetivos concretos y metodología de trabajo

3.1. Objetivo general

Disponer de un entorno de seguridad informática que cumpla las siguientes características: multiplataforma, portable, ligero y preconfigurado con las herramientas esenciales para poder realizar actividades, ejercicios y prácticas relacionadas con los conceptos de ciberseguridad más relevantes en el ámbito web.

De esta manera, se agiliza la curva de aprendizaje al minimizar el tiempo que se debe dedicar a los procesos de instalación de herramientas y configuración de entornos para realizar estos ejercicios de seguridad, permitiendo centrar los esfuerzos en el aprendizaje en sí mismo de seguridad informática proporcionando para ello casos de usos docentes con la visión teórica esencial relacionada con las principales vulnerabilidades de seguridad informática en el ámbito web y aplicaciones dentro del entorno para su ejecución.

Además, es importante reseñar que, en este entorno de seguridad informática, el usuario debe poder ir activando o desplegando las herramientas que necesite para realizar ejercicios que se le puedan plantear de manera autónoma y bajo la demanda de dichos ejercicios o para explorar sus usos de forma independiente utilizando para ello tecnología de contenedores.

3.2. Objetivos específicos

Para llevar a cabo el objetivo general, se identifica el siguiente listado de objetivos específicos que permiten segmentar el proceso completo de construcción del laboratorio de seguridad:

- Identificar las herramientas de seguridad informática utilizadas para dar cobertura a los principales y más relevantes tipos de ejercicios y prácticas utilizando como referencia las que se usan para la resolución de los ejercicios *Capture the Flag* (CTFs).
- Acotar el alcance del entorno de seguridad en su primera versión para abordar las herramientas, aplicaciones y casos de uso docentes relacionados con los aspectos más relevantes de ciberseguridad en el ámbito web.
- Diseñar el modelo de solución del entorno de seguridad informática con la tecnología de contenedores y que cumpla las características de ser portable, ligero, multiplataforma y preconfigurado, establecidas en el objetivo general que se persigue.

- Diseñar los casos de uso docentes de ámbito web para incorporar en el entorno de seguridad junto con aplicaciones donde poder ejecutarlos convenientemente para adquirir su aprendizaje:
    - Caso de uso de la vulnerabilidad web de inyección.
    - Caso de uso de la vulnerabilidad web de pérdida de autenticación.
    - Caso de uso de la vulnerabilidad web de exposición de datos sensibles.
    - Caso de uso de la vulnerabilidad web de entidades externas XML (XXE).
    - Caso de uso de la vulnerabilidad web de pérdida de control de acceso.
    - Caso de uso de la vulnerabilidad web de secuencia de comandos en sitios cruzados.
    - Caso de uso explotación vulnerabilidad postgres máquina metasploitable.
- Implementar el entorno de seguridad informático con el uso de la tecnología de contenedores Docker como base donde desplegar y configurar las herramientas aplicaciones y casos de uso docentes identificados y definidos.
- Testear el laboratorio ejecutando los casos de uso docentes en el entorno de seguridad informática implementado para verificar que se ha alcanzado y cumplido el objetivo.

3.3. Metodología de trabajo

De cara a alcanzar los objetivos específicos y con ellos el objetivo general, se han determinado una serie de pasos que se han ejecutado para lograr su consecución con la siguiente distribución de esfuerzos invertidos en su realización:

Tabla 1. ***Dedicación de esfuerzos en las tareas realizadas en el trabajo.***

Tarea	% Esfuerzo
Identificación de herramientas de seguridad esenciales	10%
Selección de aplicaciones web de entrenamiento en ciberseguridad	10%
Diseño de casos de uso docentes en el ámbito web de la ciberseguridad	30%
Diseño e implementación del modelo de solución del entorno de seguridad informática	35%
Evaluación del entorno de seguridad informático ejecutando los casos de uso docentes web	15%

3.3.1. Identificación de herramientas de seguridad esenciales

El primer paso consiste en la identificación de las principales herramientas de ciberseguridad que se utilizan en el ámbito de ejercicios, prácticas y resolución de retos como los CTFs y que por tanto son esenciales para su incorporación en el entorno de seguridad informática.

Para ello, se utiliza la división del conocimiento de los tipos de retos más relevantes en ciberseguridad identificados y explicados en el capítulo 2 de este documento, seleccionando los que se consideran más significativos y relevantes:

- Fuerza Bruta (*Bruceforcers*): relacionados con ejercicios en los que hay que aplicar fuerza bruta como por ejemplo para conocer el password de un usuario en concreto.
- Criptografía (*Crypto*): relacionados con ejercicios de información cifrada mediante algún sistema o algoritmo concreto.
- Explotación (*Exploits*): relacionados con vulnerabilidades en servidores.
- Análisis Forense (*Forensincs*): ejercicios de análisis de imágenes de memoria, discos duros, etc para encontrar anomalías provocadas por un tercero o atacante.
- Redes (*Networking*): ejercicios con análisis de información, funcionamiento y configuración de una red.
- Ingeniería Inversa (*Reversing*): partiendo de un código binario de una aplicación, ser capaz de analizar y conocer su funcionamiento.
- Web: problemas relacionados con vulnerabilidades de una aplicación web.

Como punto de partida para el desarrollo de este apartado, se utilizará el reposito de herramientas que se listan en el artículo de GitHub Awesome CF (Singh, 2015/2021) para la resolución de problemas de seguridad informática.

Para cada uno de ellos se investiga y analiza las herramientas más utilizadas para la resolución de los retos, determinando cual o cuales son las más relevantes en base al siguiente criterio establecido:

- Herramientas más valoradas por referencias externas para usar en ejercicios de ciberseguridad.
- Herramientas gratuitas.
- Existencia como herramienta en Kali Linux al ser una distribución referente y muy utilizada en el ámbito de la ciberseguridad.

3.3.2. Definir el alcance del entorno de seguridad informático

Una vez que se dispone del listado de herramientas por tipología de retos con el criterio de relevancia aplicado, el siguiente paso consiste en acotar el alcance para la realización del entorno de seguridad informático que se aborda en el presente trabajo y que permite definir la selección de las herramientas que se van a incorporar en el entorno.

Se ha determinado que el alcance se acota al ámbito web de los retos de ciberseguridad dado que se trata de uno de los aspectos más relevantes y usados hoy en día y para el que existe además listados sobre las vulnerabilidades más frecuentes relativas a la ciberseguridad que se detectan en las aplicaciones (*OWASP*, 2017).

De esta manera, el entorno de seguridad informática debe proporcionar las herramientas esenciales que se utilizan para los retos Web y que además permitan adquirir conocimientos relacionados con las principales vulnerabilidades que se detectan en las aplicaciones web.

3.3.3. Selección de aplicaciones web de entrenamiento en ciberseguridad

Una vez acotado el alcance y seleccionadas las herramientas que lo van a conformar, se realiza un análisis de las aplicaciones web que existen actualmente y que permiten realizar ejercicios de ciberseguridad relacionados con las principales vulnerabilidades OWASP TOP 10 y para los que se utilizan las herramientas esenciales anteriormente identificadas.

El objetivo que se persigue es el de poder incorporar en el entorno de seguridad informática las aplicaciones más relevantes ya preconfiguradas para su utilización por parte de los usuarios para que nuevamente no sea necesario invertir tiempo en búsquedas y configuraciones por parte de dichos usuarios para acceder y usar estas aplicaciones donde practicar, entrenar y aprender sobre ciberseguridad.

Como resultado, se obtiene el listado de las aplicaciones más relevantes que se incorporan al entorno de seguridad informático en el presente trabajo nuevamente acotadas al aprendizaje en el ámbito web de retos de ciberseguridad.

3.3.4. Diseño de casos de uso docentes en el ámbito web de la ciberseguridad

Tras determinar las herramientas, aplicaciones y haber acotado el alcance del entorno de seguridad informática al ámbito web, se procede a diseñar los casos de uso docentes que

permiten a los usuarios aprender los conceptos fundamentales de ciberseguridad a través de su ejecución en este entorno.

Los casos de uso docentes se acotan al aprendizaje de los conceptos relacionados con las principales vulnerabilidades en ciberseguridad de las aplicaciones web publicadas por OWASP TOP 10 (*OWASP*, 2017) y entre las que se han seleccionado las siguientes:

- Caso de uso de la vulnerabilidad web de inyección.
- Caso de uso de la vulnerabilidad web de pérdida de autenticación.
- Caso de uso de la vulnerabilidad web de exposición de datos sensibles.
- Caso de uso de la vulnerabilidad web de entidades externas XML (XXE).
- Caso de uso de la vulnerabilidad web de pérdida de control de acceso.
- Caso de uso de la vulnerabilidad web de secuencia de comandos en sitios cruzados.
- Caso de uso explotación vulnerabilidad postgres máquina metasploitable.

Cada caso de uso se diseña de manera que:

- Incorpore la descripción de los aspectos teóricos que se van a aprender y utilizar.
- Describa la aplicación del entorno de seguridad que se debe utilizar para su ejecución.
- Defina el reto que se debe resolver, incorporando pistas que permitan avanzar a los usuarios en su resolución sólo si fuera necesario.
- Se obtenga una puntuación como resultado de su ejecución que permita autoevaluar el nivel de aprendizaje obtenido en la resolución del reto en base al tiempo que se ha tardado en resolverlo y el número de pistas que ha necesitado aplicar el usuario.

Finalmente es importante reseñar que los casos de uso y su incorporación en el entorno de seguridad informática permiten que la curva de aprendizaje por parte de los usuarios sea más rápida dado que se les facilita la información teórica y práctica que aprender y usar sin que tengan que dedicar tiempos y esfuerzos para su búsqueda y ejecución.

3.3.5. Diseño e implementación del modelo de solución del entorno de seguridad informática

Una vez determinados los requisitos de herramientas, aplicaciones y casos de uso que deben formar parte del entorno de seguridad informática, se procede a realizar su diseño, implementación y puesta en marcha utilizando la tecnología de contenedores Docker.

En primer lugar, se realiza un análisis de la herramienta Docker para conocer qué permite realizar y para diseñar el modelo de solución en base a los componentes de los que se dispone: Docker file, Docker image, Docker container.

Además, se analiza Docker Hub, que es un repositorio que proporciona Docker con componentes ya existentes y a disposición de cualquier usuario para poder descargar y usar con esta tecnología, buscando opciones oficiales que puedan ya existir para las herramientas y aplicaciones seleccionadas para formar parte del entorno de seguridad informática.

Con toda esta información, se procede a definir el modelo de solución del entorno, identificando las posibilidades de configurar agrupaciones de servicios o herramientas que puedan o deban trabajar conjuntamente para su posterior implementación y configuración en Docker, explicando y detallando cada uno de los pasos y decisiones establecidas, así como configuraciones o requisitos para su puesta en marcha que deben ser considerados y ejecutados por los usuarios del entorno.

3.3.6. Evaluación del entorno de seguridad informático ejecutando los casos de uso docentes web

Finalmente, se realiza la evaluación de la funcionalidad y consecución de los objetivos marcados para el entorno de seguridad informática.

Para ello, se ejecutan los casos de uso docentes que se han diseñado para verificar que es viable realizarlos utilizando tan solo las herramientas y aplicaciones que se han implementado en el entorno obteniendo las conclusiones y próximos pasos que se pueden abordar para continuar la evolución del entorno de seguridad informática en futuras líneas de desarrollo.

4. Identificación de requisitos

4.1. Análisis de herramientas de seguridad informática

En primer lugar, se ha realizado un estudio de las herramientas de seguridad informática más esenciales para así poder determinar cuáles son las que deben formar parte del laboratorio.

Para ello se parte del reposito de herramientas que se listan en el artículo de GitHub Awesome CF (Singh, 2015/2021) para la resolución de problemas de seguridad informática y agrupado por los segmentos de tipologías de problemas más relevantes que son:

- Fuerza Bruta (*Bruceforcers*): relacionados con ejercicios en los que hay que aplicar fuerza bruta como por ejemplo para conocer el password de un usuario en concreto.
- Criptografía (*Crypto*): relacionados con ejercicios de información cifrada mediante algún sistema o algoritmo concreto.
- Explotación (*Exploits*): relacionados con vulnerabilidades en servidores.
- Análisis Forense (*Forensincs*): ejercicios de análisis de imágenes de memoria, discos duros, etc para encontrar anomalías provocadas por un tercero o atacante.
- Redes (*Networking*): ejercicios con análisis de información, funcionamiento y configuración de una red.
- Ingeniería Inversa (*Reversing*): partiendo de un código binario de una aplicación, ser capaz de analizar y conocer su funcionamiento.
- Web: problemas relacionados con vulnerabilidades de una aplicación web.

En el siguiente paso, se definen los ejes de valoración que se han aplicado sobre cada una de las herramientas asociadas a dichos bloques:

- Herramientas más valoradas por referencias para usar en ejercicios de ciberseguridad: este criterio de valoración persigue identificar la relevancia de las herramientas de ciberseguridad al ser de las más encontradas en diferentes artículos y estudios en internet sobre este tipo de herramientas de seguridad de la información y en donde además se establece un ranking de las más utilizadas. Para ello, se ha realizado una búsqueda y análisis de resultados sobre las siguientes referencias (*Ramiro*, 2018) (*Z3R0*, 2019) (Israel, 2020) (Rodolfo, 2019) (*19 Powerful Penetration Testing Tools Used*

By Pros in 2021, 2021) (Porup, 2020) (*Poston*, 2021) donde se han realizado estudios en este sentido, de manera que:

  - Se establece un sistema de puntuación, sumando cada ocurrencia de aparición de una herramienta en alguna de las referencias anteriormente citadas.
  - No se ha identificado la necesidad de aplicar ponderaciones sobre las referencias identificadas.
  - El resultado final obtenido permite delimitar las más relevantes desde el punto de vista de análisis de terceros actores a partir de las referencias citadas.

- Herramientas gratuitas: entre los objetivos del laboratorio se encuentra que sea usable y portable por lo que, aquellas herramientas que sean relevantes pero que sean de pago, serán penalizadas y descartadas.
- Existencia como herramienta en Kali Linux: finalmente, se establece el eje de valoración sobre la aparición de las herramientas en Kali Linux, distribuciones GNU/Linux que se caracteriza por disponer más de 600 herramientas relacionadas con la ciberseguridad y por tanto es un referente para tener en cuenta de manera que:
  - Se cruza con la fuente oficial de herramientas de Kali Linux (*Kali Linux Tools Listing*, s. f.) para conocer si forman parte de esta lista.
  - En caso de existir, se marca de esa manera, siendo un factor diferencial y complementario a los ejes de valoración anteriormente descritos.

A continuación, se listan los resultados obtenidos para cada tipología de CTF:

Tabla 2. ***Herramientas esenciales para retos de Fuerza Bruta.***

Herramienta	Descripción	Existe en Kali Linux	Referencias externas
Hashcat	Cracker de contraseñas.	x	2
Hydra	Un cracker de inicio de sesión en paralelo que admite numerosos protocolos para atacar.	x	2
John The Jumbo	Versión mejorada de John the Ripper.		1
John The Ripper	Cracker de contraseñas.	x	7
Nozzlr	Marco de trabajo de fuerza bruta, modular y compatible con scripts.		1

Ophcrack	Descifrador de contraseñas de Windows basado en rainbow tables.	x	1
Patator	Herramienta de fuerza bruta polivalente, con un diseño modular.	x	1
Turbo Intruder	Extensión Burp Suite para enviar un gran número de solicitudes HTTP.		0

Tabla 3. ***Herramientas esenciales para retos de Criptografía.***

Herramienta	Descripción	Existe en Kali Linux	Referencias externas
CyberChef	Aplicación web para analizar y decodificar datos.		0
FeatherDuster	Una herramienta de criptoanálisis modular y automatizada.		2
Hash Extender	Una herramienta de utilidad para realizar ataques de extensión de longitud de hash.		1
padding-oracle-attacker	Una herramienta CLI para ejecutar ataques de padding oracle.		0
PkCrack	Una herramienta para romper el cifrado de PkZip.		1
QuipQuip	Una herramienta en línea para descifrar cifrados de sustitución o cifrados vigentes (sin clave).		0
RSACTFTool	Una herramienta para recuperar la clave privada RSA con varios ataques.		1
RSATool	Genera una clave privada con conocimiento de p y q.		2
XORTool	Una herramienta para analizar cifrado xor multibyte.		2

Tabla 4. ***Herramientas esenciales para retos de Exploits.***

Herramienta	Descripción	Existe en Kali Linux	Referencias externas
DLLInjector	Inyecta dlls en procesos.		1
libformatstr	Simplifica la explotación de cadenas de formato.		1
Metasploit	Software de pruebas de penetración.	x	6

one_gadget	Una herramienta para encontrar la execve('/bin/sh', NULL, NULL)llamada de un gadget .		1
Pwntools	CTF Framework para escribir exploits.		1
Qira	Analizador de tiempo de ejecución interactivo QEMU.		1
ROP Gadget	Marco para la explotación de ROP.		2
V0lt	Kit de herramientas de seguridad CTF.		1

Tabla 5. ***Herramientas esenciales para retos de Análisis Forense.***

Herramienta	Descripción	Existe en Kali Linux	Referencias externas
Aircrack-Ng	descifra las claves 802.11 WEP y WPA-PSK.	x	2
Audacity	analiza archivos de sonido (mp3, m4a, lo que sea).		1
Bkhive and Samdump2	volcar archivos SYSTEM y SAM.		1
CFF Explorer	Editor PE.		1
Creddump	VolcaR las credenciales de Windows.	x	1
DVCS Ripper	Rips sistemas de control de versiones (distribuidos) accesibles a través de la web.		1
Exif Tool	Lee, escribe y edita metadatos de archivos.		1
Extundelete	Se utiliza para recuperar datos perdidos de imágenes montables.	x	1
Fibratus	Herramienta para la exploración y el seguimiento del kernel de Windows.		1
Foremost	Extrae tipos particulares de archivos utilizando encabezados.	x	1
Fsck.ext4	Se utiliza para reparar sistemas de archivos corruptos.		1
Malzilla	Herramienta de búsqueda de malware.		1
NetworkMiner	Herramienta de análisis forense de redes.		1

PDF Streams Inflater	Busca y extrae archivos zlib comprimidos en archivos PDF.		1
Pngcheck	Verifica la integridad de PNG y vuelca toda la información a nivel de fragmentos en un formato legible por humanos.		0
ResourcesExtract	Extrae varios tipos de archivos de ex.		1
Shellbags	Investiga los archivos NT_USER.dat.		2
Snow	Una herramienta de esteganografía de espacios en blanco.		0
USBRip	Herramienta forense CLI simple para rastrear artefactos de dispositivos USB (historial de eventos USB) en GNU / Linux.		0
Volatility	Para investigar volcados de memoria.	x	2
Wireshark	Se utiliza para analizar archivos pcap o pcapng.	x	0

Tabla 6. ***Herramientas esenciales para retos de Redes***

Herramienta	Descripción	Existe en Kali Linux	Referencias externas
Masscan	Escáner de puerto IP masivo, escáner de puerto TCP.	x	2
Monit	Una herramienta de Linux para verificar un host en la red (y otras actividades ajenas a la red).		1
Nipe	Es un script para hacer de Tor Network su puerta de enlace predeterminada.		1
Nmap	Una utilidad de código abierto para el descubrimiento de redes y la auditoría de seguridad.	x	5
Wireshark	Analiza los volcados de red.	x	6
Zeek	Un monitor de seguridad de red de código abierto.		0
Zmap	Un escáner de red de código abierto.		2

Tabla 7. ***Herramientas esenciales para retos de Ingeniería Inversa.***

Herramienta	Descripción	Existe en Kali Linux	Referencias externas
Androguard	Aplicaciones de Android de ingeniería inversa.		1
Angr	Marco de análisis binario independiente de la plataforma.		1
Apk2Gold	Otro descompilador de Android.		2
ApkTool	Descompilador de Android	x	2
Barf	Marco de análisis binario e ingeniería inversa.		2
Binary Ninja	Marco de análisis binario.		1
BinUtils	Colección de herramientas binarias.		1
BinWalk	Analiza, realiza ingeniería inversa y extrae imágenes de firmware.	x	2
Boomerang	Descompile los archivos binarios x86 / SPARC / PowerPC / ST-20 en C.		2
cwe_checker	Cwe_checker encuentra patrones vulnerables en ejecutables binarios.		0
demovfuscator	Un desofuscador de trabajo en progreso para binarios movfuscated.		0
Frida	Inyección de código dinámico.		2
GDB	El depurador de proyectos GNU.		2
GEF	Complemento de GDB.		2
Ghidra	Conjunto de herramientas de ingeniería inversa de código abierto. Similar a IDA Pro.		0
Hopper	Herramienta de ingeniería inversa (desensamblador) para OSX y Linux.		2
IDA Pro	El software de marcha atrás más utilizado.		2
Jadx	Descompila archivos de Android.		2
Java Decompilers	Un descompilador en línea para los APK de Java y Android.		1
Krakatau	Descompilador y desensamblador de Java.		1

Objection	Exploración móvil en tiempo de ejecución		1
PEDA	Complemento GDB (solo python2.7).		1
Pin	Una herramienta de instrumentación binaria dinámica de Intel.		1
PINCE	Herramienta de ingeniería inversa / front-end de GDB, centrada en la piratería y la automatización de juegos.		0
PinCTF	Una herramienta que utiliza el pin de Intel para el análisis de canal lateral.		0
Plasma	Un desensamblador interactivo para x86 / ARM / MIPS que puede generar pseudocódigo sangrado con sintaxis coloreada.		1
Pwndbg	Un complemento de GDB que proporciona un conjunto de utilidades para piratear GDB fácilmente.		1
radare2	Un marco de inversión portátil.		1
Triton	Marco de análisis binario dinámico (DBA).		1
Uncompyle	Descompile los binarios de Python 2.7 (.pyc).		2
WinDbg	Depurador de Windows distribuido por Microsoft.		1
Xocopy	Programa que puede copiar ejecutables con ejecución, pero sin permiso de lectura.		1
Z3	Un demostrador de teoremas de Microsoft Research.		1

Tabla 8. ***Herramientas esenciales para retos Web.***

Herramienta	Descripción	Existe en Kali Linux	Referencias externas
BurpSuite	Una herramienta gráfica para probar la seguridad del sitio web.		6
Commix	Herramienta automatizada de inyección y explotación de comandos de sistema operativo todo en uno.	x	1

Hackbar	Complemento de Firefox para una fácil explotación web.		1
OWASP ZAP	Proxy de interceptación para reproducir, depurar y desviar solicitudes y respuestas HTTP		4
Postman	Complemento de Chrome para depurar solicitudes de red.		1
Raccoon	Una herramienta de seguridad ofensiva de alto rendimiento para reconocimiento y escaneo de vulnerabilidades.		1
SQLMap	Herramienta automática de inyección SQL y adquisición de bases de datos. pip install sqlmap	x	5
W3af	Marco de auditoría y ataque de aplicaciones web.	x	2
XSSer	Probador XSS automatizado.	x	1

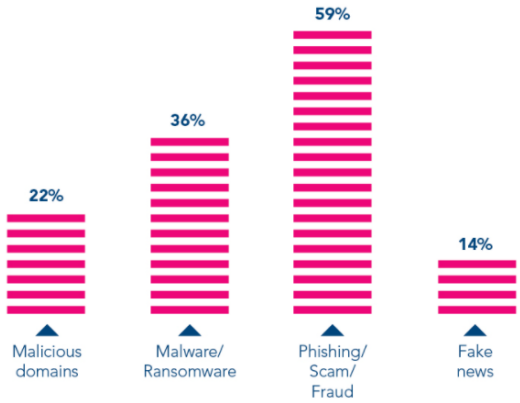
4.2. Criterios establecidos para determinar el alcance del entorno de seguridad informática

Ante la imposibilidad de abarcar en este trabajo la totalidad de tipología de retos y herramientas más relevantes que se utilizan en su resolución, es necesario acotar el alcance para la primera versión de entorno de seguridad informática.

Para establecer este criterio, se parte de las estadísticas de tipología de ciberataques dado que permite centrar el objetivo para abordar los elementos más esenciales que incorporar al entorno y que dé cobertura a aprender y practicar sus conceptos de ciberseguridad para poder hacerles frente.

En el estudio realizado por la Interpol (*INTERPOL*, 2020) se aprecian los siguientes resultados en cuanto al volumen de los tipos de ciberataques que se reciben:

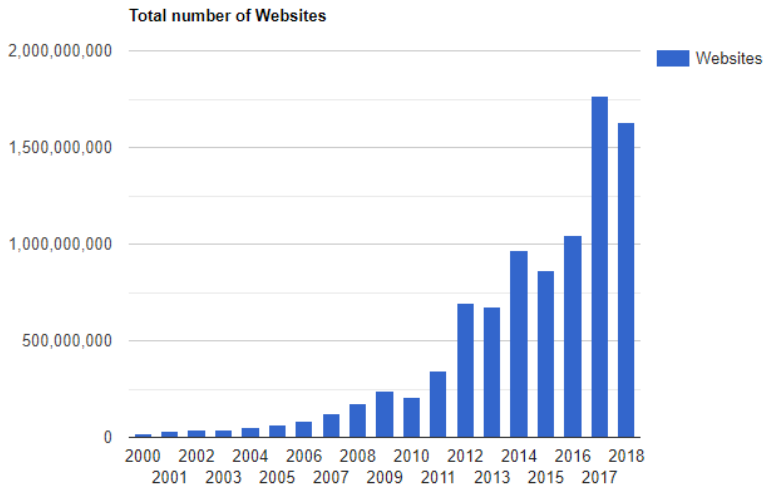
Figura 11. *Volumen de tipos de ciberataques en 2021*



Fuente: (*INTERPOL*, 2020)

De esta manera, los ciberataques relacionados con el Phishing y el Malware son los más habituales, existiendo posibles puntos de entrada para ejecutarlos y siendo uno de ellos las aplicaciones web ya que, como se aprecia en las estadísticas generadas por Internet Live Stats (*Total number of Websites - Internet Live Stats*, s. f.), existen millones en la actualidad:

Figura 12. *Evolución del número de aplicaciones web existentes*



Fuente: (*Total number of Websites - Internet Live Stats*, s. f.)

Dada la relevancia e importancia de aplicar seguridad en las aplicaciones web, es importante que los profesionales en ciberseguridad sean conocedores de las vulnerabilidades más importantes que pueden ocasionar unos desarrollos defectuosos desde el punto de vista de la seguridad de la información para así ser capaces de mitigarlos.

OWASP (https://owasp.org) es una comunidad abierta sin ánimo de lucro dedicada a la seguridad en aplicaciones web que genera, proporciona y mantiene actualizadas herramientas de seguridad, material docente, estándares de seguridad, etc.

Entre todo este material se destaca el denominado OWASP Top 10 (*OWASP*, 2017) que muestra el listado de las 10 vulnerabilidades más frecuentes en aplicaciones web, información generada a partir de:

- Recepción de datos de más de 40 empresas dedicadas a la seguridad de aplicaciones web.
- Recopilación de información de más de 100.000 aplicaciones web reales.

Esta información permite a los participantes en el desarrollo de aplicaciones web el poder revisar y hacer frente a estas vulnerabilidades para garantizar que se desarrollan de manera segura mitigando así la posibilidad de recibir ciberataques.

Las últimas publicaciones de OWASP Top 10 son del año 2013 y 2017 y serán de referencia en el presente trabajo para la selección de casos de uso docentes que estarán ligados con estas vulnerabilidades:

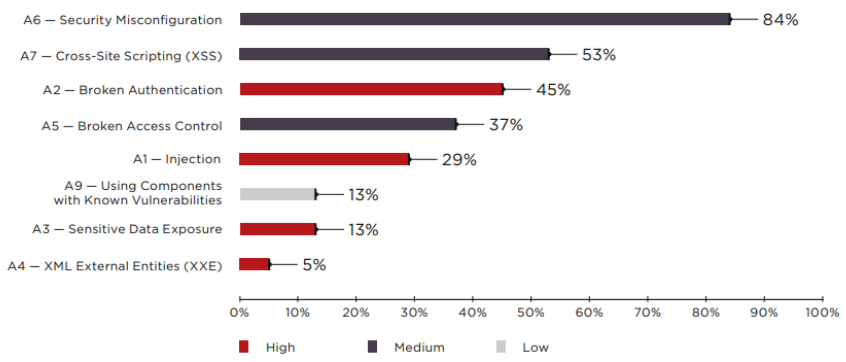
Figura 13. *OWASP Top 10 - 2013 y 2017*

OWASP Top 10 2013	±	OWASP Top 10 2017
A1 – Inyección	→	A1:2017 – Inyección
A2 – Pérdida de Autenticación y Gestión de Sesiones	→	A2:2017 – Pérdida de Autenticación y Gestión de Sesiones
A3 – Secuencia de Comandos en Sitios Cruzados (XSS)	↘	A3:2017 – Exposición de Datos Sensibles
A4 – Referencia Directa Insegura a Objetos [Unido+A7]	∪	A4:2017 – Entidad Externa de XML (XXE) [NUEVO]
A5 – Configuración de Seguridad Incorrecta	↘	A5:2017 – Pérdida de Control de Acceso [Unido]
A6 – Exposición de Datos Sensibles	↗	A6:2017 – Configuración de Seguridad Incorrecta
A7 – Ausencia de Control de Acceso a las Funciones [Unido+A4]	∪	A7:2017 – Secuencia de Comandos en Sitios Cruzados (XSS)
A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)	☒	A8:2017 – Deserialización Insegura [NUEVO, Comunidad]
A9 – Uso de Componentes con Vulnerabilidades Conocidas	→	A9:2017 – Uso de Componentes con Vulnerabilidades Conocidas
A10 – Redirecciones y reenvíos no validados	☒	A10:2017 – Registro y Monitoreo Insuficientes [NUEVO, Comunidad]

Fuente: (*OWASP*, 2017)

Adicionalmente, se disponen de estadísticas de ciberataques (*web application vulnerabilities and threats,* 2020) relacionadas con las vulnerabilidades de OWAPS Top 10 para conocer así cuales son las más frecuentes y las que generan mayor impacto. Esta información también será tenida en cuenta a la hora de acotar los casos de uso docentes que se abordan en este trabajo.

Figura 14. *Vulnerabilidades web más comunes en 2020*



Fuente ((*web application vulnerabilities and threats* , 2020)

Por tanto, en base a este contexto, se define el alcance en este trabajo al ámbito web de seguridad de la información de manera que la selección de herramientas, aplicaciones y casos de uso docentes se realiza bajo este enfoque.

- Con relación a los casos de uso docentes, se seleccionas las siguientes vulnerabilidades para realizar su diseño:
    - Caso de uso de la vulnerabilidad web de inyección.
    - Caso de uso de la vulnerabilidad web de pérdida de autenticación.
    - Caso de uso de la vulnerabilidad web de exposición de datos sensibles.
    - Caso de uso de la vulnerabilidad web de entidades externas XML (XXE).
    - Caso de uso de la vulnerabilidad web de pérdida de control de acceso.
    - Caso de uso de la vulnerabilidad web de secuencia de comandos en sitios cruzados.
- Acerca de las aplicaciones de entrenamiento, se analizarán aquellas existentes que permitan practicar sobre estas vulnerabilidades
- Sobre las herramientas para incorporar al entorno, se seleccionan de entre las analizadas, aquellas más relevantes y que se utilizan para retos relacionados con las vulnerabilidades web:
    - John the Ripper.
    - Aircrack-ng.
    - Whireshark.
    - OWASP ZAP.
    - SQLMap.
    - Burpsuite.
- Adicionalmente y para explorar futuras líneas de actuación, se decide incorporar alguna aplicación y herramientas en el entorno y que permitan practicar y aprender con relación a los retos para descubrir vulnerabilidades en los servidores (*exploits*):
    - Metasploit.
    - Nmap.
    - Caso de uso explotación vulnerabilidad postgres máquina metasploitable.

4.3. Análisis y selección de aplicaciones web de entrenamiento

Una vez identificadas las herramientas más relevantes, se determina como interesante el poder completar el laboratorio con aplicaciones para poder poner en práctica el uso de las herramientas y también para poder utilizar de manera sencilla y sin tener que realizar esfuerzos de configuraciones, el uso de dichas aplicaciones pensadas para practicar y aprender de la seguridad de la información.

Para ello, se ha realizado un estudio de las aplicaciones existentes determinando las siguientes como aquellas a incorporar en el laboratorio.

- **DVWA**: Como se hace referencia en la página oficial de los desarrolladores de DVWA (https://dvwa.co.uk), *Damn Vulnerable Web App* (DVWA) es una aplicación web PHP / MySQL que incorpora en su desarrollo vulnerabilidades para así aprender los procesos de protección de dichas vulnerabilidades web y ayudar a profesores y estudiantes a enseñar y aprender sobre seguridad de las aplicaciones web.

  Entre las funcionalidades de DVWA se destaca lo más relevante:

  - Permite aprender y explotar las principales vulnerabilidades web y más concretamente (Waisen, 2011):
    - Fuerza bruta.
    - Ejecución de comandos.
    - *Cross Site Request Forgery*.
    - Inclusión de ficheros.
    - Inyección SQL.
    - Subida de ficheros.
    - *Cross Site Scripting* en su modalidad de almacenado y reflejado.
  - Dispone de niveles de dificultad configurables por el usuario para ir practicando en función de su conocimiento.
  - También dispone de cierta información formativa para conocer cómo explotar las vulnerabilidades en función de los niveles de dificultad.
  - El motivo de la selección de DVWA es debido a su uso docente extendido en el ámbito web y, además, para la explotación de las vulnerabilidades se deben

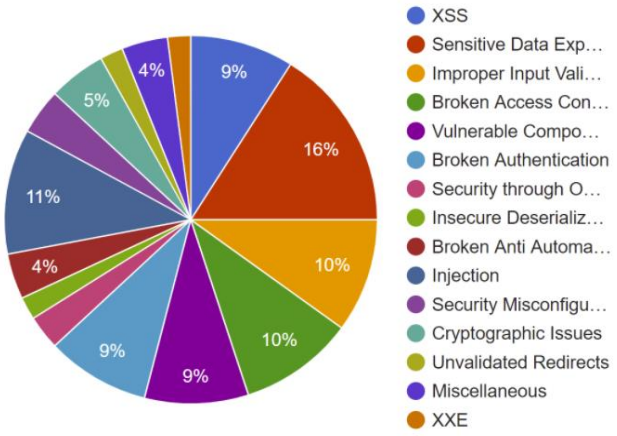
aplicar otras herramientas como SQLMap y OWASP zap por lo que también permite a los usuarios reforzar su aprendizaje y uso.

- **OWASP Juice Shop**: Tomando como referencia la página oficial de OWASP (*Juice Shop - Insecure Web Application for Training | OWASP*, s. f.), esta aplicación es probablemente la aplicación web insegura más moderna y sofisticada y que se puede usar para formaciones o autoformaciones así como para eventos CTFs.

  Lo más relevante es:

  - Open Web Application Security Project® (OWASP https://owasp.org) es una fundación sin fines de lucro que trabaja para mejorar la seguridad del software. A través de proyectos de software de código abierto liderados por la comunidad, la Fundación OWASP es la fuente para que los desarrolladores y tecnólogos protejan la web.
  - OWASP está muy comprometida con la labor docente y de poner a disposición de cualquier usuario herramientas para el aprendizaje. Muestra de ello es que publican en Docker estas aplicaciones y las mantienen constantemente actualizadas.
  - En su web ofrecen numerosa documentación de ayuda y de enseñanza para realizar la explotación de vulnerabilidades de la aplicación.
  - *Juice Shop* está escrito en *Node.js*, *Express* y *Angular*. Las vulnerabilidades encontradas en ella se clasifican cubriendo diferentes tipos de riesgo o vulnerabilidad de listas o documentos conocidos.
  - En concreto, existen más de 100 vulnerabilidades explotables desde la aplicación, englobadas en las siguientes categorías de vulnerabilidades

Figura 15. Clasificación de las vulnerabilidades web de la aplciación Juice Shop



Fuente: (Vulnerability categories · Pwning OWASP Juice Shop, s. f.)

- **Metasploitable**: Metasploitable es una máquina virtual Linux intencionalmente vulnerable. Esta máquina virtual se puede utilizar para realizar capacitación en seguridad, probar herramientas de seguridad y practicar técnicas comunes de prueba de penetración.

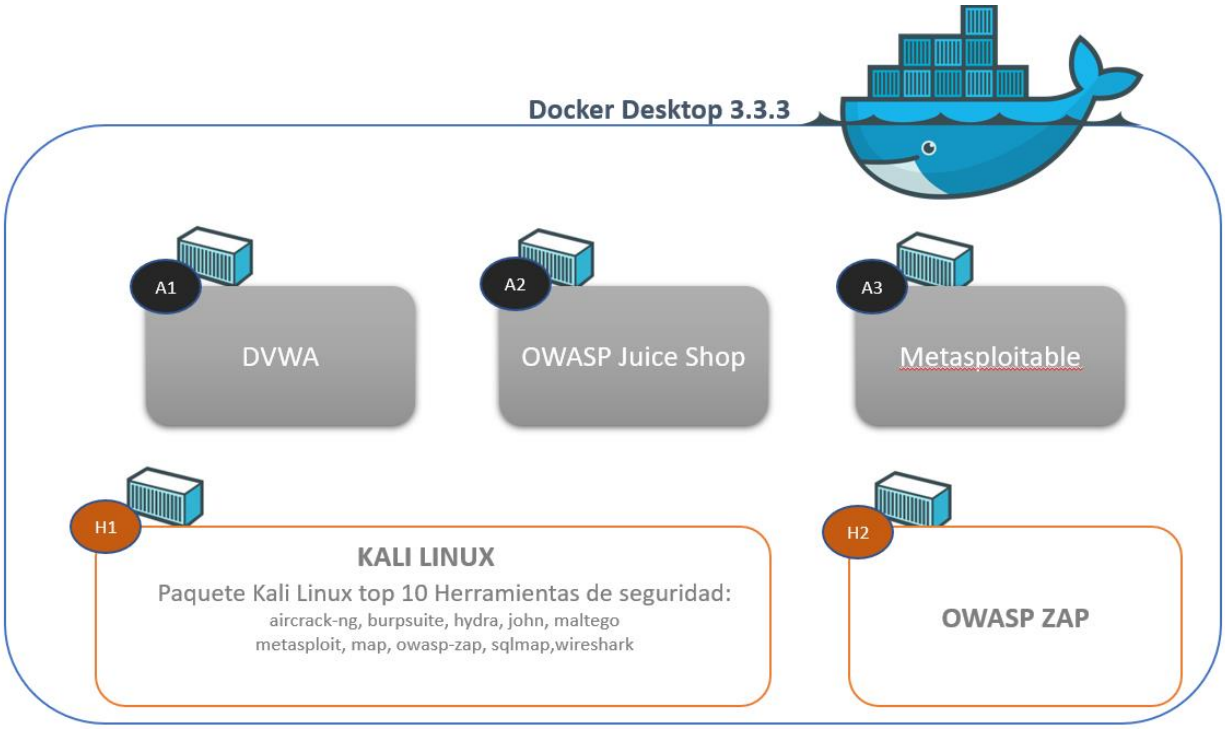
  Lo más relevante es:

  - Permite poner en práctica el uso de herramientas seleccionadas para el laboratorio como son Nmap y Metasploit.
  - Se ha extractado de la referencia (Guarino, 2017), las agrupaciones de vulnerabilidades de esta aplicación analizadas con Acunetix, encontrado 8 amenazas críticas y unas 491 altas quedando estructuradas de la siguiente manera las críticas:
    - 3 vulnerabilidades *Cross site scripting*.
    - 1 vulnerabilidad *Directory Trasversal*.
    - 1 vulnerabilidad *File inclusión*.
    - 1 vulnerabilidad PHP-CGI *remote code execution*.
    - 2 vulnerabilidad SQL *Injection*.

5. DESCRIPCIÓN DEL ENTORNO DE SEGURIDAD INFORMÁTICA

En este capítulo se encuentra la definición del diseño que se ha determinado para el entorno de seguridad informática, tanto de los casos de uso docentes como del modo en el que se ha realizado la implementación con Docker de las aplicaciones (A) y de las herramientas (H) identificadas como esenciales en el capítulo 4 y que, de manera esquemática, ha quedado definido de la siguiente manera.

Figura 16. *Diagrama del diseño del entorno de seguridad informática*



5.1. Diseño de casos de uso docentes para aplicar en el entorno de seguridad

Una vez establecidos los requisitos funcionales en relación con las herramientas, aplicaciones de entrenamiento y haber acotado el alcance del entorno de seguridad informática al ámbito web, en este apartado se incorpora el diseño y la definición de los casos de uso docentes que va a permitir a los usuarios aprender los conceptos fundamentales de ciberseguridad mediante su ejecución en el entorno de seguridad utilizando las herramientas y aplicaciones que lo conforman.

Cada caso de uso docente se diseña bajo una misma estructura de elementos y descripciones:

- Descripción de la vulnerabilidad que se va a estudiar en el caso de uso.

- Impacto de la vulnerabilidad que permita conocer el daño que puede ocasionar.
- Acciones que se pueden aplicar para mitigar la vulnerabilidad.
- Descripción de aspectos docentes adicionales complementarios que se deben conocer y estén asociados a la vulnerabilidad como por ejemplo CORS, CSP, autenticación, etc. y cuyo contenido en detalle se encuentra en el Anexo C para cada caso de uso.
- Definición del reto que se debe resolver y en el que se aplican los conceptos teóricos anteriormente descritos.
- Descripción de las herramientas y aplicaciones de entrenamiento del entorno de seguridad que se deben utilizar para la resolución del reto.
- Descripción de pistas para su uso de manera opcional en la resolución del reto.
- Definición de los parámetros que se establecen para que un usuario que resuelva el reto pueda determinar mediante una autoevaluación el nivel de aprendizaje obtenido en la resolución del reto en base al tiempo que se ha tardado en resolverlo y el número de pistas que ha necesitado aplicar el usuario:
    - La puntuación máxima que se puede obtener es de 10 puntos.
    - Cada pista utilizada resta un punto a la nota total dado que el aprendizaje máximo se obtiene si no se ha necesitado hacer uso de ellas.
    - Se establecen dos umbrales de tiempo, tiempo medio y tiempo máximo de manera que:
        - si se tarda menos que el tiempo medio para su resolución, no hay penalizaciones.
        - si se tarda entre tiempo medio y tiempo máximo, se penaliza con un punto menos.
        - si se tarda más de tiempo máximo, se penaliza con dos puntos menos.

Como los casos de uso docentes se acotan al aprendizaje de los conceptos relacionados con las principales vulnerabilidades en ciberseguridad de las aplicaciones web publicadas por OWASP TOP 10, se ha utilizado la documentación sobre las mismas (*OWASP*, 2017) para realizar su diseño.

5.1.1. Caso de uso de la vulnerabilidad web de inyección

- Descripción de la vulnerabilidad:

La vulnerabilidad de inyección sucede cuando se utilizan datos no confiables como parte de una consulta o comando como en sentencias SQL, No SQL, LDAP, analizadores XML. De esta manera, un atacante puede engañar al sistema introduciendo dichos datos no confiables para explotar el sistema y acceder a datos o información sin tener autorización para ello.

- Impacto de la vulnerabilidad:

  El impacto que puede ocasionar será tan alto como la relevancia que tengan los datos sensibles a nivel de negocio a los que podrá acceder el atacante a través de esta vulnerabilidad ya sea para su consulta y posible divulgación o para su modificación. Un ejemplo sería el acceso a datos bancarios o a la información de contraseñas de los usuarios de la aplicación web.

  Además, la vulnerabilidad es fácil de explotar dado que en una aplicación web estándar se dispone de puntos de acceso para poder hacer la inyección como por ejemplo el formulario de acceso, registro de usuario o formularios de búsqueda de productos.

- Acciones de mitigación:
  - Realizar validación de la entrada de datos mediante listas blancas.
  - Realizar escape de caracteres especiales.
  - Utilización de procedimientos almacenados o APIs seguras para la ejecución de los procesos a partir de los datos de entradas.
- Descripción de aspectos docentes adicionales: En el Anexo C se encuentra el detalle explicativo que se debe conocer sobre la vulnerabilidad de inyección de SQL.
- Definición del reto inyección SQL en aplicación DVWA:

  El objetivo del reto es el de obtener el listado de todos los usuarios de la aplicación DVWA junto con sus contraseñas en claro.

  - Herramientas y aplicaciones de entrenamientos que se utilizan en el reto:
    - Aplicación DVWA.
    - Herramienta sqlmap.
  - Pistas
    - El usuario admin tiene como contraseña el valor password.

        - Es necesario obtener los datos de sesión del usuario para poder utilizar la herramienta sqlmap durante la explotación.
        - Para obtener los datos de la sesión, se puede utilizar un ataque *Cross Site Scripting* Reflejado desde la propia aplicación DVWA e insertando el siguiente código javascript: <script>alert(document.cookie)</script>.
    - Parámetros para la autoevaluación:
        - Penalización por uso de las pistas de un punto por cada una de ellas.
        - Tiempo medio para la resolución del reto: 30 minutos.
        - Tiempo máximo para la resolución del reto: 60 minutos.

- Definición del reto inyección SQL en aplicación OWASP *The Juice Shop*:

  El objetivo del reto es el autenticarse en la aplicación con el usuario administrador y con cualquier otro usuario mediante inyección SQL.

    - Herramientas y aplicaciones de entrenamientos que se utilizan en el reto:
        - Aplicación OWASP *The Juice Shop*.
        - Herramienta OWASP ZAP.
    - Pistas
        - Para obtener el email del administrador y de otros usuarios, se debe acceder a la información pública de las reviews de los productos y ahí se encuentra que el email del administrador es admin@juice-sh.op y, por ejemplo, un usuario sería bender@juice-sh.op
        - La consulta que se ejecuta y que se puede ver trazando los mensajes con la herramienta OWASP ZAP es “SELECT * FROM Users WHERE email = ''' AND password = '20838a8df7cc0babd745c7af4b7d94e2' AND deletedAt IS NULL”.
        - Si se utiliza -- en la consulta SQL, la sentencia ignorará el resto de texto a partir de estos dos guiones al interpretarlo como comentarios y no como parte de la sentencia.
    - Parámetros para la autoevaluación:
        - Penalización por uso de las pistas de un punto por cada una de ellas.
        - Tiempo medio para la resolución del reto: 45 minutos.
        - Tiempo máximo para la resolución del reto: 60 minutos.

5.1.2. Caso de uso de la vulnerabilidad web de pérdida de autenticación

- Descripción de la vulnerabilidad:

  En una aplicación web es fundamental que el proceso de control de acceso funcione correctamente y con la seguridad adecuada. Para ello se deben conocer y controlar los posibles problemas de seguridad que pueden tener los elementos que lo conforman y que son la autenticación, gestión de sesión y autorización.

  Esta vulnerabilidad está relacionada con las funciones de autenticación y la gestión de sesión, permitiendo a un atacante obtener las contraseñas de los usuarios o el identificador de sesión para así poder usarlas y acceder a la aplicación con ellas.

- Impacto de la vulnerabilidad:

  El impacto puede ser alto en función del usuario al que puedan suplantar con la obtención del identificador de sesión o la contraseña de acceso. Por ejemplo, si se trata de una cuenta de administrador, podría llegar a comprometer el sistema y en caso de no serlo, como mínimo podría tener acceso a información sensible del usuario como sus datos personales, bancarios o de medios de pago que podría difundir o vender a terceros para usos maliciosos.

  La vulnerabilidad es fácil de explotar por los atacantes que tan solo necesitan, por ejemplo, herramientas de fuerza bruta para obtener las contraseñas inseguras de los usuarios o utilizar las contraseñas por defecto de los administradores de aplicaciones para acceder con ellas.

- Acciones de mitigación:
  - Implementar doble factor de autenticación.
  - Disponer de un sistema de gestión de contraseñas para garantizar que sean robustas con una longitud adecuada, uso de caracteres especiales y solicitud de cambio de contraseña si se han recibido ataques.
  - Garantizar que no se usan contraseñas por defecto y en particular en las cuentas de administradores.
  - Limitar el número de intentos de autenticación a los usuarios, registrando estos errores e informando de ellos a los administradores.
  - Utilizar un identificador de sesión aleatorio y robusto.

- Descripción de aspectos docentes adicionales: En el Anexo C se encuentra el detalle explicativo del funcionamiento del proceso de autenticación y las consideraciones importantes que se deben conocer al respecto.
- Definición del reto:

  El objetivo del reto es el de obtener la contraseña en claro del usuario administrador cuyo email es admin@juice-sh.op.

  - Herramientas y aplicaciones de entrenamientos que se utilizan en el reto:
    - Aplicación OWASP *The Juice Shop*.
    - Herramienta OWASP ZAP.
    - Herramienta John the Ripper.
  - Pistas
    - Realiza el proceso de *login* con un ataque de inyección SQL y utiliza OWASP ZAP para obtener el token de la sesión.
    - Para poder interpretar el token de la sesión y conocer el hash de la contraseña del usuario administrador, se puede usar la siguiente página web: https://jwt.io/
    - Utiliza John para romper por fuerza bruta el hash del usuario administrador y obtener así la contraseña en claro.
  - Parámetros para la autoevaluación:
    - Penalización por uso de las pistas de un punto por cada una de ellas.
    - Tiempo medio para la resolución del reto: 45 minutos.
    - Tiempo máximo para la resolución del reto: 60 minutos.

5.1.3. Caso de uso de la vulnerabilidad web de exposición de datos sensibles

- Descripción de la vulnerabilidad:

  Esta vulnerabilidad sucede cuando un atacante tiene la capacidad de acceder a datos sensibles que deberían estar protegidos pero que no lo están convenientemente ya sea porque no existe un sistema de cifrado o porque no exista un proceso de gestión de la autorización de acceso a estos datos sensibles ya sea para su consulta o para su modificación.

- Impacto de la vulnerabilidad:

El impacto puede ser alto en función de la información sensible que obtenga el atacante como por ejemplo si se obtienen datos personales, bancarios o credenciales.

Esta vulnerabilidad es menos sencilla de explotar que la de los anteriores casos de uso dado que requiere en general de un primer paso para obtener los datos mediante ataques por ejemplo de *Man in the Middle* y, una vez obtenida la información y en el caso de que exista algún proceso de encriptación sobre esta información, hacer uso de sistemas de fuerza bruta para desencriptar información.

- Acciones de mitigación:
    - Disponer de un catálogo de los datos que permita identificar cuáles son sensibles para aplicarles los controles adecuados que garanticen su seguridad.
    - Almacenar solo los datos sensibles que sean realmente necesarios.
    - Todos los datos sensibles deben estar cifrados y transmitirse de manera cifrada usando TLS.
    - Deshabilitar que se puedan almacenar datos sensibles en cookies.
    - Utilizar APIS robustas para la generación de claves y aplicar algoritmos de cifrado.
    - Almacenar las contraseñas utilizando funciones de *hashing* con factor de trabajo y SALT.
    - Deshabilitar el listado de directorios del servidor web.
    - Garantizar que en las carpetas públicas del servidor no hay información sensible.
- Descripción de aspectos docentes adicionales: En el Anexo C se encuentra el detalle explicativo y las consideraciones importantes que se deben conocer al respecto de la gestión de contraseñas de usuario.
- Definición del reto:

    El objetivo del reto es el de acceder a datos sensibles de la aplicación y en concreto a un fichero de backup del equipo de desarrollo y a un fichero del sistema SIEM que registra ataques y errores sobre la aplicación. En ambos casos será viable acceder a los ficheros y realizar su descarga a pesar de ser datos muy sensibles que un atacante puede utilizar en posteriores ataques.

    - Herramientas y aplicaciones de entrenamientos que se utilizan en el reto:

- Aplicación OWASP *Juice Shop*

- Pistas
  - En la sección About Us de la página web se encuentra la ruta a una carpeta ftp en la que se pueden ver ficheros sensibles por esta vulnerabilidad.
  - Para poder solventar el error en la descarga de ficheros, se debe hacer uso del ataque denominado *Poison Null Byte* (*Emedding Null Code Software Attack | OWASP Foundation*, s. f.), introduciendo en la url de la descarga el valor %00 para así lograr vulnerar revisiones de extensiones de ficheros que realice la aplicación web.
  - Al incorporar el valor %00.md en la url, es importante tener en cuenta que hay que convertir el carater % a %25 de manera que la url queda de la siguiente manera: http://localhost:3000/ftp/package.json.bak%2500.md
- Parámetros para la autoevaluación:
  - Penalización por uso de las pistas de un punto por cada una de ellas.
  - Tiempo medio para la resolución del reto: 25 minutos.
  - Tiempo máximo para la resolución del reto: 45 minutos.

5.1.4. Caso de uso de la vulnerabilidad web de entidades externas XML (XXE)

- Descripción de la vulnerabilidad:

  Ocurre cuando una entrada XML tiene una referencia a una entidad externa y es procesada por un analizador XML mal configurado de manera que un atacante pueda acceder a contenido no autorizado a través de dicha entidad externa pudiendo obtener información sensible de archivos, realizar acciones de denegación de servicios o ejecutar código de manera remota.

- Impacto de la vulnerabilidad:

  El impacto puede ser alto nuevamente debido a que el atacante tiene la posibilidad de obtener información que sea de carácter sensible como por ejemplo contraseñas de los usuarios o provocar una denegación de servicio que imposibilite el uso de la aplicación web y que puede tener repercusiones a nivel del negocio.

En general la vulnerabilidad es explotable de manera sencilla sobre procesadores XML que permiten especificar entidades externas tan solo generando un documento XML que contenta estas entidades externas maliciosas.

- Acciones de mitigación (*XML External Entity Prevention - OWASP Cheat Sheet Series*, s. f.):
    - Utilización de firewalls de aplicaciones web (WAF) para poder detectar y bloquear ataques a este tipo de vulnerabilidad.
    - Utilización de validaciones de lista blanca sobre los documentos XML.
    - Deshabilitar las entidades externas en los procesadores XML.
    - Actualizar los procesadores y bibliotecas XML.
    - Valorar la utilización de JSON.
- Definición del reto

  El objetivo del reto es el de obtener la información sensible del fichero C:\Windows\system.ini o /etc/passwd del servidor aplicando la vulnerabilidad de entidades externas XXE.

    - Herramientas y aplicaciones de entrenamientos que se utilizan en el reto:
        - Aplicación OWASP *Juice Shop*
    - Pistas
        - Una vez logado con cualquier usuario en la aplicación, se debe acceder al formulario Complaint que es desde donde se pueden subir ficheros en la funcionalidad que desde aquí se ofrece.
        - En el fichero xml se debe incorporar la siguiente entidad externa:

          <!ENTITY xxe SYSTEM "file:///C:/Windows/system.ini" >]>.

        - Se puede utilizar la funcionalidad F12 en el navegador para evaluar la respuesta del servidor al procesar el fichero XML y ahí, junto con el error en el procesamiento se verá la información del fichero sensible incorporado como entidad externa.
    - Parámetros para la autoevaluación:
        - Penalización por uso de las pistas de un punto por cada una de ellas.
        - Tiempo medio para la resolución del reto: 25 minutos.

- Tiempo máximo para la resolución del reto: 45 minutos.

5.1.5. Caso de uso de la vulnerabilidad web de pérdida de control de acceso

- Descripción de la vulnerabilidad:

  Una vez que un usuario está autenticado en una aplicación web, como parte de la funcionalidad del control de acceso, es fundamental garantizar en todos los momentos adecuados el proceso de autorización para que el usuario acceda o ejecute procesos o recursos a través de la aplicación web. Errores en esta funcionalidad de autorización son las que explotan los atacantes con esta vulnerabilidad pudiendo de esta manera acceder o modificar información a la que no tienen autorización para realizar estas operaciones.

- Impacto de la vulnerabilidad:

  El impacto puede ser alto respecto al negocio dado que el atacante puede acceder o modificar información sensible como contraseñas, datos bancarios, etc.

  En este caso la explotación de la vulnerabilidad dependerá del diseño e implementación de la aplicación web en la incorporación de un correcto control de acceso de los usuarios solo a información o recursos públicos por defecto y en particular a datos o servicios sensibles en los mínimos casos posibles y previa verificación y autorización para poder hacerlo.

- Acciones de mitigación:
  - Aplicar política de restricción de acceso por defecto.
  - Configuración de manera correcta de CORS.
  - Implementar un único proceso de control de acceso que utilice toda la aplicación web.
  - Deshabilitar el listado de directorios del servidor web.
  - Garantizar que en las carpetas públicas del servidor no hay información sensible.
  - Garantizar que se ejecutan pruebas de control de acceso previas al despliegue e implantación de la aplicación web.
  - Si la aplicación falla, ir a un estado seguro por ejemplo invalidando la sesión del usuario y requiriendo así volver a realizar el proceso de autenticación.

- Descripción de aspectos docentes adicionales: En el Anexo C se encuentra el detalle explicativo y las consideraciones importantes que se deben conocer al respecto del proceso de autorización o control de acceso, así como de la política de seguridad CORS.
- Definición del reto:

  El objetivo del reto es el de encontrar y verificar los siguientes problemas de control de acceso en la aplicación:

  - ✓ El usuario administrador no debería poder eliminar valoraciones de los usuarios en la aplicación.
  - ✓ No se deberían poder crear valoraciones en nombre de otros usuarios.
  - ✓ Un usuario solo debería poder ver en todo momento solo su cesta de la compra.

  El resultado esperado es el incumplimiento de los tres casos anteriormente descritos y que suponen problemas importantes por carencias de control de acceso en la aplicación permitiendo hacer operaciones y ver datos de otros usuarios cuando no se debería.

  Para la realización de reto se proporcionan los siguientes usuarios de la aplicación para su utilización: jim@juice-sh.op - ncc-1701 y admin@juice-sh.op - admin123

  - Herramientas y aplicaciones de entrenamientos que se utilizan en el reto:
    - Aplicación OWASP *Juice Shop*.
  - Pistas
    - La consola de administración donde poder ver y gestionar las valoraciones de los usuarios es: http://localhost:3000/#/administration
    - El componente oculto en el formulario de feedback es el siguiente: <input _ngcontent-c23 hidden id="userId" type="text" class="ng-untouched ng-pristine ng-valid">.
    - Usa la funcionalidad F12 para explorar en *Session Storage* los valores existentes donde se encuentra el bid que está asociado al valor del id del usuario al que pertenece la cesta.
  - Parámetros para la autoevaluación:
    - Penalización por uso de las pistas de un punto por cada una de ellas.

- Tiempo medio para la resolución del reto: 60 minutos.
- Tiempo máximo para la resolución del reto: 90 minutos.

5.1.6. Caso de uso de la vulnerabilidad web de secuencia de comandos en sitios cruzados

- Descripción de la vulnerabilidad:

  Esta vulnerabilidad permite a un atacante ejecutar código en el navegador de la víctima que usa la aplicación web que, no valida datos no confiables, pudiendo provocar un secuestro de la sesión o redireccionarles a otras aplicaciones web maliciosas del atacante.

- Impacto de la vulnerabilidad:

  El impacto es alto dado que el atacante puede realizar un robo de la sesión del usuario para su posterior utilización, puede evitar el proceso de autenticación multifactor, provocar la descarga de software malicioso, incorporar troyanos de autenticación como keyloggers.

  Esta vulnerabilidad es sencilla de explotar dado que existen herramientas que permiten incluso de manera gratuita ejecutar los ataques.

- Acciones de mitigación (*Cross Site Scripting Prevention - OWASP Cheat Sheet Series*, s. f.):
  - Habilitar política de seguridad de contenido (CSP).
  - Utilizar frameworks seguros que están diseñados para prevenir esta vulnerabilidad.
  - Realizar validaciones.
- Descripción de aspectos docentes adicionales: En el Anexo C se encuentra el detalle explicativo y las consideraciones importantes que se deben conocer al respecto del mecanismo de seguridad CSP, así como de los tipos de ataques *Cross Site Scripting*.
- Definición del reto:

  El objetivo del reto es realizar un ataque DOM XSS y XSS Reflejado en la aplicación OWASP Juice Shop utilizando el usuario administrador para ello: admin@juice-sh.op - admin123

  - Herramientas y aplicaciones de entrenamientos que se utilizan en el reto:

        - Aplicación *OWASP Juice Shop*.
        - Es importante reseñar que en este caso se ha incorporado la variable de entorno NODE_ENV para usar la versión insegura de la aplicación con Docker (_vavkamil_, 2019) y así poder ejecutar este reto, siendo importante eliminar el contenedor tras su finalización para no generar situaciones inseguras.
    - Pistas
        - El ataque DOM XSS se puede realizar desde la funcionalidad de búsqueda de la aplicación incorporando ahí el código javascript.
        - El ataque XSS Reflejado se puede realizar desde la funcionalidad de seguimiento de una orden de pedido, incorporando el código javascript como valor del parámetro id en la url.
        - El código javascritp para el ataque XSS Reflejado debe tener en cuenta que será interpretado como una dirección url y por tanto será de la siguiente manera: http://localhost:3000/#/track-result?id=%3Ciframe%20src%3D%22javascript:alert(%60xss%60)%22%3E
    - Parámetros para la autoevaluación:
        - Penalización por uso de las pistas de un punto por cada una de ellas.
        - Tiempo medio para la resolución del reto: 15 minutos.
        - Tiempo máximo para la resolución del reto: 30 minutos.

5.1.7. Caso de uso explotación vulnerabilidad postgres máquina metasploitable

Aunque no se enmarca en los casos de uso docentes sobre aplicaciones web, se considera relevante realizar un caso de uso de retos de explotación (*Exploits*) dado que es un tema relevante y muy utilizado tanto a nivel docente como profesional dando cobertura a los aspectos de *fingerprinting* y *pentesting*.

- Descripción de aspectos docentes:
    - *Fingerprinting* es un proceso de recolección de información en el que se interactúa directamente con un sistema para obtener información, es decir, no se realiza ningún ataque con este proceso sino una recolección de información para poder conocer vulnerabilidades en el sistema y trazar ataques.

Para ello el proceso que se suele realizar en primer lugar es un escaneo de puertos sobre un sistema para conocer cuáles de ellos están abiertos y cuál es el servicio que se expone en cada uno de ellos, su versión, etc.

El uso de la herramienta nmap permite realizar este escaneo de puertos existiendo varios tipos: TCP SYN, TCP UDP, TCK Ack, TCP Null, XMAS, FIN, TCP Idle y siendo importante elegir el adecuado tanto desde el punto de vista de la eficiencia como desde el punto de vista de no ser detectado por el sistema y las defensas que pueda tener activadas.

- Metasploit es una herramienta de *pentesting* que permite desarrollar y ejecutar *exploits* sobre un sistema, siendo fundamental haber realizado en primera instancia un análisis completo de *fingerprinting* para poder realizar el *exploit* adecuadamente.

  Los aspectos más relevantes relacionados con esta herramienta a tener en cuenta son:

  - *exploit*: es un programa de alto nivel que aprovecha una vulnerabilidad existente en un sistema para aprovecharla a través de un *payload* que suele llevar en su interior.
  - *payload*: es un trozo de código que es la que se desea ejecutar en el sistema que se está atacando, que explota la vulnerabilidad y suele estar contenido dentro del exploit.
  - *msfconsole*: es la consola de metasploit que permite hacer las operaciones de *pentesting*. A través de sus comandos se pueden buscar *exploits* en su biblioteca sobre vulnerabilidades de herramientas concretas, configurarlos y ejecutarlos.

- Definición del reto:

  Atacar una base de datos PostgreSQL de la máquina metasploitable que permita obtener usuario y password con los que autenticarse.

  - Herramientas y aplicaciones de entrenamientos que se utilizan en el reto:
    - Aplicación Metasploitable.
    - Herramienta nmap y metasploit.

- Pistas
  - Se puede usar el comando nmap -sV para realizar un escaneo de puertos de tipo TCP SYN y obteniendo las versiones de los servicios expuestos en los puertos abiertos que detecte.
  - El exploit que se puede utilizar para la resolución del reto es auxiliary/scanner/postgres/postgres_login.
  - Se debe configurar la propiedad RHOST con la ip de la máquina metasploitable para que el exploit funcione correctamente.
- Parámetros para la autoevaluación:
  - Penalización por uso de las pistas de un punto por cada una de ellas.
  - Tiempo medio para la resolución del reto: 30 minutos.
  - Tiempo máximo para la resolución del reto: 45 minutos.

5.2. Diseño del modelo de solución usando Docker

Para la implementación del entorno de seguridad informática se ha decidido utilizar la tecnología de contenedores dado que:

- Son más ligeros que otras opciones como las máquinas virtuales.
- Son agnósticos al sistema operativo en el que se ejecutan por lo que su compatibilidad con la tecnología que pueda tener instalada un usuario es muy alta.
- La portabilidad es otra de sus virtudes dado que un contenedor se puede instalar, instanciar, arrancar y parar de manera aislada por lo que el usuario puede controlar qué quiere usar en cada momento.
- Estas características de los contenedores son las que se persiguen cumplir con los objetivos fijados que debe cumplir el entorno.

Docker es la herramienta de contenedores más significativa que existe en la actualidad por lo que se ha procedido a analizar la documentación oficial de Docker y se puede consultar el Anexo A y B dentro de este trabajo para mayor detalle sobre las funcionalidades, su modelo de arquitectura, así como los comandos más frecuentemente utilizados en su uso habitual.

Para la realización del entorno de seguridad informático se ha utilizado e instalado Docker Desktop versión 3.3.3 para Windows 10 – 64 bits y configurado para utilizar Windows

Subsystem for Linux 2 (WSL2) backend, siendo importante reseñar que para ello se necesita cumplir las siguientes características técnicas en el equipo donde se instala:

- Procesador de 64-bits con Second Level Address Translation (SLAT).
- 4GB RAM.
- Habilitar en la BIOS la característica de virtualización.

Se ha determinado usar la opción de WLS2 en vez Hyper – V como backend dado que, con la llegada de Windows 10 2004, Microsoft ha incorporado su propio kernel de Linux a su sistema operativo. De esta manera es posible habilitar Windows Subsystem for Linux 2 (WSL2) y ejecutar contenedores de Docker de forma nativa con lo que el consumo de CPU y memoria se reducen considerablemente y la velocidad de ejecución es mayor.

Continuando con el diseño, una vez definido el entorno y versión que se utiliza para la implementación del entorno de seguridad informática, el siguiente paso consiste en realizar un análisis de las opciones que ofrece Docker Hub en relación a los dos grandes bloques de herramientas y aplicaciones que se han determinado en el apartado 4 de identificación de requisitos que van a conformar este entorno:

- Aplicaciones(A) para prácticas de seguridad de la información:
    - DVWA.
    - OWASP Juice Shop.
    - Metasploitable.
- Herramientas (H) esenciales de seguridad de la información:
    - John de Ripper.
    - SQLMap.
    - Metasploit.
    - Aircrack-ng.
    - Whireshark.
    - OWASP ZAP.
    - Burpsuite.
    - nmap.

Para ello, se ha realizado una búsqueda en Docker Hub sobre las imágenes existentes para estos elementos analizando los resultados y aplicando los siguientes ejes de valoración que

permitan tomar las decisiones de diseño y selección de imágenes ya existentes que en base a estos criterios se consideran aptas:

- La imagen debe contener descripción del funcionamiento claro y preciso.
- El número de descargas y por tanto el uso por otros usuarios debe ser elevado para considerarlo como estable.
- Debe tener una buena valoración por parte de la comunidad Docker que se establece con una puntuación de estrellas.
- Es muy valorable que exista un repositorio en github con los elementos que conforman la imagen dado que así se dispone de más información técnica sobre su configuración y desarrollo.

En base a ello, se han tomado las siguientes decisiones y definición de imágenes a incorporar en el laboratorio:

Sobre las aplicaciones de entrenamiento:

Como resultado final, se han encontrado imágenes Docker de muy buena calidad para su incorporación en el entorno de seguridad informática. Tan solo para la aplicación DVWA, no se encuentra actualmente su última versión disponible en Docker Hub, pero la versión que se ha encontrado y que es la 1.9, es funcional y cumple los objetivos para realizar los retos por lo que se concluye que su utilización es apta y que en un futuro es posible cambiarla por una nueva versión más actualizada que pueda aparecer en Docker Hub.

A1 – DVWA v1.9

- Imagen en Docker Hub: citizenstig/dvwa
- Repositorio github: https://github.com/citizen-stig/dockerdvwa
- Creador: citizenstig (http://golub.pro)
- Downloads: 100K+
- Stars: 64

A2 – Metasploitable

- Imagen en Docker Hub: tleemcjr/metasploitable2
- Repositorio github:
- Creador: tleemcjr (http://www.mcwhortertechnologies.com/)

- Downloads: 100K+
- Stars: 17

A3 – OWASP Juice Shop

- Imagen en Docker Hub: bkimminich/juice-shop
- Repositorio github: https://github.com/bkimminich/juice-shop
- Creador: bkimminich (https://bkimminich.github.io/)
- Downloads: 10M+
- Stars: 106

Sobre las herramientas de ciberseguridad en el ámbito web:

Si bien se han encontrado imágenes en Docker para todas las herramientas esenciales, el enfoque a aplicar será diferente que el utilizado con las aplicaciones de entrenamiento.

Esto es debido a que todas las herramientas que se han catalogado como esenciales se encuentran disponibles en Kali Linux por lo que se ha buscado si existe alguna imagen de Kali en Docker para explorar la vía de su utilización.

Kali Linux dispone de imágenes oficiales en Docker (*Official Kali Linux Docker Images | Kali Linux Documentation*, s. f.) que además son constantemente actualizadas y con una frecuencia semanal. Se trata de versiones de Kali Linux con los mínimos elementos para su funcionamiento y delegando en el usuario para que realice la instalación de los elementos que necesite para sus casos particulares.

Dado que Kali Linux es una fuente oficial fiable en constante desarrollo y actualización, se decide la utilización de su imagen Docker en el laboratorio, siendo necesario generar los mecanismos oportunos para incorporar las actualizaciones o instalaciones de las herramientas esenciales en la imagen de Kali para hacer transparente a los usuarios de estas funciones y para ello se desarrolla una imagen Docker propia para el laboratorio y que a partir de la imagen de Kali, incorpore estos aspectos automáticamente.

Además, en Kali Linux existen paquetes ya predefinidos con sets de herramientas de seguridad informática (*Kali Linux Metapackages | Kali Linux Blog*, 2014). Tras analizar las diferentes opciones y en base a las herramientas seleccionadas en el capítulo 4, se concluye

que el paquete más recomendable para la utilización en el entorno de seguridad es el Kali Linux Top 10:

- Tamaño de Instalación: 3,5 GB.
- Herramientas que lo componen: aircrack-ng, burpsuite, hydra, john, maltego, metasploit, map, owasp-zap, sqlmap,wireshark.

De esta manera, se determina la utilización de una imagen oficial básica en Docker que provee Kali Linux sobre la que se realizará una vez instalada la incorporación del paquete Kali Linux Top 10.

Por otra parte, en relación a la herramienta OWASP ZAP, la empresa que desarrolla esta herramienta, OWASP, está muy comprometida con la formación de seguridad de la información y pone a la disposición de los usuarios dicha herramienta en una imagen de Docker que es constantemente actualizada con nuevas versiones o actualizaciones y que además permite mediante la configuración de la imagen en Docker, establecer un modo visual para su utilización, por tanto, se determina la utilización de esta imagen de Docker dado que permite una experiencia de usuario y funcionalidad igual a realizar una instalación directamente de la herramienta en el ordenador de un usuario.

H1 – Kali Linux

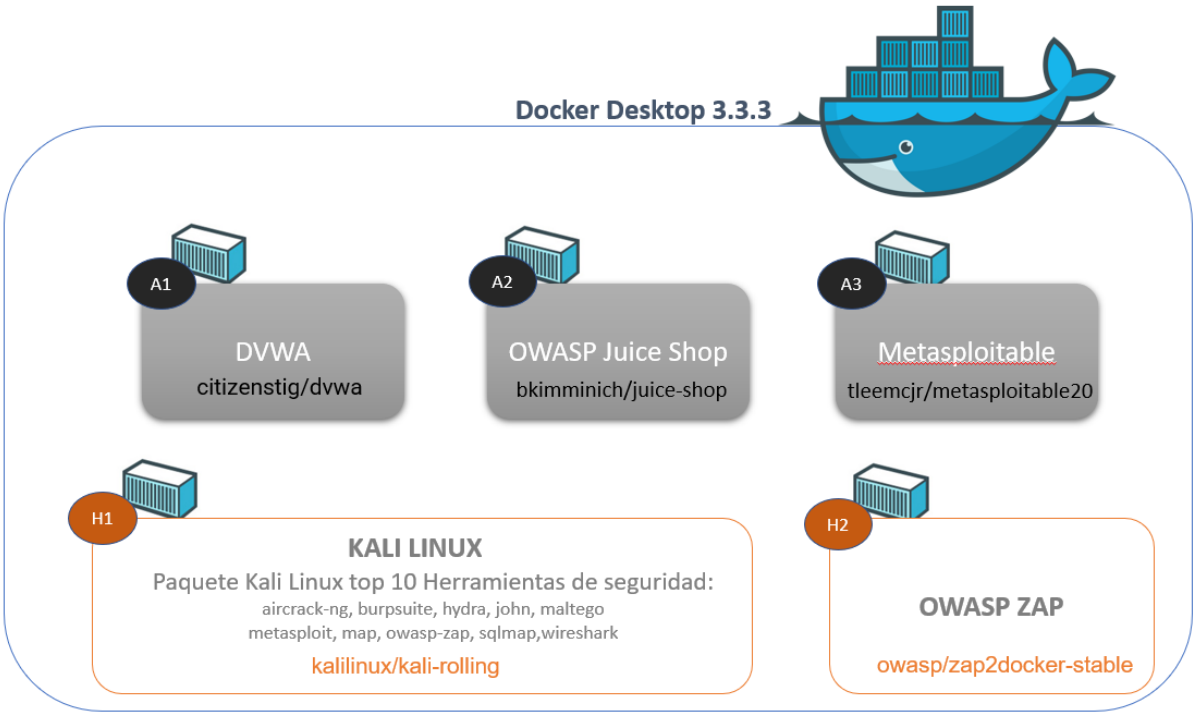
- Imagen en Docker Hub: kalilinux/kali-rolling.
- Creador: kalilinux (https://www.kali.org/).
- Downloads: 500K+.
- Stars: 276.

H2 – OWASP Zap

- Imagen en Docker Hub: owasp/zap2docker-stable.
- Creador: owasp (https://owasp.org/).
- Downloads: 10M+.
- Stars: 147.

De esta manera, se concluye que el diseño del entorno de seguridad informática con herramientas y plataformas de aplicación web queda conformado con Docker y las imágenes de Docker Hub de la siguiente manera:

Figura 17. *Diagrama implantación componentes en el entorno de seguridad informática*



5.3. Implementación y puesta en marcha

Una vez definido el diseño del entorno de seguridad informática con los componentes que lo conforman y las diferentes imágenes de Docker Hub que se van a utilizar, se procede a realizar la implementación y puesta en marcha del entorno.

A continuación, se describe cómo se realiza la implementación de cada una de las aplicaciones y herramientas partiendo de la premisa de disponer de Docker Desktop ya instalado y pudiendo consultar en el Anexo D las capturas de imágenes de la puesta en marcha de los componentes para que sirvan de referencia:

A1 – DVWA

Se utiliza la imagen de Docker Hub citizenstif/dvwa, utilizando los siguientes comandos desde el terminal:

- `docker pull citizenstig/dvwa`: realiza la instalación de la imagen de la aplicación DVWA.
- `docker run -d -p 80:80 citizenstig/dvwa`: crea el contenedor Docker con la aplicación DVWA en el puerto 80.

- La primera vez que se accede a la aplicación, se debe realizar la configuración de la base de datos que utiliza DVWA: http://localhost/setup.php.
- Una vez finalizada esta configuración, ya se puede acceder a la aplicación desde la pantalla de login para su utilización: http://localhost/login.php

A2 – OWASP Juice Shop

Se utiliza la imagen de Docker bkimminich/juice-shop, utilizando los siguientes comandos desde el terminal:

- `docker pull bkimminich/juice-shop`: realiza la instalación de la imagen de la aplicación.
- `docker network create zapnet`: Antes de generar el contenedor con la instanciación de la imagen instalada, se realiza un paso previo que consiste en la creación de una red en Docker para así incorporar la aplicación a esta red y así poder usar convenientemente la herramienta OWASP ZAP en los casos de prueba.
- `docker run --rm -p 3000:3000 --net zapnet bkimminich/juice-shop`: crea el contenedor Docker en el puerto 3000
- Una vez finalizada la configuración, se accede a la aplicación web: http://localhost:3000

A3 – Metasploitable

Se utiliza la imagen de Docker Hub tleemcjr/metasploitable2, utilizando los siguientes comandos desde el terminal:

- `docker pull tleemcjr/metasploitable2`: realiza la instalación de la imagen de la aplicación.
- `docker run -it tleemcjr/metasploitable2`: genera un contenedor docker con la instancia de la imagen de metasploitable2.
- Tras finalizar la instanciación, se dispone del terminal de la máquina metasploitable2 para acceder a ella.

H1 – Kali Linux

Se utiliza la imagen de Docker Hub kalilinux/kali-rolling, utilizando los siguientes comandos desde el terminal:

- `docker pull kalilinux/kali-rolling`: permite realizar la instalación de la imagen oficial que proporciona Kali Linux y que actualizan semanalmente.
- `docker run -ti kalilinux/kali-rolling /bin/bash`: genera el contenedor docker con la instanciación de la imagen de Kali, abriendo un terminal bash tras la finalización.
- En el terminal se procede a ejecutar los siguientes comandos [35](Airman, 2020) para actualizar e instalar el paquete kali-tools-top10
    - apt update.
    - apt dist-upgrade.
    - apt autoremove.
    - apt clean.
    - apt install kali-tools-top10.
- Una vez finalizada la instalación del paquete, se procede a generar una nueva imagen que sea copia del contenedor que contiene la versión de Kali Linux sobre la que hemos realizado la instalación del paquete Kali tolos top10:
    - Ejecutar el comando docker ps para conocer el id del contenedor de Kali.
    - Ejecutar el commando docker commit <ID Contenedor> mi-kali.
    - De esta manera, a partir de ahora sólo se trabaja con la imagen mi-kali que ya contiene la configuración necesaria.
- Ahora se arranca el contenedor con la instanciación de mi-kali:
    - `docker run -ti --rm --mount src=kali-root,dst=/root --mount src=kali-postgres,dst=/var/lib/postgresql mi-kali`
- Finalmente se dispone del terminal para la ejecución de Kali.

H2 – OWASP ZAP

- `docker pull owasp/zap2docker-stable`: realiza la instalación de la imagen zap2docker-stable que es la versión de la herramienta que mantiene actualizada la compañía OWASP para su uso desde Docker.
- `docker run -v $(pwd):/zap/wrk/:rw -u zap -p 8080:8080 -p 8090:8090 --net zapnet -i owasp/zap2docker-stable zap-webswing.sh`: permite generar el contenedor con la instanciación de la imagen de la

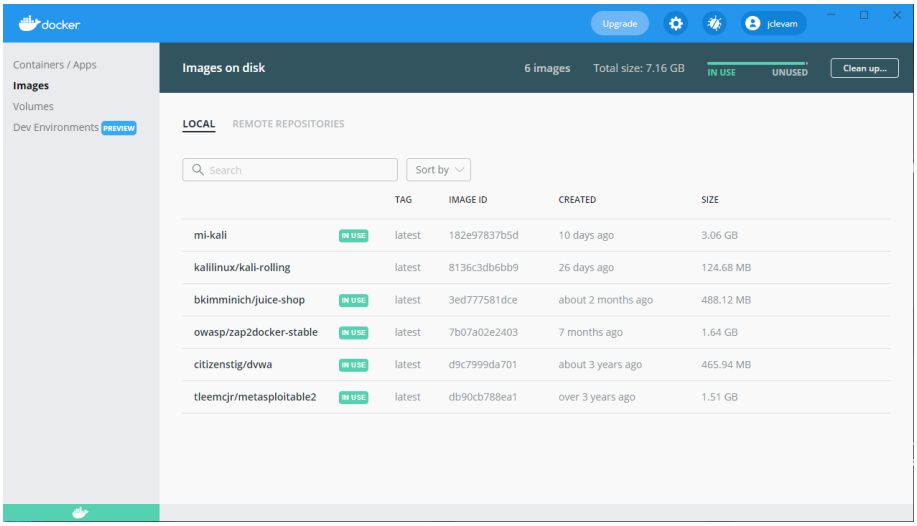
herramienta owasp zap teniendo en cuenta que [36] (*Run ZAP without Java Using Docker and Webswing*, 2021):

- se utiliza zap-webswing.sh para poder usar la interfaz gráfica de la herramienta.
- se incorpora la opción -v $(pwd):/zap/wrk/:rw para que se generen los certificados necesarios para poder hacer uso de la funcionalidad de escaneo manual de una web con la herramienta OWASP zap.

- Finalmente, ya se puede acceder a la herramienta mediante la url: http://localhost:8080/zap/

Una vez finalizada la instalación, configuración y puesta en marcha de las herramientas y aplicaciones que conforman el entorno de seguridad informática, se puede analizar desde Docker Desktop las imágenes instaladas y su tamaño.

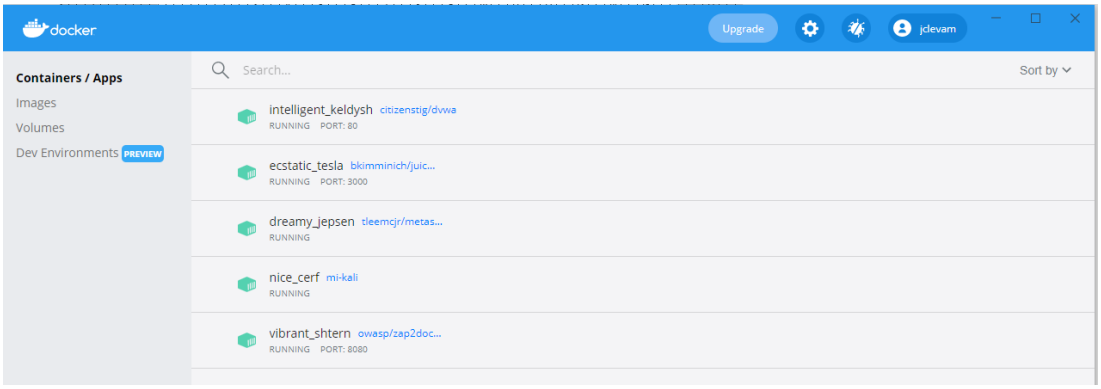
Figura 18. *Docker Desktop con las imágenes de herramientas y aplicaciones instaladas en el entorno de seguridad informática*



Se aprecia que la instalación de todos los componentes ha requerido de 7,16 GB que es un tamaño más que razonable para permitir su uso en casi la totalidad de portátiles u ordenadores de sobremesa de la actualidad que pueda disponer una persona que desee utilizar este entorno para aprendizaje.

Finalmente, y también desde Docker Desktop, se dispone de la opción de controlar la parada, arranque y acceso a la web o al terminal de cada una de las herramientas y aplicaciones que se han instalado:

Figura 19. *Docker Desktop con contenedores de herramientas y aplicaciones instaladas en el entorno de seguridad informática*



6. Evaluación

Una vez finalizado el diseño e implementación del entorno de seguridad de la información, se procede a evaluar su correcto funcionamiento. Para ello, se han ejecutado cada uno de los casos de uso docentes que se han definido, verificando en su resolución que con el entorno implementado:

- No ha sido necesario invertir tiempo en tareas de configuración de herramientas o del entorno de trabajo para hacer las explotaciones de seguridad.
- No ha sido necesario utilizar terceras herramientas no disponibles en el laboratorio.
- Los tiempos de respuesta al ejecutar las acciones necesarias para la explotación de seguridad son buenas no detectándose lentitud en ninguna de ellas.

A continuación, se detalla el proceso de preparación del entorno de seguridad y los pasos dados en detalle para la resolución de cada reto dado que se considera otro elemento fundamental de aporte de valor para los usuarios que quieren aprender de los retos el que puedan disponer de las soluciones.

En el Anexo E se han incorporado las capturas de imágenes de Docker Desktop para mostrar los contenedores que se arrancan en la preparación del entorno de seguridad en cada reto y que sirvan, así como guía y referencia.

Finalmente, para la resolución de los retos que se ejecutan en la aplicación OWASP Juice Shop se ha utilizado como referencia la documentación oficial detallada sobre dichos retos que proporciona OWASP (*Challenge solutions · Pwning OWASP Juice Shop*, s. f.)

6.1. Reto vulnerabilidad inyección SQL en aplicación DVWA

6.1.1. Preparación del entorno de seguridad informático

En primer lugar, se procede a arrancar los elementos del entorno de seguridad que son necesarios para la ejecución de este reto:

- Aplicación DVWA:
  - `docker run -d -p 80:80 citizenstig/dvwa`
- Herramienta Kali Linux:
  - `docker run -ti --rm --mount src=kali-root,dst=/root --mount src=kali-postgres,dst=/var/lib/postgresql mi-kali`

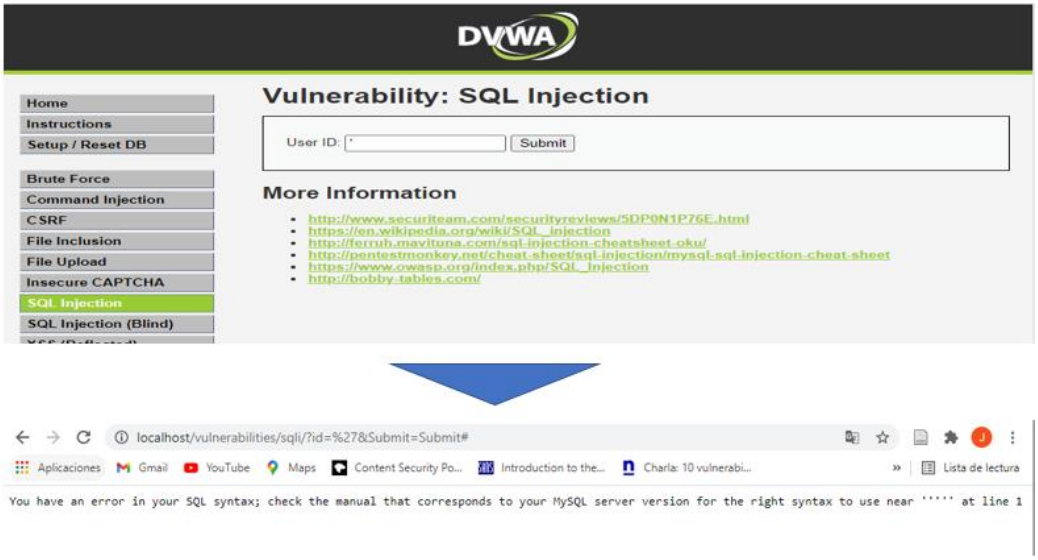
Para la ejecución del reto, se deben realizan las siguientes configuraciones:

- Primero, se accede a la aplicación DVWA: http://localhost/login.php y se utiliza el usuario admin y la contraseña password para el proceso de autenticación.
- A continuación, se accede a la sección de setup: http://localhost/setup.php y se configura el nivel de seguridad a low.
- Una vez finalizada la configuración, para la resolución de reto se utilizará la sección SQL Injection: http://localhost/vulnerabilities/sqli/
- No es necesario ningún elemento adicional ni configuración especial dado que solo es necesaria la aplicación dvwa y la herramienta sqlmap que se encuentra ya instalada para su uso en el contenedor Docker de Kali Linux.
- Para el uso de sqlmap, es necesario conocer la IP de acceso a la aplicación DVWA desde fuera del contenedor en el que se ha desplegado. Para ello se ejecutan los siguientes comandos de Docker:
    - `Docker ps` para obtener el id del contenedor con la aplicación DVWA.
    - `Docker inspect "id del contenedor"` en donde se puede ver que la IP asignada para la aplicación y que debemos usar en el reto es 172.17.0.2

6.1.2. Resolución del reto

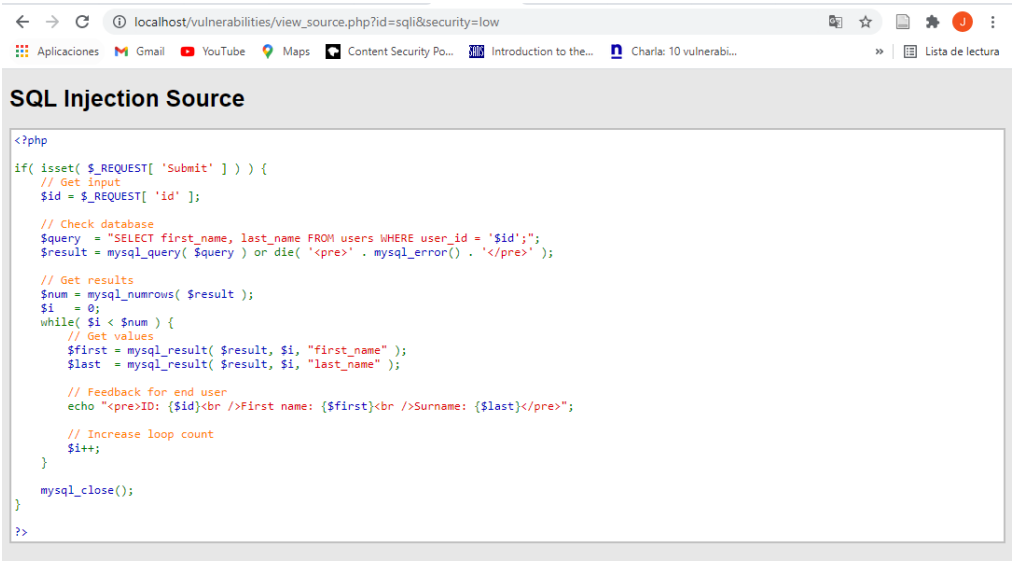
1. En primer lugar se accede a la página web de la aplicación DVWA desde la que se ejecuta el reto: http://localhost/vulnerabilities/sqli/ y se introduce en el campo del formulario User Id el valor ‘ para intentar forzar un error en la consulta SQL y así poder conocer información al respecto de la aplicación de base de datos que se está utilizando.

Figura 20. *Reto vulnerabilidad inyección SQL en aplicación DVWA - Forzar error en consulta SQL*



2. Se accede al código fuente de la página para ver el tipo de consulta SQL que se está construyendo, pudiendo apreciar que no está bien programada al no estar realizándose un saneamiento de los datos que introduce el usuario en el formulario y es lo que nos permite hacer el ataque por SQL Injection.

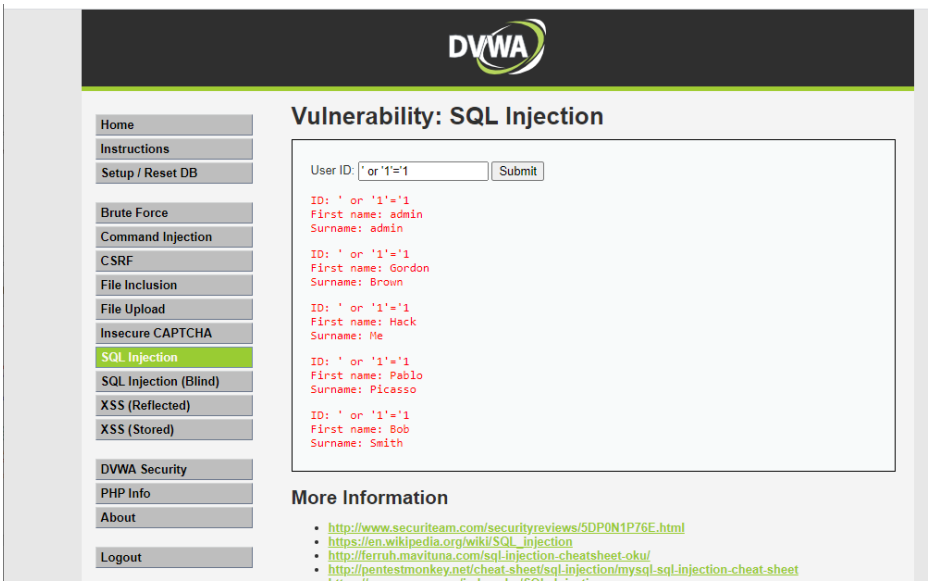
Figura 21. *Reto vulnerabilidad inyección SQL en aplicación DVWA - Código fuente con la consulta SQL*



3. Ahora que ya se conoce la consulta que se construye a partir del dato User Id que se introduce en el formulario, se procede a diseñar el valor que se debe introducir en User

Id para forzar que las condición de la cláusula WHERE se cumpla siempre y así se obtenga el listado completo de todos los usuarios, lo cual ya es una explotación de vulnerabilidad en la se obtienen datos sensibles. Para ello basta con escribir en el campo del formulario User ID el valor: 1' OR '1'='1

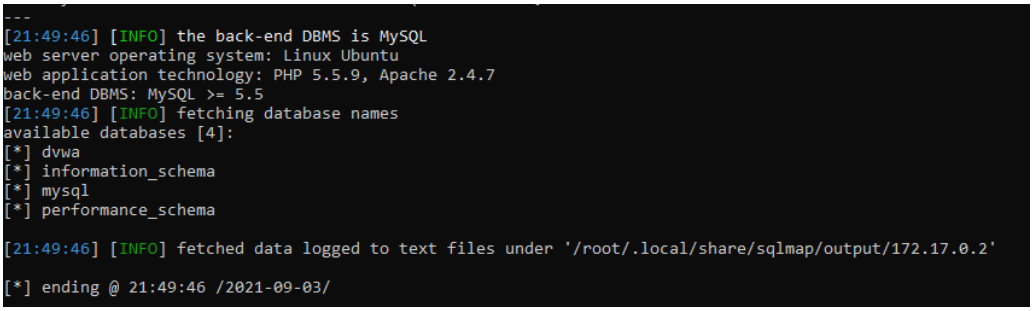
Figura 22. *Reto vulnerabilidad inyección SQL en aplicación DVWA - ataque consulta SQL*



4. Se procede a utilizar la herramienta sqlmap para seguir con el proceso de explotación y listar el nombre de todas las bases de datos que tiene el sistema. Para ello es necesario incorporar cookies con el id de sesión del usuario en la ejecución del comando sqlmap y que se procede a obtener usando un ataque ataque XSS reflected desde la aplicación DVWA ejecutando el siguiente código javascrip: `<script>alert(document.cookie)</script>` para acabar obteniendo la cookie de la sesión: PHPSESSID=1t8o0gu1p1mj07dcmj9h1n45p5; security=low
5. Se ejectua el comando sqlmap con la opción -dbs para listar el nombre de todas las bases de datos que tiene:

```
sqlmap -u "http://172.17.0.2/vulnerabilities/sqli/?id=1&Submit=Submit#" --dbs
--cookie="security=low; PHPSESSID=1t8o0gu1p1mj07dcmj9h1n45p5"
```

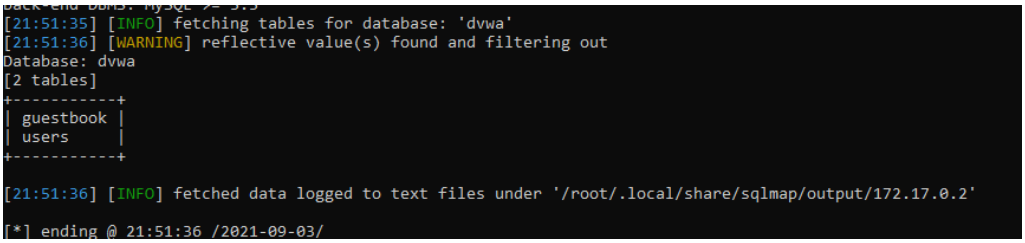
Figura 23. *Reto vulnerabilidad inyección SQL en aplicación DVWA - Listado de bases de datos de la aplicación DVWA*



6. Ahora se procede a seguir avanzando en la explotación para así obtener el nombre de las tablas de la base de datos dvwa de la siguiente manera:

```
sqlmap -u "http://172.17.0.2/vulnerabilities/sqli/?id=1&Submit=Submit#" -D dvwa --tables --cookie="security=low; PHPSESSID=1t8o0gu1p1mj07dcmj9h1n45p5"
```

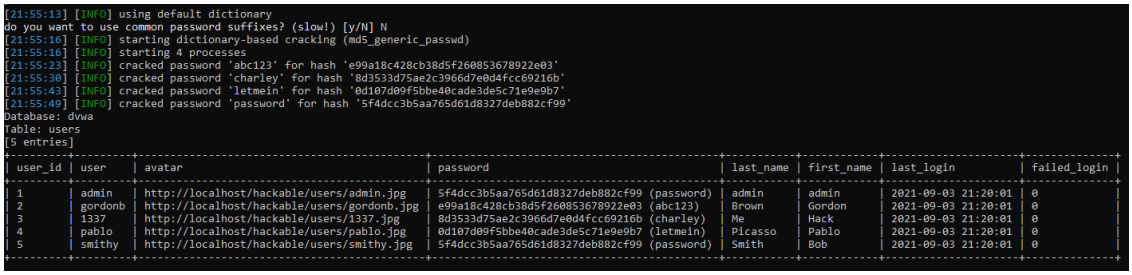
Figura 24. *Reto vulnerabilidad inyección SQL en aplicación DVWA - tablas de la base de datos dvwa*



7. Finalmente para obtener el contenido de las tablas de la bbdd dvwa, se utiliza el comando sqlmap las opciones -D dvwa -T users --dump, haciendo uso de un diccionario de contraseñas para realizar el ataque y sacar en claro sus contraseñas. Se obtiene como resultado todos los usuarios y contraseñas en claro, finalizando así el reto.

```
sqlmap -u "http://172.17.0.2/vulnerabilities/sqli/?id=1&Submit=Submit#" -D dvwa -T users --dump --cookie="security=low; PHPSESSID=1t8o0gu1p1mj07dcmj9h1n45p5"
```

Figura 25. *Reto vulnerabilidad inyección SQL en aplicación DVWA - Listado de usuarios y contraseñas en claro*



6.2. Reto vulnerabilidad inyección SQL en aplicación OWASP Juice Shop

6.2.1. Preparación del entorno de seguridad informático

En primer lugar, se procede a arrancar los elementos del entorno de seguridad que son necesarios para la ejecución de este reto:

- Aplicación The Juice Shop:
  - `docker run --rm -p 3000:3000 --net zapnet bkimminich/juice-shop`
- Herramienta Kali Linux:
  - `docker run -ti --rm --mount src=kali-root,dst=/root --mount src=kali-postgres,dst=/var/lib/postgresql mi-kali`
- Herramienta OWASP ZAP:
  - `docker run -v /mnt/c/Users/jclev/:/zap/wrk/:rw -u zap -p 8080:8080 -p 8090:8090 --net zapnet -i owasp/zap2docker-stable zap-webswing.sh`

Para el uso de OWASP ZAP, es necesario conocer la IP de acceso a la aplicación The Juice Shop y de OWASP ZAP desde fuera del contenedor en el que se han desplegado. Para ello se ejecutan los siguientes comandos de Docker:

- Docker ps para obtener el id del contenedor con la aplicación The Juice Shop y OWASP ZAP.
- Docker inspect “id del contenedor” en donde se puede ver que la IP asignada para la aplicación 172.18.0.2 y para la herramienta OWASP ZAP 172.18.0.3.

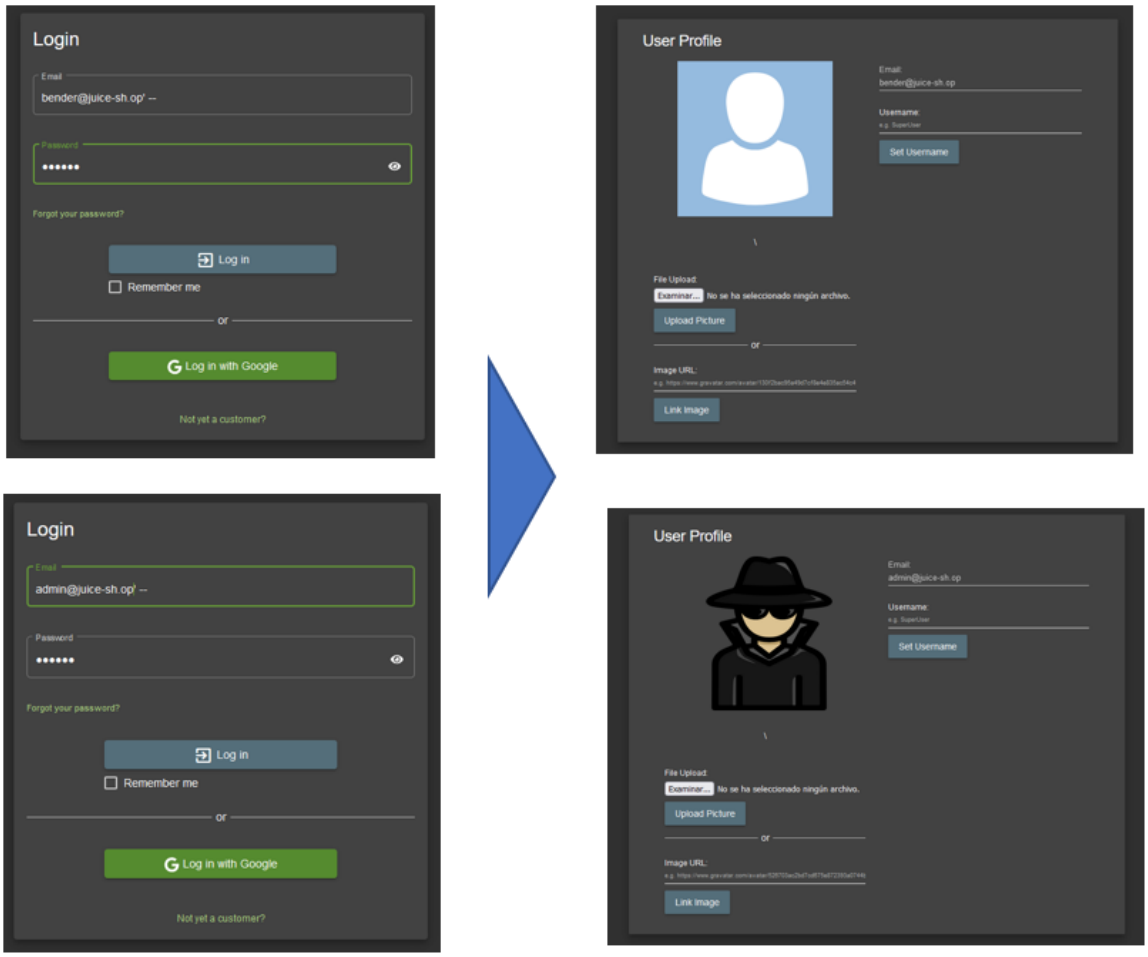
6.2.2. Resolución del reto

1. El primer paso consiste en acceder a la página de la aplicación OWASP Juice Shop: http://localhost:3000/#/
2. Ahora se procede a usar la funcionalidad de Login con OWASP ZAP activado en modo manual y habilitado el modo HUD para poder hacer la traza de los mensajes en la comunicación.
3. Se procede a forzar un error en la consulta SQL para poder analizar el resultado. Para ello se introduce en el campo user ‘ y en password cualquier valor. Tras ejecutarlo paso a paso con OWASP ZAP, se obtiene el error y también la consulta SQL que se ha lanzado

Figura 26. *Reto vulnerabilidad inyección SQL en aplicación Juice Shop - código consulta SQL*

4. Ahora que se conoce la consulta SQL, se procede a diseñar el ataque que en este caso bastaría con introducir un email de un usuario e incorporar un comentario para dejar de procesar el resto de la consulta. Así, para logarse con el usuario administrador, se incorpora en email el valor: admin@juice-sh.op’ -- y en password es indiferente el valor que se introduzca.
5. Esta misma operación se podría realizar también con cualquier usuario, por ejemplo, bender@juice-sh.op, utilizando en el campo email nuevamente el valor: bender@juice-sh.op’ --.
6. De esta manera, como resultado obtenido, ha sido posible logarse con ambos usuarios sin necesidad de conocer su usuario mediante el ataque SQL Injection.

Figura 27. *Reto vulnerabilidad inyección SQL en aplicación Juice Shop - Resolución del reto*



6.3. Caso de uso de la vulnerabilidad web de pérdida de autenticación

6.3.1. Preparación del entorno de seguridad informático

En primer lugar, se procede a arrancar los elementos del entorno de seguridad que son necesarios para la ejecución de este reto:

- Aplicación The Juice Shop:
  - `docker run --rm -p 3000:3000 --net zapnet bkimminich/juice-shop`
- Herramienta Kali Linux:
  - `docker run -ti --rm --mount src=kali-root,dst=/root --mount src=kali-postgres,dst=/var/lib/postgresql mi-kali`
- Herramienta OWASP ZAP

- `docker run -v /mnt/c/Users/jclev/:/zap/wrk/:rw -u zap -p 8080:8080 -p 8090:8090 --net zapnet -i owasp/zap2docker-stable zap-webswing.sh`

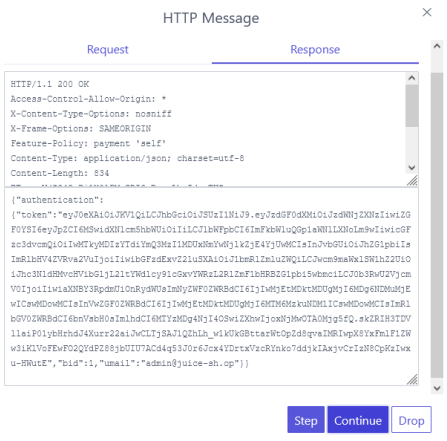
Para el uso de OWASP ZAP, es necesario conocer la IP de acceso a la aplicación The Juice Shop y de OWASP ZAP desde fuera del contenedor en el que se han desplegado. Para ello se ejecutan los siguientes comandos de Docker:

- Docker ps para obtener el id del contenedor con la aplicación The Juice Shop y OWASP ZAP.
- Docker inspect "id del contenedor" en donde se puede ver que la IP asignada para la aplicación 172.18.0.2 y para la herramienta OWASP ZAP 172.18.0.3.

6.3.2. Resolución del reto

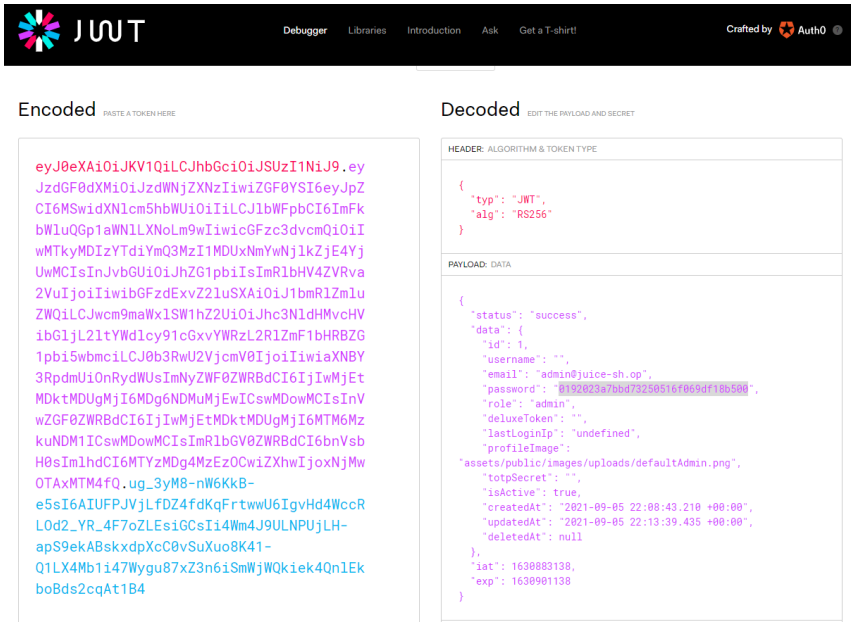
1. El primer paso consiste en acceder a la página de la aplicación OWASP Juice Shop: http://localhost:3000/#/
2. Ahora se procede a usar la funcionalidad de Login con OWASP ZAP activado en modo manual y habilitado el modo HUD para poder hacer la traza de los mensajes en la comunicación.
3. A continuación, se realiza un ataque SQL como el realizado en el caso de uso docente inyección SQL para así autenticarse con el usuario administrador: admin@juice-sh.op'– y se usa OWASP ZAP para traza el proceso, obteniendo así el valor del token que se le asigna a la sesión del usuario administrador.

Figura 28. *Reto pérdida de autenticación - Obtención del token sesión administrador*



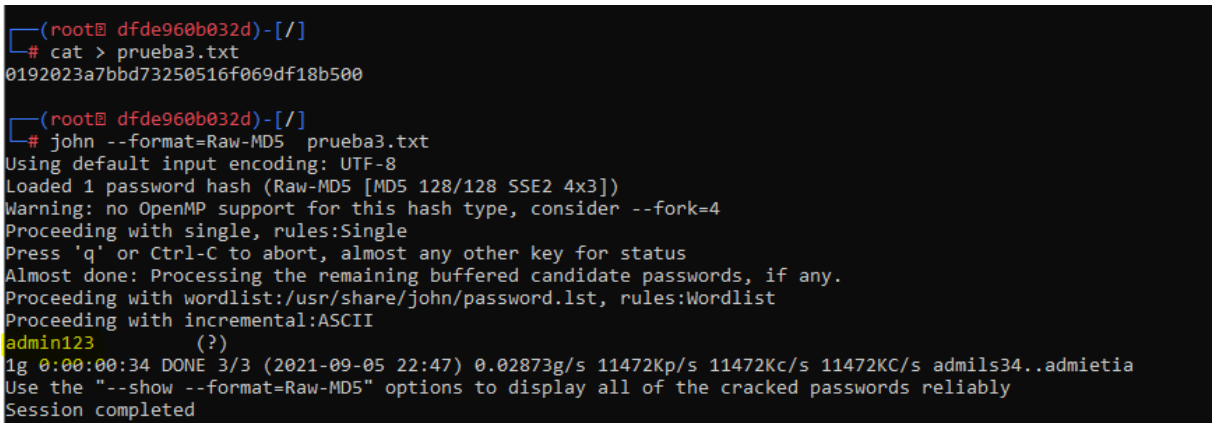
4. Para leer el token, se utiliza la web https://jwt.io/ y así se obtiene el valor del hash del password del usuario administrador.

Figura 29. *Reto pérdida de autenticación - hash contraseña del administrador*



5. Finalmente, se usa la herramienta John the Ripper para aplicar fuerza bruta sobre el hash, obteniendo el valor de la contraseña y que es admin123

Figura 30. *Reto pérdida de autenticación - contraseña administrador con fuerza bruta*



6.4. Caso de uso de la vulnerabilidad web de exposición de datos sensibles

6.4.1. Preparación del entorno de seguridad informático

En primer lugar, se procede a arrancar los elementos del entorno de seguridad que son necesarios para la ejecución de este reto:

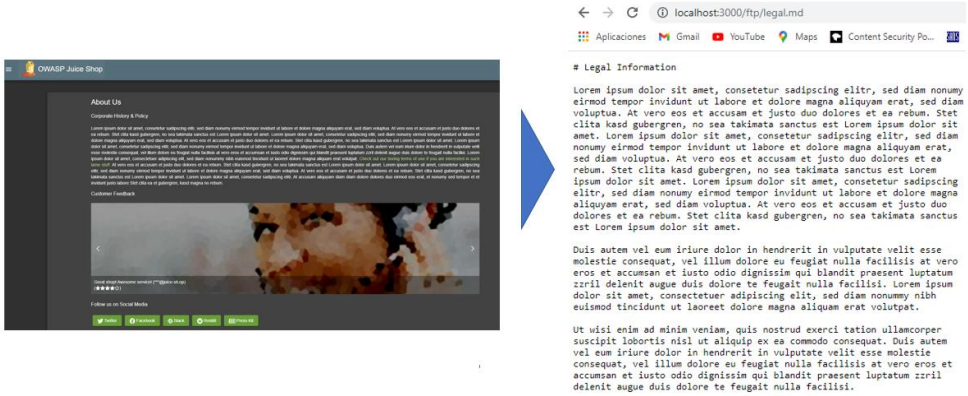
- Aplicación The Juice Shop:
    - `docker run --rm -p 3000:3000 --net zapnet bkimminich/juice-shop`

Para la ejecución de este reto no se necesita ninguna configuración adicional.

6.4.2. Resolución del reto

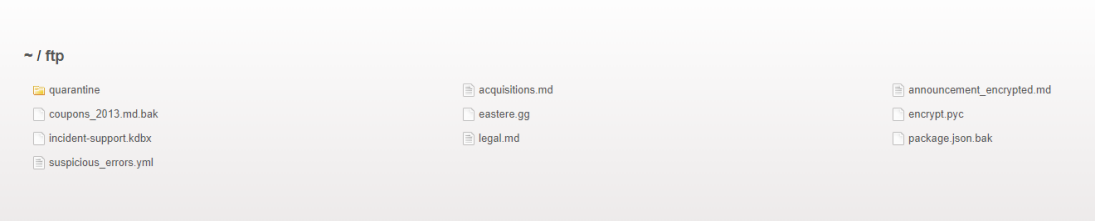
1. El primer paso consiste en acceder a la página de la aplicación OWASP Juice Shop: http://localhost:3000/#/ y navegar a la sección About Us
2. En esta sección se aprecia un link que en la url muestra la ruta /ftp/legal.md

Figura 31. *Reto pérdida de autenticación - Descubrir carpeta ftp*



3. A partir de esta información obtenida se prueba a intentar acceder a la ruta http://localhost:3000/ftp/ y se aprecia que hay una vulnerabilidad de acceso a datos sensibles dado que se pueden ver ficheros para los que no se deberían ni exponer su nombre.

Figura 32. *Reto pérdida de autenticación - acceso y contenido carpeta ftp*

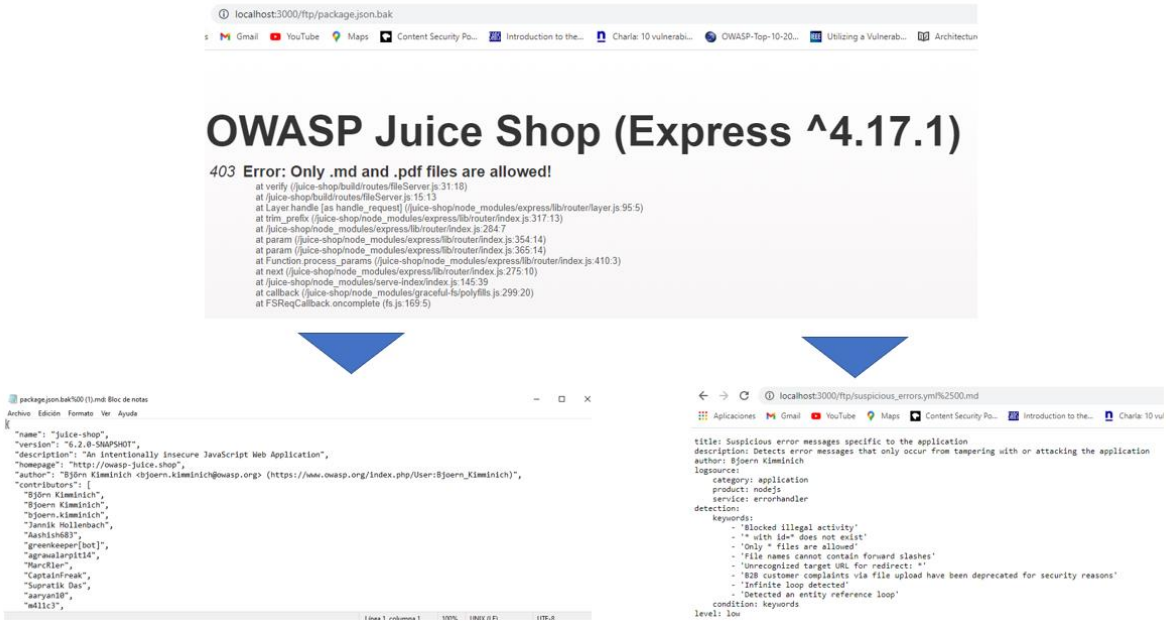


4. Utilizando el ataque *Poison Null Byte*, incorporamos a la url para la descarga de los ficheros mencionados en el reto el valor %2500 de la siguiente manera, pudiendo descargar ambos ficheros y finalizando así el reto:

a. http://localhost:3000/ftp/suspicious_errors.yml%2500.md
b. http://localhost:3000/ftp/package.json.bak%00.md

5. El ataque *Poison Null Byte* aprovecha las cadenas con una longitud conocida que puede contener bytes nulos, y si la API atacada utiliza cadenas terminadas en nulo o no. Al colocar un byte NULL en la cadena en un byte determinado, la cadena terminará en ese punto, anulando el resto de la cadena, como una extensión de archivo.

Figura 33. *Reto pérdida de autenticación - Descarga de ficheros con información sensible*



6.5. Caso de uso de la vulnerabilidad web de entidades externas XML (XXE)

6.5.1. Preparación del entorno de seguridad informático

En primer lugar, se procede a arrancar los elementos del entorno de seguridad que son necesarios para la ejecución de este reto:

- Aplicación The Juice Shop:
  - `docker run --rm -p 3000:3000 --net zapnet bkimminich/juice-shop`

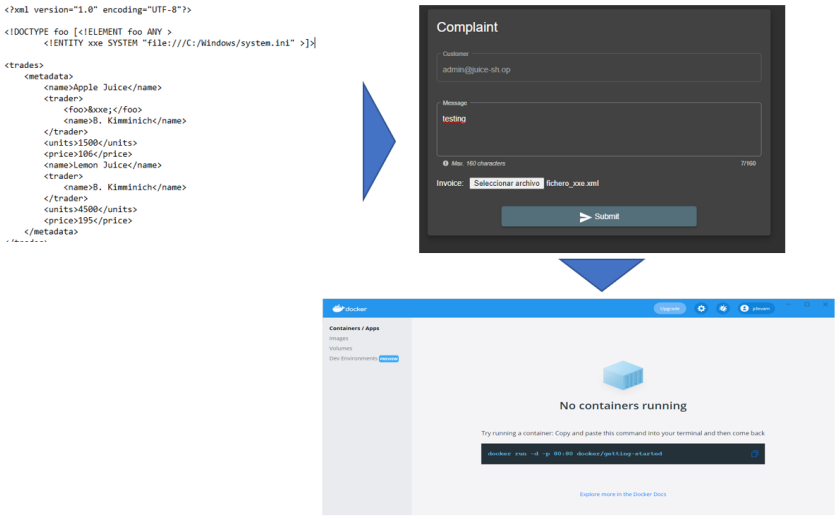
Para la ejecución del reto no se necesita ninguna configuración adicional por lo que simplemente se comprueba desde Docker Desktop que está arrancada correctamente la aplicación:

6.5.2. Resolución del reto

1. El primer paso consiste en acceder a la página de la aplicación OWASP Juice Shop: http://localhost:3000/#/ y logarse con un usuario como por ejemplo el administrador: admin@juice-sh.op - admin123
2. Se procede a construir un fichero xml con la entidad externa para acceder a System.ini:

   <!ENTITY xxe SYSTEM "file:///C:/Windows/system.ini" >]>
3. Se accede al formulario Complaint de la aplicación y se incorpora el fichero xml generado en el paso anterior

Figura 34. *Reto entidad externa XXE - resultado de la ejecución*



4. El resultado no es el esperado y se obtiene un error que cierra el contenedor.
5. Tras investigar al respecto en la documentación oficial de OWASP Juice Shop (*XML External Entities (XXE) · Pwning OWASP Juice Shop*, s. f.), se detecta que esta vulnerabilidad no puede ser explotada ni desde Docker ni desde *Heroku dyno* dados los problemas de seguridad que podrían ocasionar si no se ejecuta en un entorno perfectamente controlado por lo que no es posible realizar el reto con el entorno.
6. Revisando el resto de documento de OWASP Juice Shop, este sería el único caso en el que no sería viable si bien el contenido teórico y práctico del reto descrito en este documento sigue siendo válido para el aprendizaje del usuario y ejecución en una instalación sin Docker ni *Heroku dyno*.

6.6. Caso de uso de la vulnerabilidad web de pérdida de control de acceso

6.6.1. Preparación del entorno de seguridad informático

En primer lugar, se procede a arrancar los elementos del entorno de seguridad que son necesarios para la ejecución de este reto:

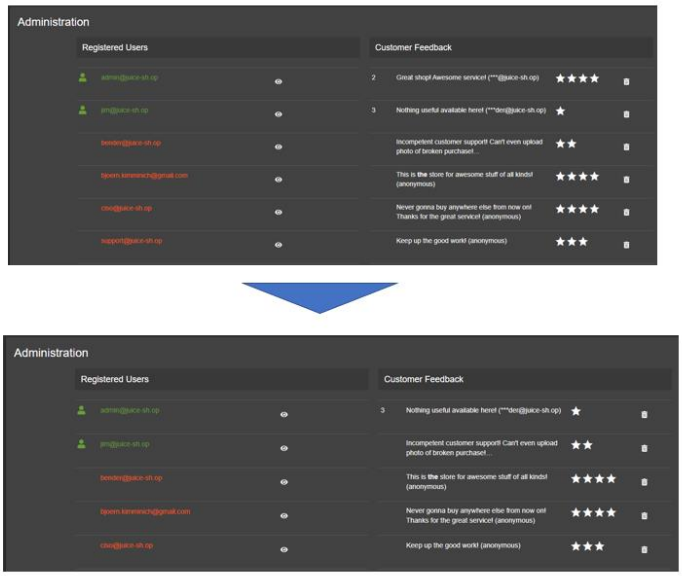
- Aplicación The Juice Shop:
  - `docker run --rm -p 3000:3000 --net zapnet bkimminich/juice-shop`

Para la ejecución del reto no se necesita ninguna configuración adicional.

6.6.2. Resolución del reto

1. El primer paso consiste en acceder a la página de la aplicación OWASP Juice Shop: http://localhost:3000/#/ y se realiza el login con el administrador: admin@juice-sh.op - admin123
2. Se accede a la consola de administración: http://localhost:3000/#/administration desde donde se visualizan los comentarios y valoraciones realizadas por los usuarios.
3. Se intenta borrar alguna entrada de 5 y 4 estrellas obteniendo que la aplicación le permite al usuario realizarlo, lo que significa un problema de control de acceso dado que se debería controlar que esta operación de borrado no se pudiera ejecutar.

Figura 35. *Reto pérdida de control de acceso - resultado borrado valoraciones*

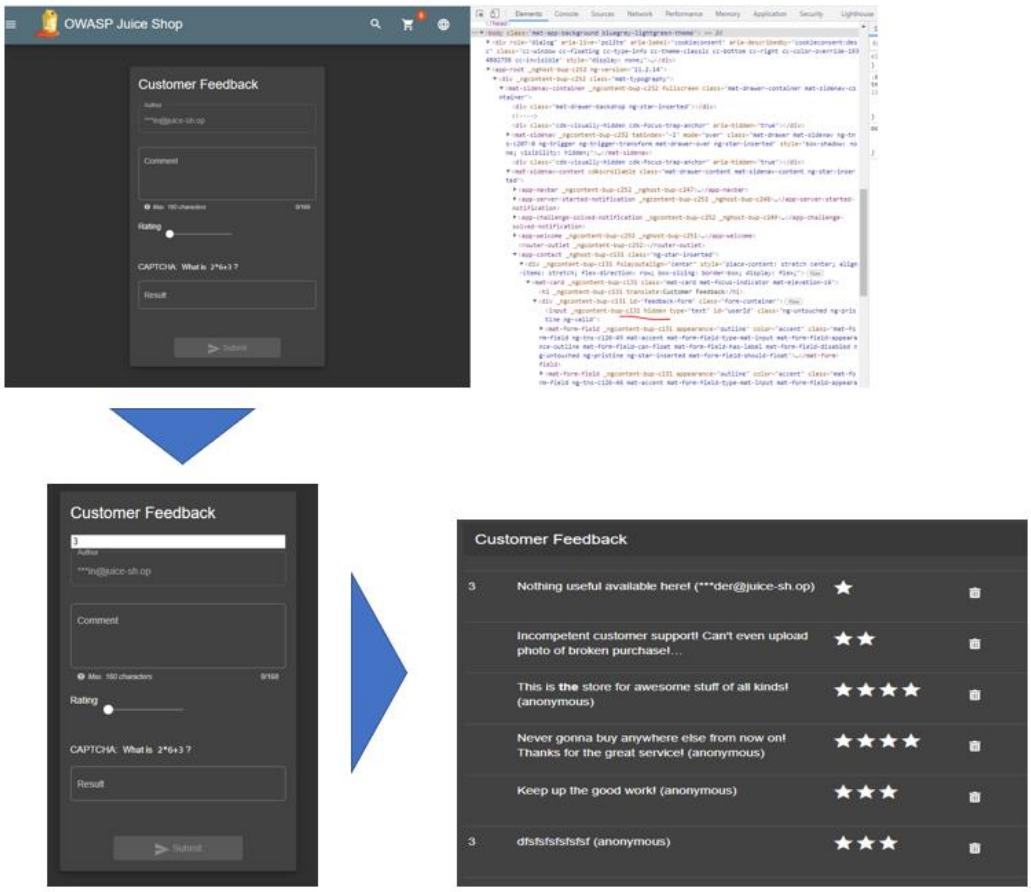


4. El siguiente paso consiste en acceder a la funcionalidad para incorporar feedback en la aplicación: http://localhost:3000/#/contact
5. Se realiza una exploración del formulario en busca de alguna anomalía y se descubre un campo oculto que contiene el valor del user id

```
<input _ngcontent-c23 hidden id="userId" type="text"
class="ng-untouched ng-pristine ng-valid">
```

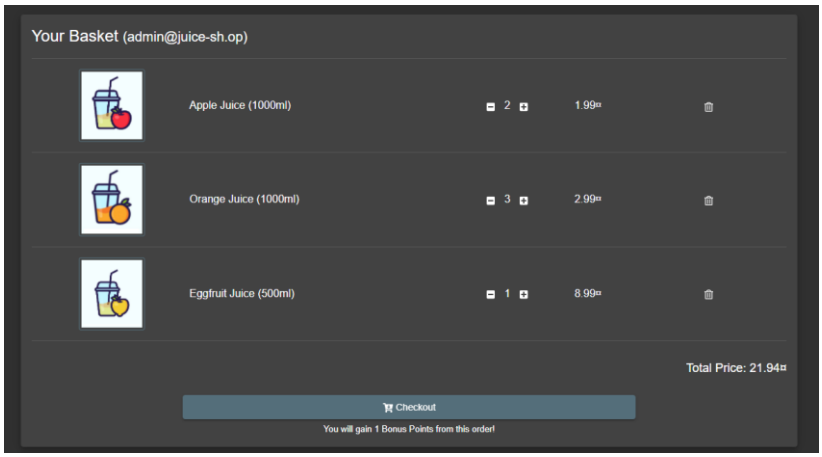
6. Desde el explorador de código del navegador, se elimina la propiedad hidden de este campo user id y se procede a insertar el valor 3 de manera que, tras completar el resto de los campos, el resultado obtenido es que se ha creado una nueva valoración como si se fuera el usuario 3.

Figura 36. *Reto pérdida de control de acceso - Resultado feedback en nombre de otro usuario*



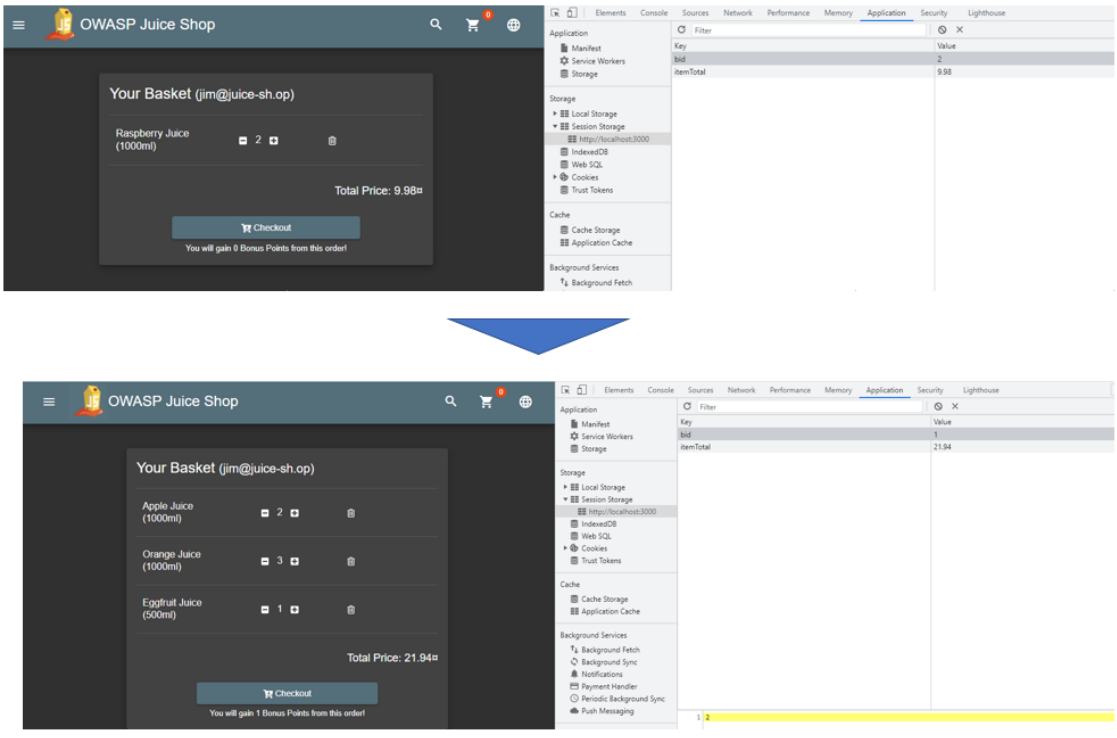
7. Finalmente, se accede a la cesta de la compra del administrador para ver su contenido.

Figura 37. *Reto pérdida de control de acceso - Cesta de la compra del administrador*



8. Se hace logout y ahora se utiliza el usuario jim@juice-sh.op ncc-1701 para volver a acceder a la aplicación OWASP Juice Shop.
9. Se accede a la cesta de la compra y desde el explorador de código se aprecia la variable bid que, si la modificamos a valor 1 y se refresca la página, permite ver desde el usuario de Jim, la cesta de la compra del administrador.

Figura 38. *Reto pérdida de control de acceso - Resultado acceso cesta de la compra de otro usuario*



6.7. Caso de uso de la vulnerabilidad web de secuencia de comandos en sitios cruzados

6.7.1. Preparación del entorno de seguridad informático

En primer lugar, se procede a arrancar los elementos del entorno de seguridad que son necesarios para la ejecución de este reto:

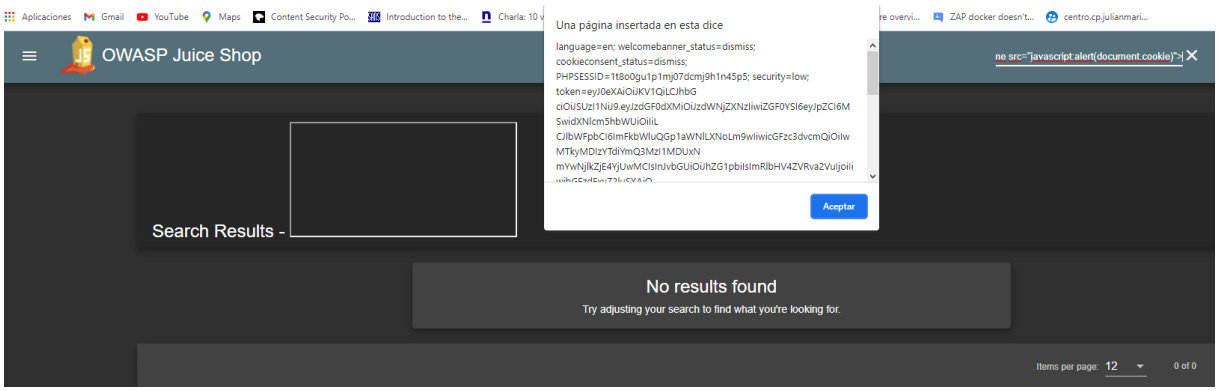
- Aplicación The Juice Shop:
  - `docker run –rm -e "NODE_ENV=unsafe" -p 3000:3000 --net zapnet bkimminich/juice-shop`
  - Es importante reseñar que en este caso se ha incorporado la variable de entorno NODE_ENV para usar la versión insegura de la aplicación (_vavkamil_, 2019) y así poder ejecutar este reto, siendo importante eliminar el contenedor tras su finalización para no generar situaciones inseguras.

Para la ejecución de este reto no se necesita ninguna configuración adicional.

6.7.2. Resolución del reto

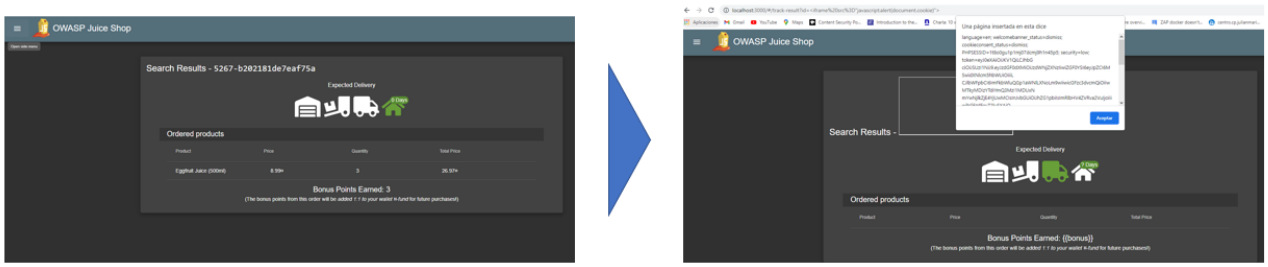
1. El primer paso consiste en acceder a la página de la aplicación OWASP Juice Shop: http://localhost:3000/#/ accediendo con el usuario administrador, aunque sería válido con cualquier usuario.
2. Para el ataque DOM XSS, tan solo basta con incorporar en el buscador el siguiente código javascript: `<iframe src="javascript:alert(document.cookie)">` y al lanzar la búsqueda se obtienen los datos de sesión del usuario.

Figura 39. *Reto Cross Site Scripting - resultado DOM XSS*



3. Para el ataque XSS Reflejado, se accede a la sección de seguimiento de pedidos de la aplicación donde se aprecia que la url de detalle de cualquier pedido es de la siguiente manera: http://localhost:3000/#/track-result?id=xxxxxxxxxxxxx, mostrando el valor xxxxxxx en la pantalla.
4. Encontrado esta puerta de ataques, se usa esta url incorporando como valor al id el siguiente trozo javascript para obtener los datos de sesión del usuario: http://localhost:3000/#/track-result?id=%3Ciframe%20src%3D%22javascript:alert(%60xss%60)%22%3E

Figura 40. *Reto Cross Site Scripting -Resultado XSS reflejado*



6.8. Caso de uso explotación vulnerabilidad postgres máquina metasploitable

cómo atacar una base de datos PostgreSQL, leer y escribir archivos a través de SQL, usar permisos débiles para ejecutar el código en la máquina de destino

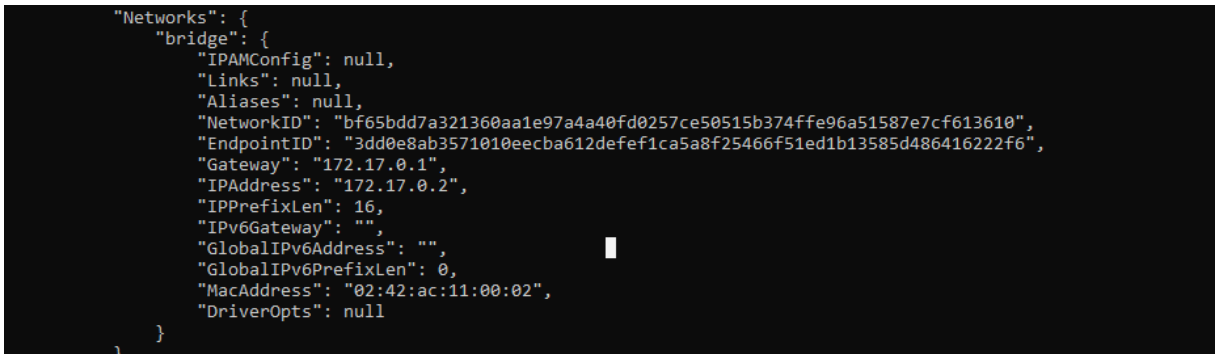
6.8.1. Preparación del entorno de seguridad informático

En primer lugar, se procede a arrancar los elementos del entorno de seguridad que son necesarios para la ejecución de este reto:

- Aplicación Metasploitable:
  - `docker run -it tleemcjr/metasploitable2`
- Herramienta Kali Linux:
  - `docker run -ti --rm --mount src=kali-root,dst=/root --mount src=kali-postgres,dst=/var/lib/postgresql mi-kali`
- No es necesaria ninguna preparación previa adicional de la aplicación o del contenedor de Kali Linux que ya tiene instaladas las herramientas que son necesarias para el reto: nmap y msfconsole.

- Para el uso de nmap y las explotaciones a realizar desde msfconsole, es necesario conocer la IP de acceso a la aplicación metasploitable2 desde fuera del contenedor en el que se ha desplegado. Para ello se ejecutan los siguientes comandos de Docker:
    - Docker ps para obtener el id del contenedor con la aplicación metasploitable2.
    - Docker inspect "id del contenedor" en donde se puede ver que la IP asignada para la aplicación y que debemos usar en el reto es 172.17.0.2

Figura 41. *Reto vulnerabilidad postgres máquina metasploitable - obtener ip de la máquina*

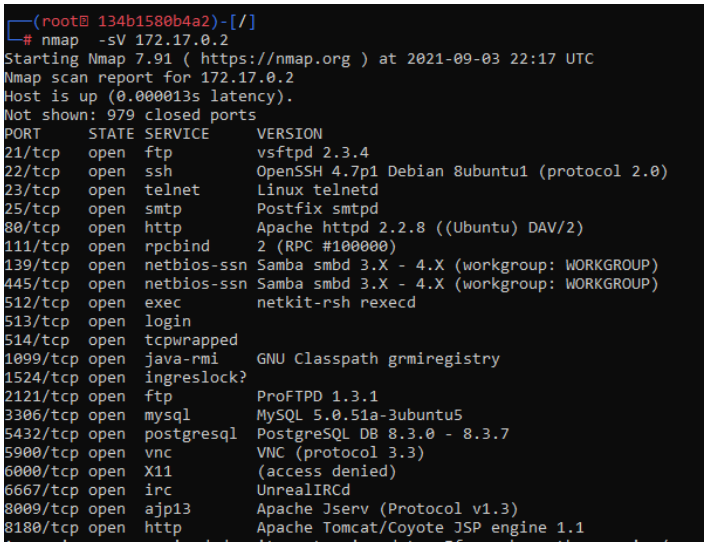


6.8.2. Resolución del reto

1. Se realiza en primer lugar un escaneo de puertos abiertos en la máquina metasploitable2 utilizando el comando nmap desde Kali Linux y activando la opción para conocer la versión de cada herramienta que se detecte en el escaneo:

```
nmap  -sV 172.17.0.2
```

Figura 42. *Reto vulnerabilidad postgres máquina metasploitable - resultado escaneo nmap*



2. Se observa que un servicio es postgresql para el que se va a realizar el ataque de fuerza bruta que nos permita encontrar usuario y password para poder logarnos en la base de datos.
3. Para ello utilizamos msfconsole y usamos el ataque postgres_login [64] (*CVE-1999-0502 : A Unix account has a default, null, blank, or missing password.*, s. f.) de la siguiente manera:
    - `msfconsole`
    - `use auxiliary/scanner/postgres/postgres_login`
    - `set rhosts 172.17.0.2`
    - `run`
4. Como resultado, se ha obtenido que se puede utilizar el usuario y pasword postgres para acceder a la base de datos, por lo que realizamos la comprobación de que efectivamente funciona, finalizando así el reto: psql -h 172.17.0.2 -U postgres

Figura 43. *Reto vulnerabilidad postgres máquina metasploitable - Resultado del reto*

```
   set RHOSTS www.example.test/24
msf6 auxiliary(scanner/postgres/postgres_login) > set RHOSTS 172.17.0.2
RHOSTS => 172.17.0.2
msf6 auxiliary(scanner/postgres/postgres_login) > run

[!] No active DB -- Credential data will not be saved!
[-] 172.17.0.2:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: :tiger@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: :postgres@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: :password@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: :admin@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: postgres:@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: postgres:tiger@template1 (Incorrect: Invalid username or password)
[+] 172.17.0.2:5432 - Login Successful: postgres:postgres@template1
[-] 172.17.0.2:5432 - LOGIN FAILED: scott:@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: scott:tiger@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: scott:postgres@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: scott:password@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: scott:admin@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: admin:@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: admin:tiger@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: admin:postgres@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: admin:password@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: admin:admin@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: admin:admin@template1 (Incorrect: Invalid username or password)
[-] 172.17.0.2:5432 - LOGIN FAILED: admin:password@template1 (Incorrect: Invalid username or password)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/postgres/postgres_login) > exit

┌──(root㉿134b1580b4a2)-[/]
└─# psql -h 172.17.0.2 -U postgres
Password for user postgres:
psql (13.3 (Debian 13.3-1), server 8.3.1)
Type "help" for help.

postgres=#
```

7. Conclusiones y líneas de trabajo futuro

Ante la necesidad creciente de disponer de perfiles profesionales formados en ciberseguridad para hacer frente al número incremental de ciberataques que se están produciendo, se ha identificado el problema y necesidad de disponer de elementos que permitan obtener una curva de aprendizaje efectiva y eficiente.

En la búsqueda de una solución, se han analizado todos los elementos que debe utilizar cualquier persona que quiere formarse ya sea a título individual, docente o empresarial y que son los retos o ejercicios para aprender en ciberseguridad, así como las herramientas y aplicaciones de entrenamientos donde poder ejecutarlos.

Se ha determinado que la curva de aprendizaje de los usuarios depende en gran medida conocimientos técnicos para realizar instalaciones, configuraciones y uso de las aplicaciones y herramientas, así como conocimiento teórico en ciberseguridad para poder hacer frente a los retos:

- No todos los usuarios tienen un perfil técnico que les permita enfrentarse con garantías al proceso de instalación y configuración de herramientas.
- La utilización de herramientas y aplicaciones requieren de unas características de infraestructura para su funcionamiento que en ocasiones los usuarios no disponen en sus ordenadores.
- El aprendizaje basado en retos es el más efectivo para asentar los conocimientos teóricos en ciberseguridad y ser capaz de poder aplicarlos con garantías en un entorno profesional.

Y así, tras analizar estudios similares que hayan abordado esta problemática, se ha determinado el objetivo a desarrollar y cumplir con este trabajo y que es disponer de un entorno de seguridad informática que cumpla las siguientes características: multiplataforma, portable, ligero y preconfigurado con casos de uso docentes, aplicaciones y herramientas esenciales para poder realizar actividades, ejercicios y prácticas relacionadas con los conceptos de ciberseguridad más relevantes y con aplicación docente.

Se ha diseñado y desarrollado este entorno de seguridad utilizando como base la tecnología de contenedores Docker, acotando el alcance al ámbito web dada la relevancia de los

ciberataques que se sufren desde este punto de entrada, identificando, instalando y configurando las aplicaciones de entrenamiento y herramientas esenciales que permite resolver los casos de uso docentes para aprender sobre las vulnerabilidades más relevantes que existen en las aplicaciones web.

Y finalmente, se ha verificado que se ha logrado el objetivo marcado, ejecutando cada caso de uso docente en el entorno de seguridad que se ha desarrollado en este trabajo y llegando a las siguientes conclusiones sobre este entorno:

- Se ha unificado bajo un mismo entorno de seguridad desarrollado con Docker, aplicaciones de entrenamiento, herramientas esenciales de ciberseguridad para su uso con las aplicaciones de entrenamientos u otros fines y casos de uso sobre los aspectos de ciberseguridad más relevantes relacionados con las vulnerabilidades web.
- Este entorno permite a un usuario usarlo autónomamente para otros fines docentes más allá de los casos de uso que se han proporcionado, activando y desactivando de manera autónoma las herramientas o aplicaciones que desee entrenar en cada momento, ejecutando otros retos ajenos a los facilitados.
- Este entorno favorece las aproximaciones iniciales de usuarios con conocimientos teóricos en ciberseguridad para su aprendizaje dado que en el entorno encontrará por dónde empezar al estar formado por herramientas, aplicaciones de entrenamientos y casos de uso esenciales.
- El entorno es exportable y requiere una infraestructura razonable, características que se han obtenido al aplicar la tecnología de contenedores Docker, y que por tanto es muy adecuada para un uso docente o para usuarios que no disponen de infraestructuras dedicadas solo para trabajar con ciberseguridad.
- Permite aplicaciones docentes dado que los alumnos podrían acceder de manera sencilla y poco costosa en recursos a las herramientas que habitualmente se utilizan en la docencia para poder verlas, usarlas y aprenderlas dentro de las propias clases o prácticas de laboratorio.

En relación con las líneas de trabajo futuras, se establecen dos ámbitos de actuación: evolución y aplicación.

- Realizar una evolución incremental del entorno, incorporando más aplicaciones de entrenamiento y herramientas esenciales de entre las identificadas en este trabajo y asociadas a la tipología de retos en los que se aplican. De esta manera irá creciendo progresivamente el entorno dotándolo de mayor contenido docente en el resto de los ámbitos de ciberseguridad.
- Generar un sistema integrado para la gestión y uso de las pistas en los casos de uso docentes de manera que el usuario pueda acceder a ellas durante la ejecución del reto de manera sencilla y directa, dejando registro de estos eventos y, por otra parte, que los administradores puedan generarlas o modificarlas, así como configurar bajo qué circunstancias deben activarse.
- Desarrollar un sistema de gestión de logros y puntuaciones asociados a los usuarios que se enfrentan a los casos de uso para potenciar las bondades del aprendizaje basado en retos. De esta manera, el usuario dispondrá de un perfil al que asociar y almacenar estos logros, permitiendo impulsar potenciadores para que se enfrente a más retos: tabla de puntos globales de todos los usuarios, premios asociados al tiempo de ejecución de los retos, número de retos completados, etc.
- Evolucionar las características del entorno de seguridad para continuar reduciendo los tiempos que el usuario debe dedicar a configuraciones, mediante la utilización de Dockerfile para hacer transparente al máximo todos los pasos de configuraciones que se han identificado para la instalación y puesta en marcha en el entorno de seguridad de las herramientas y aplicaciones que lo conforman.
- En esta misma línea de evolución, explorar y aplicar la utilización de Dockerfile y Docker Compose en los casos de uso docentes para facilitar y seguir reduciendo los pasos y tiempos de puesta en marcha al ejecutarlos. De esta manera, utilizando estos elementos de Docker se puede empaquetar y preconfigurar los pasos iniciales de arranque y configuración de la información, herramientas y aplicaciones necesarias para la ejecución de los casos de uso.
- Realizar pilotos docentes donde aplicar este entorno de seguridad para la enseñanza de seguridad en aplicaciones web, analizando y obteniendo los resultados de su uso en un entorno docente real para identificar puntos de mejoras y futuras líneas de actuación. El ámbito docente puede ser desde académico a un entorno empresarial.

Referencias bibliográficas

(ISC)².(2018). Cybersecurity Workforce Study. Recuperado 27 de agosto de 2021, de https://www.isc2.org/-/media/7CC1598DE430469195F81017658B15D0.ashx

19 Powerful Penetration Testing Tools Used By Pros in 2021. (2021). Recuperado 27 de agosto de 2021, de https://www.softwaretestinghelp.com/penetration-testing-tools/

Fernández, M (2019). Laboratorio de Pentesting basado en tecnología de contenedores. [Trabajo fin de master, Unican]. Recuperado 27 de agosto de 2021, de https://repositorio.unican.es/xmlui/bitstream/handle/10902/16950/419452.pdf?sequence=1&isAllowed=y

Access Control—OWASP Cheat Sheet Series. (s. f.). Recuperado 3 de septiembre de 2021, de https://cheatsheetseries.owasp.org/cheatsheets/Access_Control_Cheat_Sheet.html

Aguiar, A. R. (2021, marzo 10). *El gran hackeo al SEPE es obra de Ryuk: Cómo ha podido llegar el ciberataque, qué ha fallado y por qué ahora mismo es una de las mayores amenazas de internet*. Business Insider España. https://www.businessinsider.es/ryuk-ransomware-detras-ataque-sepe-ha-fallado-826459

Airman. (2020, junio 11). Kali Linux In a Docker Container. *Medium*. https://airman604.medium.com/kali-linux-in-a-docker-container-5a06311624eb

Barea, F., Romero, I., y Rojo, J. I. (2018.). *Plataforma de gestión de escenarios de ciberseguridad para aprendizaje y entrenamiento*. http://oa.upm.es/55294/1/INVE_MEM_2018_297000.pdf

Challenge solutions · Pwning OWASP Juice Shop. (s. f.). Recuperado 5 de septiembre de 2021, de https://bkimminich.gitbooks.io/pwning-owasp-juice-shop/content/appendix/solutions.html

Content Security Policy (CSP)—HTTP | MDN. (s. f.). Recuperado 3 de septiembre de 2021, de https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP

Control de acceso HTTP (CORS)—HTTP | MDN. (s. f.). Recuperado 3 de septiembre de 2021, de https://developer.mozilla.org/es/docs/Web/HTTP/CORS

Cross Site Scripting Prevention—OWASP Cheat Sheet Series. (s. f.). Recuperado 2 de septiembre de 2021, de https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

Pablos, R. CTF: Entrenamiento en seguridad informática. (2014, febrero 26). INCIBE-CERT. https://www.incibe-cert.es/blog/ctf-entrenamiento-seguridad-informatica

CVE-1999-0502: A Unix account has a default, null, blank, or missing password. (s. f.). Recuperado 5 de septiembre de 2021, de https://www.cvedetails.com/cve/CVE-1999-0502/

Dabrowski, A., Kammerstetter, M., Thamm, E., & Weippl, E. (s. f.). *Leveraging Competitive Gamification for Sustainable Fun and Profit in Security Education*. 8.

Day 18: Essential CTF Tools. We are about to kick off the 2019 CTF… | by Z3R0 | Medium. (2019). Recuperado 27 de agosto de 2021, de https://int0x33.medium.com/day-18-essential-ctf-tools-1f9af1552214

Rodríguez, T. De Docker a Kubernetes: Entendiendo qué son los contenedores y por qué es una de las mayores revoluciones de la industria del desarrollo. (2019). Recuperado 27 de agosto de 2021, de https://www.xataka.com/otros/docker-a-kubernetes-entendiendo-que-contenedores-que-mayores-revoluciones-industria-desarrollo

Docker overview. (2021, agosto 20). Docker Documentation. https://docs.docker.com/get-started/overview/

Emedding Null Code Software Attack | OWASP Foundation. (s. f.). Recuperado 4 de septiembre de 2021, de https://owasp.org/www-community/attacks/Embedding_Null_Code

Centro Criptológico Nacional (2020). Ciberamenazas y tendencias. Recuperado 27 de agosto de 2021, de https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/5377-ccn-cert-ia-13-20-ciberamenazas-y-tendencias-edicion-2020/file.html

Guarino, L. (2017). *Analizando vulnerabilidades con Metasploitable2*. 15.

Herrador, C. (2021, enero 8). El sector salud duplica su cifra ciberataques en España. *IPMARK | Información de valor sobre marketing, publicidad, comunicación y tendencias digitales*. https://ipmark.com/el-sector-salud-duplica-su-cifra-ciberataques-en-espana/

Install Docker Desktop on Windows. (2021, agosto 20). Docker Documentation. https://docs.docker.com/desktop/windows/install/

INTERPOL. (2020). INTERPOL report shows alarming rate of cyberattacks during COVID-19. Recuperado 27 de agosto de 2021, de https://www.interpol.int/News-and-Events/News/2020/INTERPOL-report-shows-alarming-rate-of-cyberattacks-during-COVID-19

Israel. (2020, agosto 5). *Las 8 herramientas imprescindibles de pentesting*. Viewnext. https://www.viewnext.com/8-herramientas-imprescindibles-pentesting/

Johnson, J. (s. f.). *Salting vs Stretching Passwords for Enterprise Security*. BMC Blogs. Recuperado 3 de septiembre de 2021, de https://www.bmc.com/blogs/salting-stretching-passwords/

Juice Shop—Insecure Web Application for Training | OWASP. (s. f.). Recuperado 27 de agosto de 2021, de https://owasp.org/www-project-juice-shop/

Kali Linux. (2021). En *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Kali_Linux&oldid=137920344

Kali Linux Metapackages | Kali Linux Blog. (2014). Kali Linux. Recuperado 27 de agosto de 2021, de https://www.kali.org/blog/kali-linux-metapackages/

Kali Linux Tools Listing. (s. f.). Recuperado 27 de agosto de 2021, de https://tools.kali.org/tools-listing

Karagiannis, S., y Magkos, E. (2020). Adapting CTF challenges into virtual cybersecurity learning environments. *Information and Computer Security*. https://doi.org/10.1108/ICS-04-2019-0050

La campaña de Hillary Clinton, afectada por el ciberataque al Partido Demócrata. (2016). *BBC News Mundo*. Recuperado 27 de agosto de 2021, de https://www.bbc.com/mundo/noticias-internacional-36931608

Ramiro, R. (2018, septiembre 16). *Las mejores herramientas hacking*. CIBERSEGURIDAD .blog. https://ciberseguridad.blog/las-mejores-herramientas-hacking/

Luburić, N., Sladić, G., & Milosavljević, B. (2019). Utilizing a Vulnerable Software Package to Teach Software Security Design Analysis. *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1169-1174. https://doi.org/10.23919/MIPRO.2019.8757149

Martínez, J. E. (2020, junio 4). Utilizar Docker con WSL 2 en Windows 10. *Adictos al trabajo*. https://www.adictosaltrabajo.com/2020/06/04/utilizar-docker-con-wsl-2-en-windows-10/

Montes, L. (2018). *Itinerario de retos para la formación de profesionales. [Trabajo fin de master, UOC]*. http://openaccess.uoc.edu/webapps/o2/bitstream/10609/89527/4/diegolopmonTFM1218memoria.pdf.

Official Kali Linux Docker Images | Kali Linux Documentation. (s. f.). Kali Linux. Recuperado 27 de agosto de 2021, de https://www.kali.org/docs/containers/official-kalilinux-docker-images/

OWASP Foundation | Open Source Foundation for Application Security. (s. f.). Recuperado 27 de agosto de 2021, de https://owasp.org/

OWASP. (2017). OWASP-Top-10-2017-es.pdf. Recuperado 27 de agosto de 2021, de https://wiki.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf

Waisen. J. y Pérez. F.J (2011). Web Vulnerable DVWA. [Trabajo fin de máster, Universidad de Almería]. Recuperado 27 de agosto de 2021, de http://www.adminso.es/recursos/Proyectos/PFM/2011_12/PFM_DVWA.pdf

PricewaterhouseCoopers. (2021). *2021 Global Digital Trust Insights*. PwC. Recuperado 27 de agosto de 2021, de https://www.pwc.es/es/publicaciones/transformacion-digital/global-digital-trust-insights.html

¿Qué es Docker? (s. f.). Recuperado 27 de agosto de 2021, de https://www.redhat.com/es/topics/containers/what-is-docker

Zepeda, E. (2020). ¿Qué es Docker y para qué sirve? Explicación. DEV Community. Recuperado 27 de agosto de 2021, de https://dev.to/neon_affogato/que-es-docker-y-para-que-sirve-explicacion-5h2n

Docker Doc Reference documentation. (2021, agosto 20). Docker Documentation. https://docs.docker.com/reference/

Rodolfo. (2019). *Diez herramientas de pentesting con las que poner a prueba la seguridad de tu empresa*. MuySeguridad. Seguridad informática. Recuperado 27 de agosto de 2021, de https://www.muyseguridad.net/2019/06/05/diez-herramientas-pentatesting/

Run ZAP without Java using Docker and Webswing. (2021, febrero 3). https://www.zaproxy.org/blog/2021-02-03-run-zap-without-java-using-docker-and-webswing/

Singh, A. (2021). *Awesome CTF* [JavaScript]. https://github.com/apsdehal/awesome-ctf (Original work published 2015)

SQL Injection | OWASP. (s. f.). Recuperado 2 de septiembre de 2021, de https://owasp.org/www-community/attacks/SQL_Injection

Poston, H. (2021). The top 5 pentesting tools you will ever need [updated 2021]—Infosec Resources. Recuperado 27 de agosto de 2021, de

https://resources.infosecinstitute.com/topic/the-top-5-pentesting-tools-you-will-ever-need/

Total number of Websites—Internet Live Stats. (s. f.). Recuperado 27 de agosto de 2021, de https://www.internetlivestats.com/total-number-of-websites/#trend

Tutorial Docker: Instalación y primeros pasos—IONOS. (2019). Recuperado 27 de agosto de 2021, de https://www.ionos.es/digitalguide/servidores/configuracion/tutorial-docker-instalacion-y-primeros-pasos/

vavkamil. (2019, octubre 23). *Does XXE challenge works?* [Reddit Post]. r/owasp_juiceshop. www.reddit.com/r/owasp_juiceshop/comments/dlyt09/does_xxe_challenge_works/

Vulnerability categories · Pwning OWASP Juice Shop. (s. f.). Recuperado 27 de agosto de 2021, de https://pwning.owasp-juice.shop/part1/categories.html

Web-vulnerabilities-2020-eng.pdf. (2020). Recuperado 27 de agosto de 2021, de https://www.ptsecurity.com/upload/corporate/ww-en/analytics/web-vulnerabilities-2020-eng.pdf

Porup, J.M. (2020). What is penetration testing? 11 hacking tools the pros use | CSO Online. Recuperado 27 de agosto de 2021, de https://www.csoonline.com/article/2943524/11-penetration-testing-tools-the-pros-use.html

XML External Entities (XXE) · Pwning OWASP Juice Shop. (s. f.). Recuperado 5 de septiembre de 2021, de https://pwning.owasp-juice.shop/part2/xxe.html

XML External Entity Prevention—OWASP Cheat Sheet Series. (s. f.). Recuperado 2 de septiembre de 2021, de

https://cheatsheetseries.owasp.org/cheatsheets/XML_External_Entity_Prevention_Cheat_Sheet.html

Anexo A. Arquitectura, funciones y características de Docker

Docker es la herramienta de contenedores más significativa que existe en la actualidad por lo que se ha procedido a analizar la documentación oficial de Docker (*Install Docker Desktop on Windows*, 2021) (*Docker Overview*, 2021) para conocer su arquitectura, funciones y características, así como diversas fuentes de referencia adicionales para profundizar en su estudio. (Martínez, 2020) (*Tutorial Docker: instalación y primeros pasos - IONOS*, 2019.) (*¿Qué es Docker?*, s. f.) (*Zepeda*, 2020)

Docker es una plataforma abierta que permite el desarrollo y ejecución de aplicaciones separándola de la infraestructura, obteniendo una mayor agilidad en todo el proceso.

Docker permite empaquetar una aplicación en lo que se denomina imagen para su posterior ejecución mediante una instanciación de la imagen en lo que se denomina un contenedor.

De esta manera, el contenedor dispondrá de todo lo que necesite para ejecutar la aplicación, con independencia de lo que esté instalado en el host, además de permitir instanciar tantos contenedores como se puedan requerir en una misma máquina host.

Docker dispone de un set de herramientas y entorno para el desarrollo completo end to end de un contenedor, así como de su administración que facilitan el trabajo tanto a desarrolladores como a los usuarios finalistas de las aplicaciones a través de los contenedores, así como de su administración para arrancarlos, pararlos, configurarlos, eliminarlos, etc.

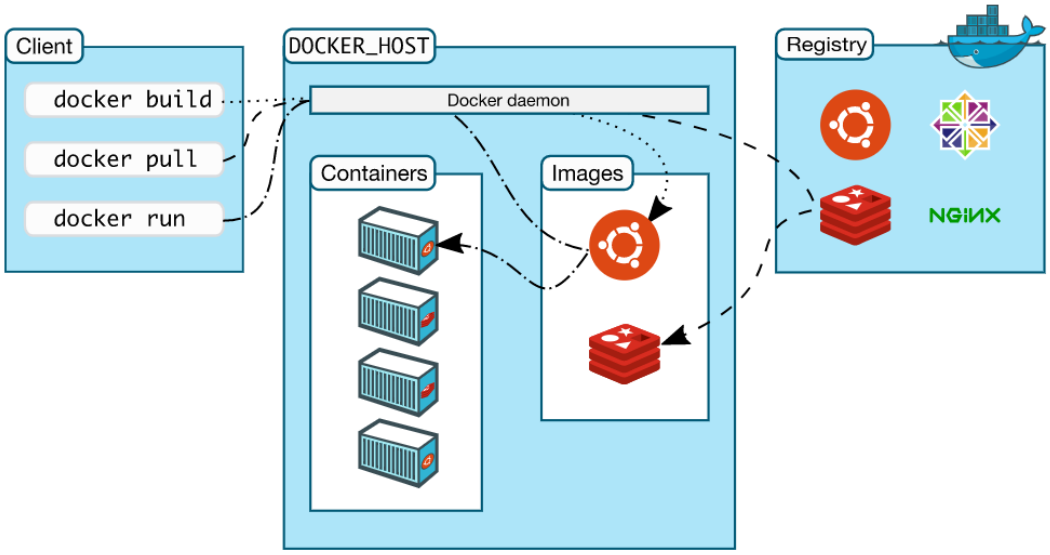
Está disponible tanto para Linux como para Windows y permite que las aplicaciones –imágenes se ejecuten de la misma manera con independencia del sistema operativo en el que se desplieguen a través de Docker mediante los contenedores que se encargan de aislar la aplicación de su entorno y garantizar su funcionamiento.

Docker se basa en una arquitectura cliente-servidor en la que:

- Desde el cliente de Docker se ejecutan los comandos que se trasladan al demonio de Docker para que realice las funciones solicitadas a través de los comandos para la construcción, ejecución, configuración, distribución y mantenimiento de los contenedores.
- La comunicación entre el cliente y el demonio se realiza mediante un API REST, a través de sockets UNIX o una interfaz de red.

- Ambos componentes, cliente y demonio, pueden ejecutarse en el mismo sistema o de manera remota.

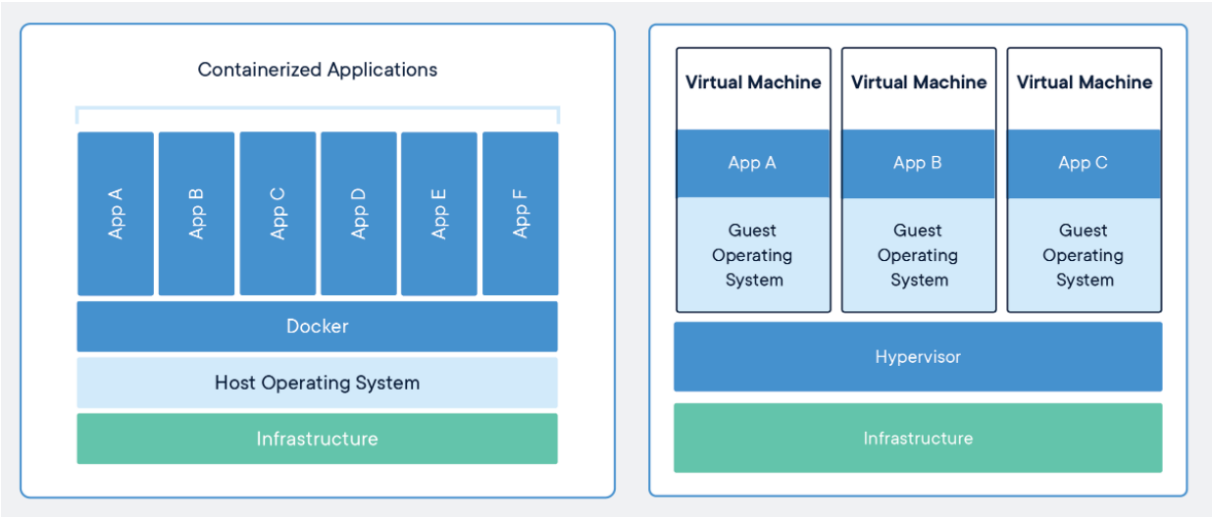
Figura 44. *Arquitectura funcional de Docker*



Fuente: (*Docker Overview*, 2021)

En comparación con las máquinas virtuales, la tecnología de contenedores en general y Docker en particular, permiten obtener beneficios comunes en cuanto a asignación y aislamiento de recursos, pero funcionan de manera diferente:

Figura 45. *Comparativas arquitecturas contenedores y máquinas virtuales*



Fuente: (*Docker Overview*, 2021)

- Los contenedores realizan abstracción en la capa de la aplicación que empaqueta todo lo que necesita junto:

    - Se pueden ejecutar varios contenedores en un mismo sistema, compartiendo el mismo sistema operativo que tenga el sistema.
    - Cada contenedor con la aplicación se ejecuta como un proceso aislado.
    - De esta manera, los contenedores no ocupan más espacio del que necesitan al contener sólo los elementos empaquetados con lo que necesitan para su ejecución.
- Las máquinas virtuales en cambio realizan una abstracción de hardware donde el hipervisor es la pieza fundamental que permite la ejecución de más de una máquina virtual dentro de un mismo sistema, pero:
    - Cada máquina virtual debe tener su sistema operativo completo además de la aplicación y todo lo que necesita para su ejecución.
    - Esto provoca que ocupe mucho más espacio que un contendedor.
    - Además, también provoca que arrancar o trabajar con las máquinas virtuales sea más lento.

Finalmente, reseñamos los elementos de Docker y que utilizaremos para el desarrollo del laboratorio:

- Imágenes Docker: es un fichero de texto, dockerfile, que contiene todas las instrucciones que el Docker Engine requiere ejecutar para crear un contenedor.
- Contenedor Docker: Es una instanciación de una imagen de Docker en la que se preparan todos los sistemas de archivos, código y dependencias con las configuraciones necesarias para que la aplicación funcione correctamente. Se puede considerar por tanto a un contenedor como un proceso en ejecución de una imagen.
- Docker Hub: Se trata de un repositorio online en el que se pone a disposición de los usuarios registrados en Docker de imágenes subidas por otros miembros de la comunidad y que han decidido hacer públicas sus imágenes para su uso por terceros. Existe un gran número de imágenes disponibles en la actualidad tanto de usuarios particulares como también de empresas que se encargan además de realizar un mantenimiento y actualización de las mismas. Existe además una marca de imágenes oficiales que provee Docker y de un sistema para valorar las aplicaciones. Se puede acceder a Docker Hub desde la siguiente url hub.docker.com o desde la aplicación de escritorio Docker Desktop.

Anexo B. Comandos utilizados con Docker

Docker dispone de una serie de comandos descritos en su documentación oficial (Docker Doc *Reference Documentation*, 2021) y que permiten a los usuarios realizar la gestión de las imágenes y contenedores.

En este apartado se describen los comandos y opciones de estos que se utilizan para la implantación y uso de las aplicaciones y herramientas del entorno de seguridad informática.

Antes de comenzar a describir estos comandos es importante reseñar que:

- Los comandos son válidos en cualquier sistema operativo sobre el que se encuentre instalado Docker Desktop.
- Desde Docker Desktop también se pueden realizar las funcionalidades que se describen en este apartado a través de los comandos.
- Se ha optado por trabajar utilizando los comandos dado que la documentación de referencia y ayuda que se encuentra mayoritariamente publicada hace mención a los comandos y no a la interfaz gráfica de Docker Desktop por lo que será así más sencillo a los usuarios del entorno de seguridad informática poder tener el control completo del mismo y buscar cualquier ayuda adicional que pudieran necesitar.
- Para el proceso de instalación y testeo del entorno de seguridad informática se ha utilizado Docker Desktop en su versión para Windows 10 configurado con WSL2 (Windows Subsystem for Linux 2) y conectado con Ubuntu 20.04 LTS por lo que para la ejecución de los comandos de Docker se puede acceder tanto al terminar de Ubuntu como al terminal de Windows.

Una vez establecido el contexto y consideraciones previas relevantes, se procede a describir los comandos más relevantes:

- `docker ps`: muestra el listado de los contenedores en ejecución con información relevante de los mismos como es su identificador, imagen y puertos configurados. Conocer el identificador del contenedor es muy relevante dado que es un atributo que es necesario informar para usar otros comandos.
- `docker inspect <id contentedor>`: permite listar un conjunto completo y detallado de las características y atributos del id del contenedor facilitado. Es muy

recomendable su uso para conocer la IP que permite acceder a una aplicación desde fuera de Docker.

- `docker start, docker stop, docker rm <id contenedor>`: permite parar, arrancar o borrar un id de contenedor específico.
- `docker pull <imagen>`: comando que permite descargar una imagen de un repositorio de docker hub.
- `docker run`: comando que genera un contenedor, es decir, una instancia de una imagen existente en el sistema. Dispone de diferentes parámetros opcionales para este comando entre los que se destacan:
    - `-it`: este parámetro es el que se utiliza habitualmente para aplicaciones que requieren abrir un terminal en el que el usuario puede interactuar y debe ver los resultados para continuar interactuando. La i indica que sea interactivo y la t indica que se abra el TTY.
    - `-d`: ejecuta el contenedor en backgroud y muestra por pantalla el id que se le ha asignado en su creación.
    - `-p <puerto docker>:<puerto app>`: es fundamental su utilización para contenedores de aplicaciones a las que se debe acceder por IP desde fuera del contenedor. Permite configurar el puerto que se usa en el contenedor de docker y su mapeo con un puerto que esté libre del host de docker para así poder acceder también desde fuera del contenedor.
    - `-v`: permite aplicar persistencia de los datos en un contenedor de docker configurando las rutas dentro de un volumen donde almacenará los datos la aplicación, aunque se elimine el contenedor.
    - `-net <nombre network>`: incorpora el contenedor a una red en concreto para que así estén conectados.
    - `-rm`: con este parámetro se indica que se elimine el contenedor automáticamente después de cerrarlo.
- docker network créate <nombre red>: permite crear una red con el nombre específico que se le quiera asignar. Otros comandos relacionados con docker network son:
    - `docker network ls`: lista la redes que actualmente existen.
    - `docker network inspect <nombre red>`: permite conocer los atributos de la red, así como los contenedores que forman parte de ella.

Anexo C. Aspectos docentes adicionales para la resolución de los casos de uso

En este anexo se describen los aspectos docentes adicionales de ciberseguridad que se deben conocer y que están relacionados con los casos de uso docentes diseñados y la vulnerabilidad que se estudia en cada uno de ellos como por ejemplo CORS, CSP, autenticación, etc.

Caso de uso de la vulnerabilidad web de inyección

- Descripción de aspectos docentes adicionales:

  Los retos se han diseñado sobre la vulnerabilidad de inyección de SQL, por lo que se procede a dar más detalle teórico de este caso en particular (*SQL Injection | OWASP*, s. f.).

  En primer lugar, para explotar la vulnerabilidad es importante conocer el tipo de base de datos que se utiliza en la aplicación y para ello lo más sencillo es forzar a generar una consulta errónea utilizando por ejemplo el valor ‘ en el campo de user id en el formulario de login. De esta manera pueden suceder dos escenarios posibles:

  - La aplicación devuelve y muestra por pantalla al usuario el error en la consulta SQL a la base de datos de manera que en la descripción se puede conocer la base de datos.
  - La aplicación web controla el error convenientemente y no muestra información no relevante para el usuario como el tipo de base de datos. Esta tipología se denomina Inyección SQL ciega (*Blind SLS Injection*).

  Otro punto para considerar para explotar la vulnerabilidad es la de conocer el código programado en la aplicación web para generar la consulta SQL para así poder rellenar convenientemente los parámetros del formulario que se usan para hacer la inyección maliciosa. Para ello existen varios caminos que van desde la exploración de código fuente hasta la utilización de herramientas como OWASP zap para analizar la comunicación request y response en donde se puede consultar esta información.

  En relación con su mitigación, es importante considerar:

  - Parametrizar las consultas SQL.

- Validar los datos de entrada, escaparlos y convertirlos siempre el valor a su tipo correspondiente.
- Usar una cuenta con permisos restringidos a la base de datos.
- No mostrar al usuario la información de error generada por la base de datos.
- Rechazar las peticiones con caracteres sospechosos.

Caso de uso de la vulnerabilidad web de pérdida de autenticación

- Descripción de aspectos docentes adicionales:

  El proceso de autenticación tiene por objetivo el de identificar al usuario en la aplicación para así poder controlar el acceso restringido a zonas privadas o datos sensibles por lo que es un aspecto fundamental sobre el que tener un desarrollo y controles de seguridad adecuados y robustos.

  Existen diferentes tipos de procesos y mecanismos de autenticación como son Basic, Digest, autenticación en dominio (NTLM, Kerberos), autenticación con single sign on, diferentes opciones para aplicar multifactores de autenticación (password, token, Smart-card, huella dactilar, retina), autenticación desde la aplicación web.

  Profundizando más en la autenticación desde la aplicación web, las consideraciones importantes para tener en cuenta son:

  - Una vez introducido el password, si es correcto, se le asigna un identificador de sesión al usuario.
  - Normalmente, este id de sesión se gestiona mediante cookies y se usa durante todas las operaciones que realice el usuario para así no tener que solicitar nuevamente el password.
  - El id de sesión por tanto se convierte en las credenciales del usuario y es un dato que requiere máxima seguridad para evitar así posibles ataques de alto impacto.
  - El id de sesión debe tener una longitud mínima de 128 bits y generarse con un algoritmo que no sea predecible.
  - Para ello, se debe transmitir el dato de manera cifrada usando TLS con certificado de cliente.

- Es muy recomendable utilizar un framework de desarrollo para su implementación para garantizar que el desarrollo es adecuado.
- Se debe establecer un tiempo de expiración del id de sesión para garantizar que se terminan las sesiones.
- El id de sesión se debe renovar en cada proceso de autenticación.
- Se debe regenerar el id de sesión ante situación o accesos a zonas de datos protegidos para garantizar que las credenciales son correctas.

Caso de uso de la vulnerabilidad web de exposición de datos sensibles

- Descripción de aspectos docentes adicionales:

  Uno de los datos más sensibles es la contraseña del usuario por lo que se debe garantizar que se almacena de manera segura en el sistema para que así se puedan mitigar los ataques que pueden sufrir. Para ello, se deben tener en cuentas las siguientes consideraciones (Johnson, s. f.):

  - Nunca almacenar las contraseñas en claro en el sistema y utilizar en su lugar funciones hash y el almacenamiento del valor obtenido.
  - La función hash por sí sola no garantiza que sea seguro ya que existen las denominadas rainbow tables que permiten mediante fuerza bruta y diccionario de hashes y contraseñas poder obtener la contraseña en claro a partir de un valor hash.
  - Para hacer robusto el proceso, se debe incorporar el concepto de salt, que no es más que un valor aleatorio que se genera y concatena a la contraseña, generando así el hash de este nuevo valor y evitando de esta manera los ataques por rainbow table.
  - Finalmente, se debe aplicar el concepto de key stretching que consiste en aplicar el hash n veces sobre el resultado que se va obteniendo de concatenar la contraseña y la salt al valor hash que se va obteniendo en cada iteración. De esta manera se previenen ataques de fuerza bruta mediante GPUs.

Caso de uso de la vulnerabilidad web de pérdida de control de acceso

- Descripción de aspectos docentes adicionales:

- El proceso de autorización o control de acceso (*Access Control - OWASP Cheat Sheet Series*, s. f.) consiste en disponer de un mecanismo que se activa o invoca cuando un usuario autenticado intenta acceder a un recurso restringido o solicita ejecutar acciones y que se encarga de comprobar si efectivamente el usuario tiene permisos para ello en base a los permisos que tenga asignado como usuario o como rol que se le haya podido asociar.

  Existen tres maneras de gestión del control de acceso:

  - MAC (*Mandatory Access Control*): donde el administrador del sistema es el único encargado de gestionar los permisos a los recursos.
  - DAC (*Discrectional Access Control*): donde el administrador delega la función en los propietarios de los objetos para que realicen esta función.
  - RBAC (*Roled Based Access Control*) que como su propio nombre indica, está basado en roles.

  Finalmente, es interesante reseñar que el uso de procedimientos almacenados de base de datos para que desde la aplicación web se realicen consultas del control de acceso de los usuarios es una buena práctica desde el punto de vista de la seguridad.

- CORS (*Control de Acceso HTTP (CORS) - HTTP | MDN*, s. f.) es un mecanismo de seguridad que permite controlar las peticiones que se denominan de origen cruzado (cross-origin) entre diferentes dominios, permitiendo o no la ejecución de estas peticiones en función de la aplicación de la política de seguridad CORS.

  Por defecto, peticiones asíncronas (AJAX, XMLHttRequest, etc) entre diferentes dominios son bloqueadas por la política CORS mientras que, si se realizan desde HTML o Javascript con API DOM, en este caso por defecto son permitidas.

  En la comunicación que se establece entre el cliente y el servidor, se analizan los atributos de las cabeceras CORS (*CORS Headers*) para conocer si se autoriza o no la petición de origen cruzado.

Caso de uso de la vulnerabilidad web de secuencia de comandos en sitios cruzados

- Descripción de aspectos docentes adicionales:

- CSP (*Content Security Policy (CSP) - HTTP | MDN*, s. f.) es un mecanismo de seguridad del que disponen los navegadores web por el permiten impedir que un script externo de una aplicación web pueda acceder al DOM, datos o cookies de otra aplicación web.

  De esta manera, con CSP se pueden mantener separadas las aplicaciones web siendo un ejemplo habitual el de las pestañas que se abren en una navegación web, donde cada una de estas pestañas puede ser una aplicación distinta. Además, el usuario puede estar logado por lo que no disponer de este mecanismo de seguridad puede producir ataques de seguridad por las que acceder a una aplicación web de un usuario que ya estuviera logado con el daño potencial que esto supondría.

- Existen tres tipos de ataques de XSS (*Cross Site Scripting)*:
  - XSS Reflejado: permite al atacante ejecutar comandos html o javascript en el navegador del usuario.
  - XSS Almacenado: permite al atacante almacenar datos maliciosos en el servidor de la aplicación que serán ejecutados por los usuarios.
  - XSS DOM: vulnerabilidad basada en el Modelo de Objetos del Documento (DOM).

Anexo D. Herramientas y aplicaciones funcionando en el entorno de seguridad web

En este anexo se encuentra el detalle de las capturas de imágenes de las herramientas y aplicaciones del entorno de seguridad una vez que se ha puesto en marcha. Se considera información relevante para incorporar dado que puede ser utilizada como referencia para su consulta durante el proceso de arranque para verificar que es correcto.

A1 – DVWA

Figura 46. *Arranque de la aplicación DVWA en el entorno de seguridad informática*

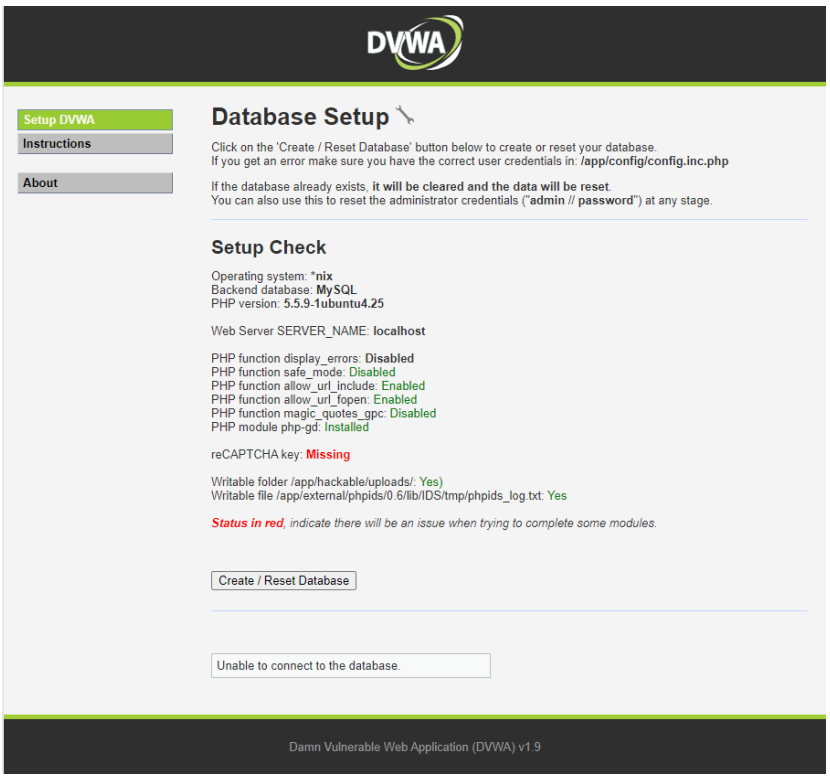
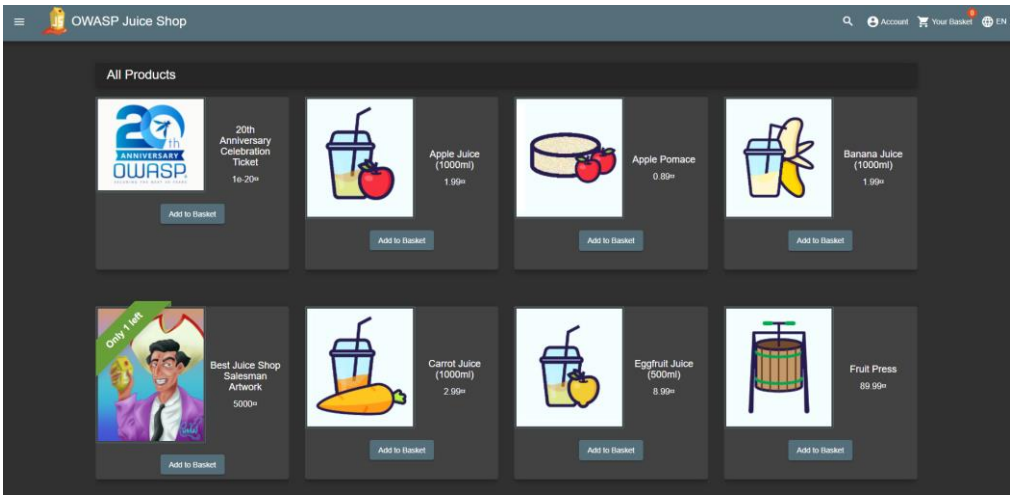


Figura 47. *Página de inicio de la aplicación DVWA ejecutada en el entorno de seguridad informática*

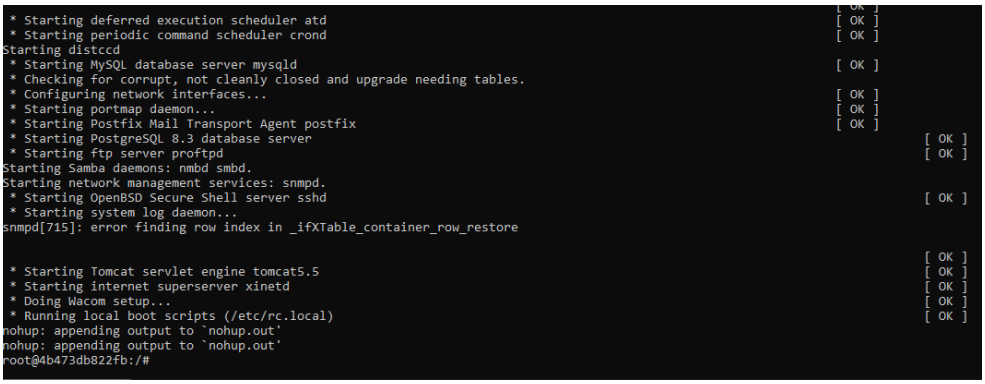
A2 – OWASP Juice Shop

Figura 48. *Página de inicio de la aplicación Juice Shop ejecutada en el entorno de seguridad informática*



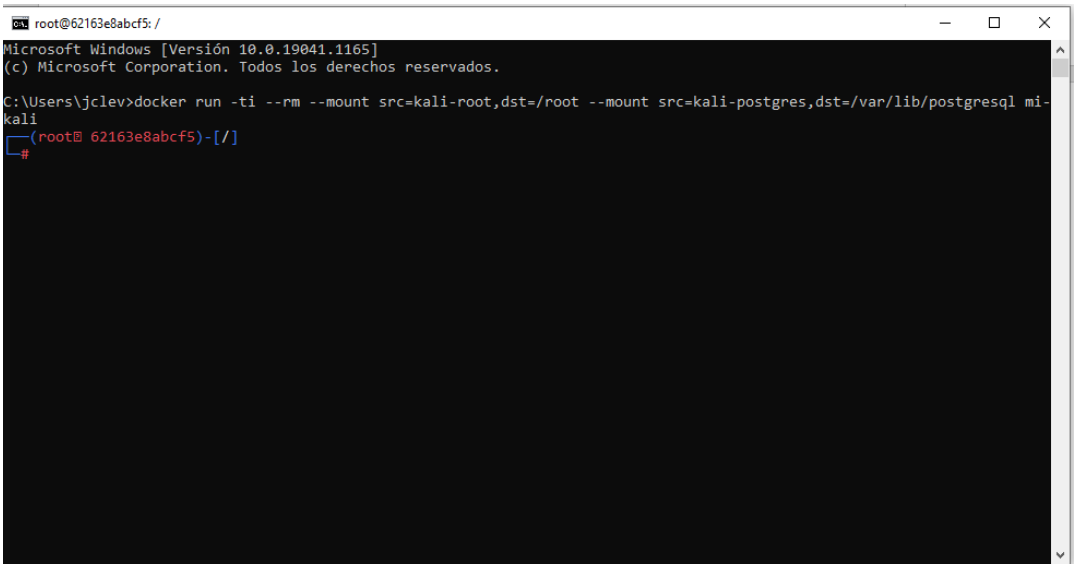
A3 – Metasploitable

Figura 49. *Terminal de inicio de la aplicación Metasploitable ejecutada en el entorno de seguridad informática*



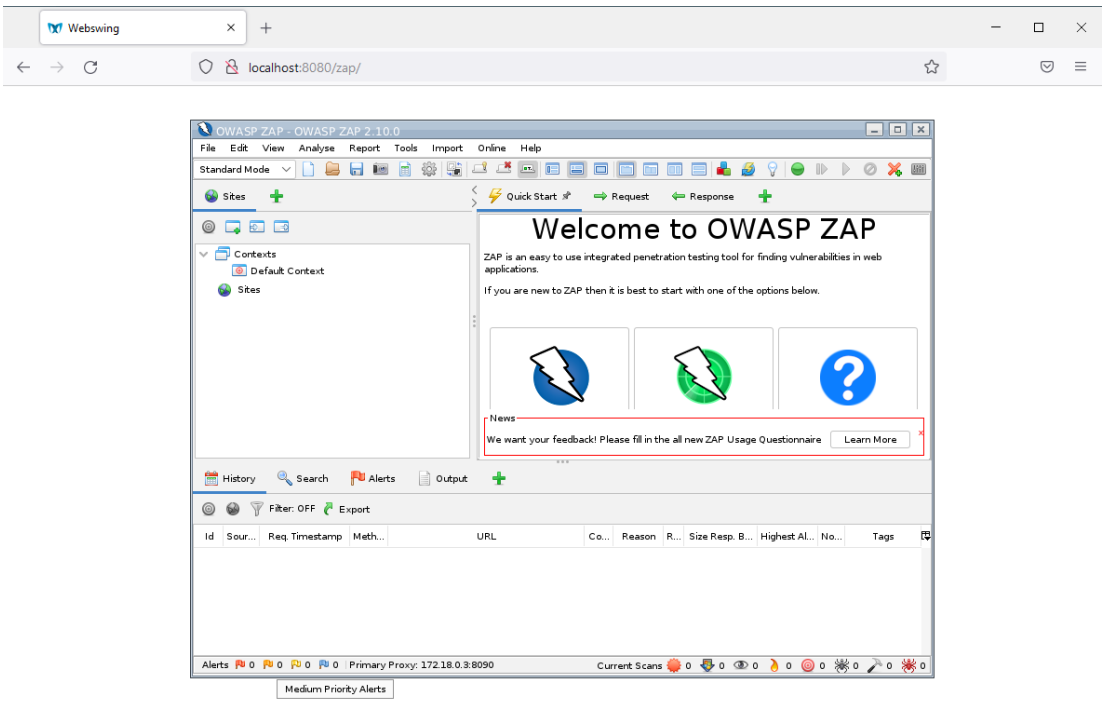
H1 – Kali Linux

Figura 50. *Terminal de inicio de la herramiena Kali Linux ejecutada en el entorno de seguridad informática*



H2 – OWASP ZAP

Figura 51. *Página de inicio de la herramienta OWASP Zap ejecutada en el entorno de seguridad informática*

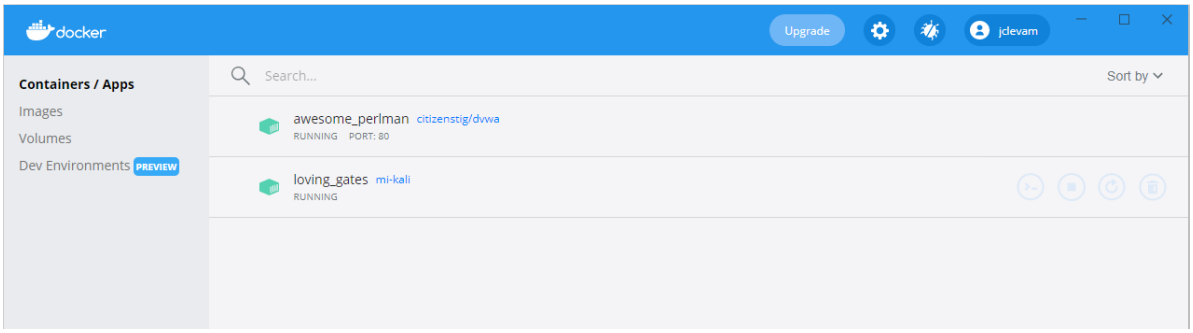


Anexo E. Contenedores arrancados en Docker para la ejecución de los retos de los casos de uso docentes

En este anexo se incorporan las capturas de imágenes de Docker Desktop para visualizar los contenedores que se arrancan y por tanto utilizan en la resolución de cada uno de los retos de los casos de uso docentes.

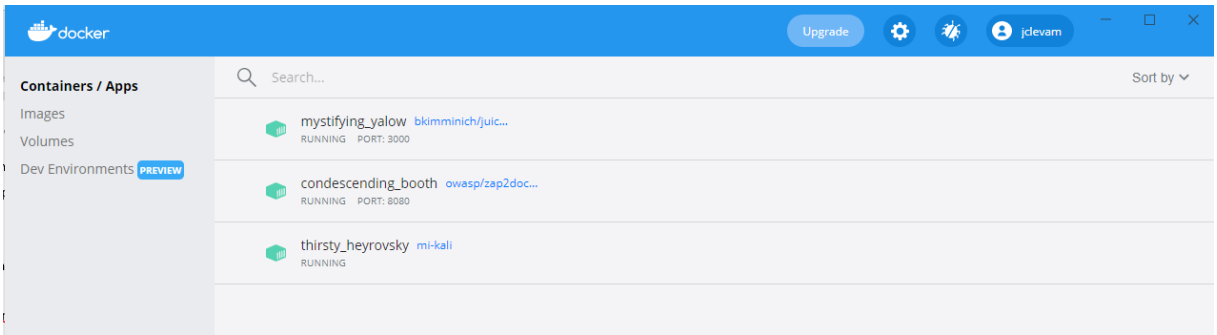
Reto vulnerabilidad inyección SQL en aplicación DVWA

Figura 52. *Contenedores arrancados para el reto Inyección SQL en la aplicación DVWA*



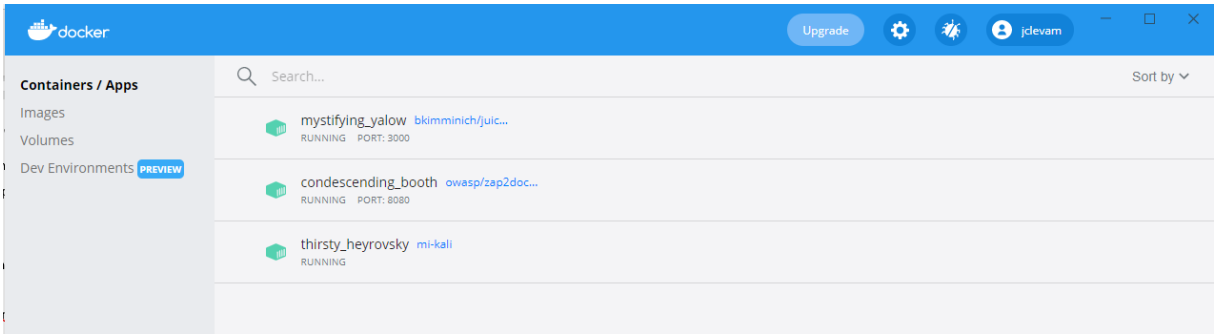
Reto vulnerabilidad inyección SQL en aplicación OWASP Juice Shop

Figura 53. *Reto vulnerabilidad inyección SQL en aplicación Juice Shop - Contenedores arrancados*



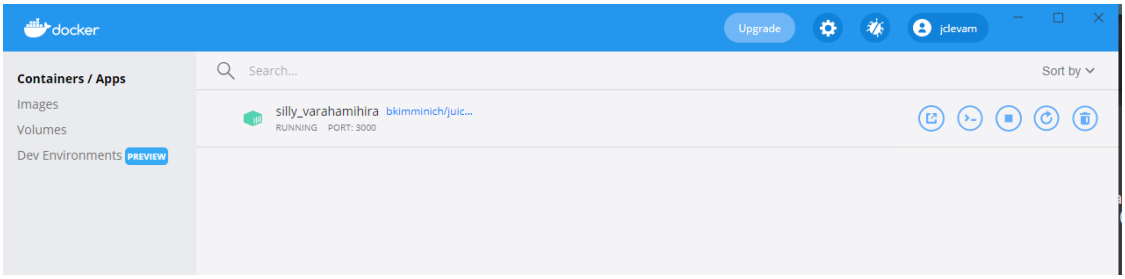
Caso de uso de la vulnerabilidad web de pérdida de autenticación

Figura 54. *Reto pérdida de autenticación - contenedores arrancados*



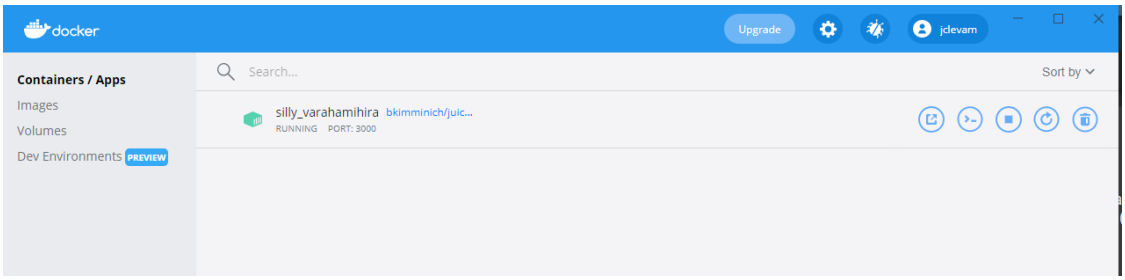
Caso de uso de la vulnerabilidad web de exposición de datos sensibles

Figura 55. *Reto pérdida de autenticación - contenedores docker arrancados*



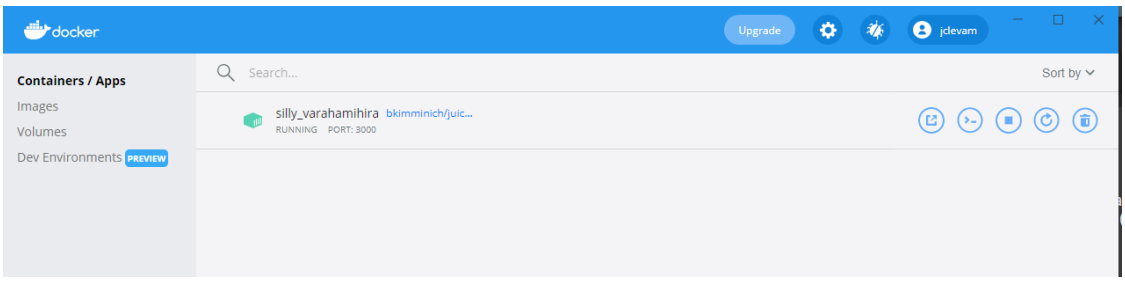
Caso de uso de la vulnerabilidad web de entidades externas XML (XXE)

Figura 56. *Reto entidad externa XXE - contenedores docker arrancados*



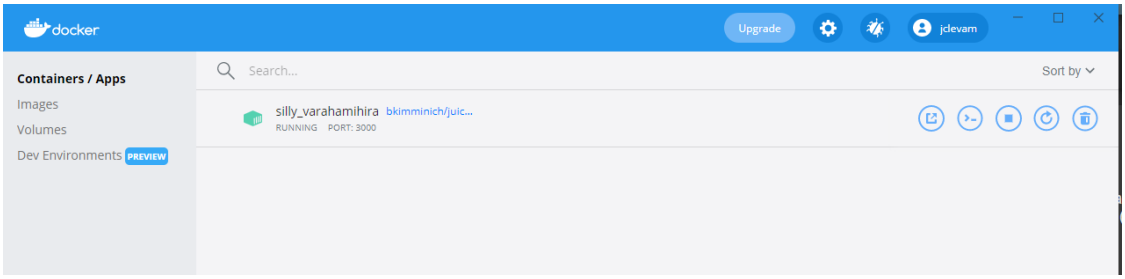
Caso de uso de la vulnerabilidad web de pérdida de control de acceso

Figura 57. *Reto pérdida de control de acceso - contenedores docker arrancado*



Caso de uso de la vulnerabilidad web de secuencia de comandos en sitios cruzados

Figura 58. *Reto Cross Site Scripting - contenedores arrancados*



Caso de uso explotación vulnerabilidad postgres máquina metasploitable

Figura 59. *Reto vulnerabilidad postgres máquina metasploitable - Contenedores arrancados*