Plataforma de gestión de escenarios de ciberseguridad para aprendizaje y entrenamiento

Francisco Barea, Irene Romero, José Ignacio Rojo, Víctor A. Villagrá y Julio Berrocal
Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid
Avda. Complutense, 30. 28040 Madrid.
Email: jfbarea@dit.upm.es, iromero@dit.upm.es, jirojo2@gmail.com, villagra@dit.upm.es, berrocal@dit.upm.es

***Resumen*—Estando el sector de la ciberseguridad en pleno auge, con una demanda de puestos de trabajo muy superior al número de profesionales cualificados en la actualidad, queda de manifiesto la importancia de la docencia de los nuevos profesionales del sector. Este artículo presenta una propuesta para proveer a los centros de enseñanza de una plataforma que posibilite la realización de prácticas docentes de laboratorio con escenarios complejos que cuentan con infraestructura dedicada al alumno para la resolución de cada ciberejercicio de forma aislada, optimizando el coste en infraestructura y operaciones necesario para su puesta en marcha. Para aislar los escenarios, se utilizarán contenedores Docker, con redes virtualizadas para dar conectividad a los diferentes contenedores que conforman el escenario. Cada equipo dispondrá de un entorno dedicado al equipo y aislado del resto de equipos para realizar el escenario. La plataforma deberá ser capaz de gestionar estos entornos y de asegurar su correcto aislamiento, minimizando la utilización de recursos en todo momento para facilitar la escalabilidad del sistema. Asimismo, este documento también incluye una propuesta de ejercicios a ser desplegados con la plataforma en formato de prácticas docentes de ciberseguridad donde el alumno se enfrentará a los conceptos fundamentales del hacking ético y las diversas categorías que debe dominar todo profesional del sector de la ciberseguridad.**

***Index Terms*—Formación, Competición, Contenedores, Docker, Laboratorios docentes, Entrenamiento, Ciberseguridad**

Tipo de contribución: Formación e innovación educativa (límite 8 páginas)

I. Introducción

I-A. Ámbito

Dada la necesidad de protección de los sistemas de información y redes de telecomunicación, el gran aumento de amenazas y ataques a estos entornos, y el aumento de la concienciación de seguridad de todos los actores involucrados, este documento se centra en el ámbito de la ciberseguridad, en concreto en metodologías alternativas de formación en este campo mediante el uso de los avances tecnológicos más recientes, primando la usabilidad y la escalabilidad, así como la optimización de recursos necesarios para llevar a cabo esta propuesta formativa en un laboratorio docente.

La ciberseguridad está cobrando especial importancia en las empresas del sector TIC, ampliando el mercado laboral de profesionales expertos tanto en hacking ético, análisis forense, bastionado de servicios, o consultoría en general.

La amplia demanda de estos días de profesionales del sector de la ciberseguridad y del hacking ético en particular ha revolucionado los métodos de formación tradicionales. La fuerte componente creativa, así como la amplia base técnica y tecnológica necesarias para el correcto desempeño de esta profesión penaliza los planes tradicionales y requiere un enfoque más práctico y directo.

Por esto mismo han cobrado especial importancia las certificaciones profesionales en este sector en particular. Estas certificaciones se suelen basar en un laboratorio práctico, con un examen final sobre los contenidos practicados en el laboratorio. Las certificaciones mejor valoradas difieren en contenidos, pero todas tienen en común el enfoque puramente práctico, siempre fomentando el pensamiento lateral de los profesionales, que en el fondo es la cualidad más importante de los mismos para este sector en particular. Los laboratorios prácticos asociados a las certificaciones profesionales cuentan con una infraestructura compleja, poco escalable y de muy alto coste, ya sea por la propia infraestructura o por el contenido docente que busca emular escenarios de ataque reales y que por tanto requieren de un profundo análisis técnico.

I-B. Objetivo

Aunque en ciertos escenarios puede permitirse asumir el alto coste de los laboratorios típicos de las empresas especializadas en certificación profesional, replicar dichos entornos con un fin educativo en una universidad o en un laboratorio interno de una empresa u organización resulta a día de hoy lejos de ser óptimo. Por esta misma razón esta propuesta busca la innovación en la infraestructura necesaria para realizar sesiones docentes de hacking ético mediante la utilización de nuevas tecnologías que se irán definiendo en los siguientes puntos de este documento, siempre con el fin de optimizar la escalabilidad y los costes necesarios para la realización de sesiones de entrenamiento en un laboratorio docente. El objetivo de este documento es fomentar la educación práctica en ciberseguridad, proporcionando una plataforma de entrenamiento escalable y con capacidad y flexibilidad suficientes para el diseño de ciberejercicios complejos que pongan a prueba las habilidades y competencias de los alumnos en un ámbito de laboratorio. La plataforma provee de ciberejercicios basados en escenarios que pondrán a prueba a los alumnos y facilitará su aprendizaje con un enfoque práctico. Asimismo, la plataforma facilita el seguimiento de los progresos de los alumnos por parte de los docentes, tanto con un sistema de puntuación general, fomentando la competitividad entre los participantes, como con un sistema de calificaciones personalizado a disposición del docente. Todo este sistema de puntuación se encuadra en el marco de la gamificación, enfoque que se ha probado eficaz a la hora de potenciar el compromiso del alumno en el aprendizaje[1]. Y no menos importante, se busca un diseño modular del contenido, en este caso los ciberejercicios a realizar por los alumnos. En este aspecto la plataforma busca la facilidad

de incorporación de nuevos ciberejercicios en sus diferentes modalidades, pudiendo ser diseñados por los propios docentes, o incluso propuestos por los estudiantes, fruto de sus estudios en la materia, para ser resueltos por sus compañeros en un modelo colaborativo.

I-C. Estructura del artículo

En esta memoria se define una propuesta diseño y desarrollo de una plataforma de gestión de escenarios de ciberseguridad para aprendizaje y entrenamiento.

La segunda sección describe el estado actual de la formación en materia de ciberseguridad, tanto títulos universitarios de grado o máster, ya sean oficiales o propios, certificaciones profesionales o competiciones prácticas relacionadas.

En la tercera sección se detalla la arquitectura de la plataforma que dotará de infraestructura a los ciberejercicios y servirá de interfaz de usuario.

En la cuarta, se definen los requisitos de los ciberejercicios para que se puedan albergar en la plataforma, así como recomendaciones y buenas prácticas a la hora de diseñar los ciberejercicios y los escenarios que los componen. Se detallan los tipos de entorno que pueden conformar los escenarios y alternativas en el caso de querer expandir las características fuera del entorno de virtualización que propone la plataforma.

La quinta sección contiene una propuesta de casos de uso.

Por último, la sexta y séptima secciones cierran el artículo comentando posibles líneas de continuación del trabajo y las conclusiones finales del mismo respectivamente.

II. Estado de la formación en ciberseguridad

El mapa actual de formación en ciberseguridad cuenta con una oferta muy diversificada, desde asignaturas de grado hasta másteres especializados, pasando por certificaciones profesionales, talleres y competiciones.

II-A. Formación universitaria

En cuanto al material impartido en las asignaturas de grado oficiales se pueden encontrar asignaturas de especialidad que conglomeran los principios básicos de la ciberseguridad, con el objetivo de conocer y aplicar las tecnologías que proporcionan seguridad a los sistemas y redes TIC, conociendo los fundamentos organizativos y criptográficos en los que se basan las tecnologías de seguridad y su aplicación a la programación y desarrollo de software seguro.

En las asignaturas de grado relativas a la ciberseguridad en España se trabajan temarios relativos a criptografía, firma electrónica, programación segura, amenazas de internet, servicios de control de acceso y servicios de protección de la comunicación.

A nivel de máster universitario, existen tanto títulos oficiales como títulos propios o bien elaborados por la propia universidad.

Los másteres oficiales especializados en ciberseguridad profundizan los conceptos introducidos en las asignaturas de nivel de grado. Cuentan con asignaturas de amenazas y contexto de la ciberseguridad, servicios de control de acceso, servicios de seguridad en red, protección de sistemas y servicios, protección de la información, ingeniería inversa y análisis de malware, gestión de riesgos y operaciones, privacidad, evidencias forenses, seguridad en el desarrollo de software, diseño de estrategias corporativas relativas a la ciberseguridad, sistemas de gestión de seguridad de la información y seguridad física. Adicionalmente, introducen una experiencia más práctica familiarizando al alumno con los procedimientos de auditoría y se complementan con prácticas en empresas del sector.

Las asignaturas tanto de grado como de máster cuentan, o sería idóneo que contaran, con componentes prácticos en formato de prácticas de laboratorio docente.

II-B. Certificaciones profesionales

Por otro lado, las certificaciones profesionales varían en ámbito y dificultad. Su reconocimiento profesional también varía en función de la región, siendo CEH la certificación más reconocida a nivel europeo y africano, y OSCP/OSCE las certificaciones más requeridas a nivel Americano. Las certificaciones profesionales están gestionadas por organizaciones de acreditación independientes, como ComTIA, EC Council, GIAC, ISACA, (ISC)2 y Offensive Security.

Típicamente, las certificaciones se dividen en tres segmentos: nivel de entrada, nivel intermedio y nivel experto, y todas ellas constan típicamente de un periodo de entrenamiento culminado en un examen final. Las certificaciones de nivel de entrada están orientadas a exponer al alumno a los conceptos fundamentales, mejores prácticas, herramientas esenciales, últimas tecnologías y metodologías.

Algunas de estas certificaciones, como es el ejemplo del OSCP y el OSCE, centran su evaluación en un componente fundamentalmente práctico, dando acceso durante un periodo de entrenamiento a un laboratorio con una serie de escenarios donde el alumno puede practicar y obtener la técnica propuesta en ese ejercicio.

II-C. Competición

Esta práctica se asemeja al formato de competición de tipo CTF o capturar la bandera por sus siglas en inglés. Capture the Flag (CTF) es un tipo de competición de ciberseguridad que cuenta con dos modalidades principales: Jeopardy y Ataque-Defensa, pudiendo considerarse una opción híbrida entre ambas.

Jeopardy en este caso se trata de un formato en el que hay una serie de retos o ejercicios agrupados por categorías, y los participantes obtienen puntos por la resolución de estos retos. Los participantes pueden organizarse por equipos si la organización así lo permite.

Para la modalidad de Ataque y Defensa se organizan los participantes por equipos, donde cada equipo dispone de una infraestructura dedicada (servidores, red, etc) con una serie de activos vulnerables. El objetivo de cada equipo es doble: arreglar sus vulnerabilidades y atacar al resto de equipos, ya que los puntos se dividen en las categorías de ataque (se obtienen puntos por atacar otros sistemas) y defensa (obteniendo puntos por no ser atacado, ya sea por no ser vulnerable o por no haber sido objetivo de ningún ataque)

Para la realización de prácticas docentes de laboratorio, el formato Jeopardy de competición CTF resulta particularmente atractivo dada su facilidad de despliegue y operación, y va a ser este mismo formato el propuesto en este documento. Una posible contraposición sería el coste y la complejidad de la infraestructura técnica para desplegar los escenarios que

componen los ejercicios, por lo que será el factor que se trata de optimizar en la plataforma propuesta.

II-D. Plataformas disponibles

Por un lado existen una considerable cantidad de soluciones gestionar infraestructuras que soporten laboratorios docentes sobre ciberseguridad, en [3], [4], [5], [**?**] se muestran varios ejemplos. El problema que acarrean estas soluciones es que están orientadas a virtualización, lo cual exige un alto coste en infraestructura que se pretende subsanar con la solución propuesta. Por un lado existen diversas plataformas para competición que no proporcionan una infraestructura sobre la que ejecutarlas, a continuación se listan unos ejemplos de dicho tipo de plataformas.

- **Mellivora:** escrita en PHP y publicada en Github [7]. Esta solución provee de una plataforma web que permite la gestión de los retos, la presentación de los mismos a través de un enunciado y un tablón de puntuaciones y la validación de los resultados.
- **CTFd:** también disponible en Github [8] y escrito en Python. La capa de presentación cumple las expectativas que se esperan de este tipo de plataformas, pero tampoco tiene una capa de gestión de infraestructura, que será la diferencia que propone la plataforma diseñada en este documento.
- **FBCTF:** también disponible en Github [9], con una plataforma web que se basa en un mapa mundial donde cada reto representa un país. De nuevo la capa de presentación cumple las expectativas, pero no cuenta con una gestión de la infraestructura.

En cuanto a las soluciones existentes en el mercado que cubren cuentan gestión de la infraestructura se venden típicamente como servicio. Existen empresas como HackingLab o Indra, que cuentan con plataformas y servicios de organización de eventos formativos o de competición. Sus soluciones se basan en tecnologías de virtualización, que suponen un requisito importante de hardware.

II-E. Solución propuesta

En definitiva, la solución propuesta en este artículo cuenta con una plataforma web capaz de orquestar y presentar los ejercicios a los participantes, incluida la infraestructura necesaria para desplegar los escenarios, así como dotar de herramientas de seguimiento y control al personal docente. La infraestructura de los diferentes escenarios que componen los ejercicios que se ofertarán a los alumnos en el laboratorio viene dada también por la plataforma, a través de tecnologías de virtualización y contenerización. Poder contar con infraestructura para los escenarios, permite un diseño más complejo de los ejercicios, eliminando los problemas de compatibilidades y de requerimientos para los participantes y permite aislar en caso de análisis de muestras de malware. En definitiva, la solución propuesta en este documento serviría para poder ofertar prácticas docentes de ciberseguridad para cubrir el itinerario expuesto en este documento, pudiendo cubrir prácticas de criptografía, certificados digitales, configuración y prueba de funcionamiento para software de defensa perimetral, análisis de muestras de malware, auditoría, ingeniería inversa, análisis forense, etc. En los siguientes puntos se entrará en el detalle de la arquitectura del sistema propuesto, y del formato de los ejercicios y escenarios que pueden ofrecerse con esta solución.

III. Arquitectura del sistema

La arquitectura que va a seguir el sistema prima la escalabilidad a la hora de gestionar recursos virtualizados para poder lanzar los retos, respetando el nivel de aislamiento que requieren.

III-A. Stack tecnológico

Las tecnologías utilizadas para desarrollar la plataforma propuesta se resumen en la figura 1

Figura 1. Tecnologías que componen las solución.

Para dotar de infraestructura al proyecto, se utilizan contenedores docker de diferentes tipos. Para un entorno sencillo se puede utilizar un único nodo docker con varios contenedores, pero para un despliegue productivo se recomienda una configuración en modo swarm (clúster de docker) con al menos tres nodos, facilitando así el consenso en caso de fallo simple.

Los servicios fundamentales de la plataforma se reducen a un servicio de base de datos, con un contenedor basado en una imagen Debian con MongoDB, y a un servicio web también desplegado en un contenedor basado en una imagen Debian con NodeJS que albergará una API REST y dos aplicaciones frontales basadas en AngularJS.

El framework sobre el que se basa la plataforma es NodeJS, un entorno en tiempo de ejecución multiplataforma y de naturaleza asíncrona. Se asocia con conceptos de alta escalabilidad, ya que su arquitectura asíncrona basada en eventos permite exprimir al máximo el rendimiento en procesos típicos de servicios de API en el que prima la métrica de peticiones atendidas por segundo.

La capa servidor pues, implementa una API REST que alimenta dos aplicaciones web basadas en AngularJS y se nutre de una fuente de datos documental MongoDB.

La implementación base para las funcionalidades HTTP elegida es ExpressJS, que dota al sistema del protocolo HTTP y el modelo REST, así como un potente enrutador basado en el concepto de middleware [10].

Para el acceso a datos se utiliza el ODM (Object Document Mapper) Mongoose, que dota de una capa de abstracción sobre el motor a bajo nivel que ofrece MongoDB.

La capa frontal o frontend estará compuesta por dos aplicaciones SPA (single page application) escritas con el framework AngularJS. Ambas aplicaciones se alimentan de la API REST que se ha visto en el comienzo de esta sección.

La primera aplicación gestiona toda la interfaz de usuario de la plataforma de entrenamiento, mientras que la segunda gestiona la parte del panel de administración de la plataforma.

III-B. Arquitectura Lógica

La arquitectura lógica seguiría el diagrama de la figura 2, con dos tipos de usuario bien diferenciados: los participantes o alumnos, que irán contra la aplicación AngularJS principal, y los admin o docentes, que utilizarán el panel de administración, que es una aplicación AngularJS diferente aunque ambas se alimenten del mismo servicio de API.

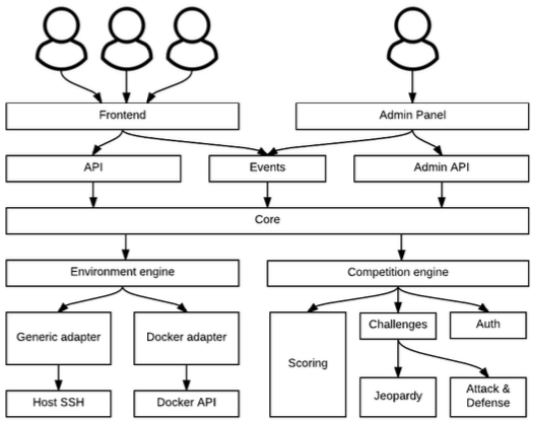

Figura 2. Arquitectura lógica de la plataforma

La arquitectura física se ilustrará con un ejemplo concreto ya que la solución propuesta tiene entre sus objetivos la flexibilidad de adaptarse al entorno del cliente final.

El sistema, por tanto, se compondrá de varias partes bien diferenciadas:

- Un conjunto de LANs para los usuarios. El escenario más común será aquel en el que los usuarios de la plataforma se dividan en equipos. Por poner números de ejemplo, se supone un total de diez equipos, cada equipo compuesto por diez integrantes, todos ellos situados en un laboratorio docente. En este caso particular se contaría con diez sendas LANs, aisladas entre sí y enrutando el tráfico hacia el punto de entrada a la plataforma, pero nunca hacia las LANs de los otros equipos.
- Después de estas LANs de equipos, estaría la LAN de la plataforma. Lo primero que se encuentra en este segmento de red será un balanceador de carga (ya sea nginx, haproxy o traefik) que repartirá las peticiones entre los servidores de la plataforma correspondientes. Después, estarían los servidores de la plataforma (dos, para este ejemplo), que sirven tanto el frontend (artefactos compilados de la aplicación SPA escrita en AngularJS) como la API REST y los Websockets que componen el servicio de la plataforma.
- Además de los servidores web, se disponen nodos Docker configurados en modo swarm (cluster de docker) para la virtualización de los retos. En este escenario concreto no se hace uso del adaptador genérico de entornos virtualizados descrito en la figura de arquitectura, sino que se usará únicamente Docker. Ya que el motor de virtualización de escenarios se comunica con el cluster docker mediante su API REST, no es relevante el número de nodos docker aprovisionados (aunque para el ejemplo expuesto se supondrá que hay un total de 5 nodos para servir a los 100 participantes)
- Finalmente se cuenta con un nodo de base de datos MongoDB, escalable a un cluster compuesto por varios nodos de ser necesario, para almacenar todos los datos relacionados al backend del sistema.

III-C. Gestión de la infraestructura de los escenarios

Para la gestión de la infraestructura se pretende utilizar un método de virtualización lo más ligero posible, con el fin de simplificar al máximo esta funcionalidad y contener los costes de infraestructura.

Aunque la plataforma permitirá diferentes soluciones para proveer la infraestructura, este artículo se centra en un método de virtualización ligera basada en contenedores Docker [2].

Docker utiliza características de aislamiento de recursos del kernel de Linux, tales como cgroups y espacios de nombres (namespaces) para permitir que ”contenedores” independientes se ejecuten dentro de una sola instancia de Linux, evitando la sobrecarga de iniciar y mantener máquinas virtuales.

Otras soluciones disponibles en el mercado utilizan virtualización convencional mediante un hypervisor ya sea nivel 1 o nivel 2, lo cual supone asignar recursos fijos desde el inicio para todos los participantes y los escenarios, y cada instancia consumirá muchos más recursos que la contrapartida basada en contenedores, ya que cada instancia tendrá que disponer de su propio kernel y de su propio sistema operativo completo.

La ventaja principal de Docker es su ligereza, y dado que los contenedores que se utilizarán en los ciberejercicios están diseñados para ejecutar con un usuario distinto de root, se previenen posibles fugas del entorno aislado o sandbox. Adicionalmente, Docker permite la virtualización de redes y conectividad entre contenedores, pudiendo diseñar entornos con varios contenedores, totalmente aislados.

IV. Diseño de la plataforma

Como ya se ha expuesto en la sección anterior, la plataforma de entrenamiento se divide en dos aplicaciones funcionales o interfaces de usuario:

- Un **panel de control** donde definir los equipos, participantes, ciberejercicios, sesiones, reglamento, etc., y donde también se puede llevar a cabo un seguimiento de la sesión en curso (y anteriores) y generar reportes personalizados.
- La propia **plataforma de entrenamiento**, que proveerá a los participantes de un medio donde obtener y validar los ciberejercicios.

La modalidad de los ciberejercicios será CTF, donde cada ciberejercicio tendrá como objetivo la obtención de un token o bandera que la plataforma valida como solución para obtener los puntos correspondientes y dar por finalizado el ciberejercicio.

Cada ciberejercicio se compondrá de los siguientes campos:

- Un título o nombre del mismo, junto con una categoría que agrupe los retos de la misma tipología y una descripción detallada del ciberejercicio conteniendo su enunciado, guiando al participante durante la resolución del mismo.

- Una solución del reto o flag, que no será visible para el participante más allá del formulario correspondiente de validación para resolver el ciberejercicio.
- Un entorno virtualizado, si procede, asociado al ciberejercicio. El propósito y los diferentes tipos de entornos virtualizados se explicarán en detalle a continuación.
- Archivos adjuntos al enunciado a ser gestionados por la plataforma, o bien incluir enlaces a sistemas de distribución de ficheros externos en el propio texto del enunciado.

Todo ciberejercicio tendrá definida una puntuación bruta del reto. A esta puntuación se le podrán descontar el valor en puntos de una serie de pistas, canjeables por los participantes a cambio de información extra para la resolución del reto. Cada ciberejercicio puede tener cero o varias pistas, por un valor combinado de puntos menor o igual a la puntuación bruta de puntos del mismo.

Las categorías disponibles en la plataforma dependen de la definición realizada por el equipo docente que haya preparado el paquete de ciberejercicios, aunque se sugieren las siguientes:

- Análisis web
- Análisis forense
- Criptografía
- Ingeniería inversa
- Exploiting

IV-A. Entorno virtualizado

Cada ciberejercicio puede tener asociado un entorno virtualizado, que proporcionará al participante una infraestructura específica para la resolución del mismo.

La plataforma está preparada para dar soporte a varios tipos de entornos virtualizados, pero este artículo se va centrar en Docker, que ya se ha introducido en la sección III.

Se diferencian los tipos de entornos virtualizados mediante Docker en dos tipos principales: Entorno simple o entorno complejo.

IV-A1. Entorno simple: Los entornos simples tienen un único contenedor y por tanto no requieren de virtualización de redes ni de orquestación de contenedores aislados. En este caso implementan un Dockerfile que, a partir de una imagen base, construye un servicio vulnerable para que el equipo participante lo comprometa y obtenga el token o bandera objetivo, para su posterior validación en la plataforma de entrenamiento descrita en este documento.

En el diagrama de la figura 3 se puede ver cómo el usuario obtiene los detalles de conexión al entorno virtualizado del escenario simple del ciberejercicio. En este caso el usuario se conectará al puerto asignado para ese entorno virtualizado en concreto, puerto 20501 en este ejemplo, que la plataforma conectará internamente al puerto 1337 del contenedor que albergará el entorno del reto. La conectividad interna es transparente para el usuario y está orquestada por la plataforma de ciberejercicios, proporcionando al usuario la información necesaria para que pueda conectar con su instancia del entorno virtualizado. Algunas instancias requieren de autenticación, como puede ser un servicio ssh o un servicio git, que serían provistas también por la plataforma al usuario.

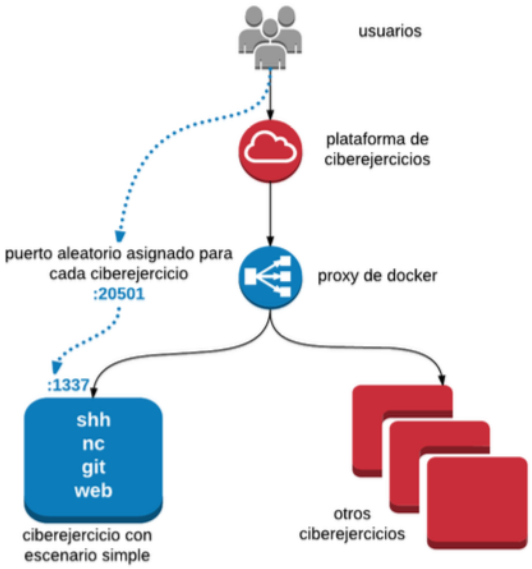

Figura 3. Diagrama de ejemplo de un entorno simple

IV-A2. Entorno complejo: Los entornos complejos aprovechan las características de virtualización de redes que brinda Docker. El entorno expondrá un servicio desde un punto de entrada o entrypoint. Este punto de entrada estará albergado en el contenedor principal del escenario, y tendrá conexión con la red general de Docker, por la que el participante accederá al servicio.

Como se ilustra en el diagrama de la figura4, Tras el contenedor principal, el escenario albergará más contenedores, conectados entre sí por, al menos, una red interna de Docker. El objetivo a la hora de diseñar estos escenarios es poder añadir complejidad a los mismos y simular saltos de redes diseñadas en capas.

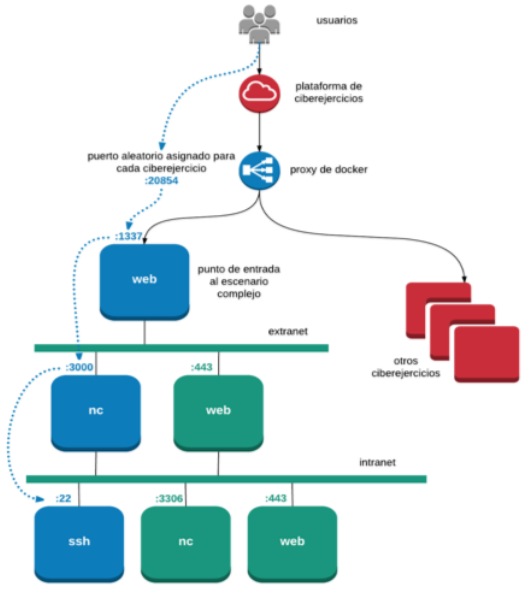

Figura 4. Diagrama de ejemplo de un entorno complejo

V. Diseño de los ciberejercicios

Los ciberejercicios deberán diseñarse con una serie de reglas para asegurar la compatibilidad con la plataforma. Estas reglas afectan especialmente a aquellos ciberejercicios que requieren de un entorno virtual para funcionar.

Aunque la plataforma está diseñada para dar soporte a otros métodos de virtualización de entornos, esta propuesta se centra en las especificaciones del entorno Docker. Por lo tanto, los ciberejercicios propuestos en este en la sección VI, estarán todos ellos adaptados para ser acoplados a la plataforma con el entorno Docker.

V-1. Especificaciones del proveedor de entornos virtualizados basado en Docker: Estas especificaciones son las reglas mencionadas al principio de esta sección y son:

1. Todo entorno virtualizado con Docker requiere exponer un único servicio al exterior. Además, el puerto de dicho servicio expuesto, deberá ser internamente a nivel contenedor el puerto 1337. La plataforma se encargará de enrutar dicho puerto y de exponer al usuario el puerto correspondiente en función a las estrategias de enrutado. Para el resto de servicios que pueda albergar el entorno virtual para ese ciberejercicio no se aplicaría esta restricción, ya que se encuentran en los segmentos internos de red del ciberejercicio en cuestión, que están totalmente aislados del resto de instancias.
2. Utilizar la misma imagen base para todos los retos, siempre que sea posible. Esta especificación no es de obligado cumplimiento, pero ayuda en gran medida a optimizar el consumo de recursos de la plataforma en su conjunto. Para el desarrollo de este documento se ha elegido la imagen base **debian:9** y derivados, primando siempre este tipo de imagen. En definitiva, es preferible evitar utilizar una imagen base diferente para cada reto, ya que esto supone almacenar todas esas imágenes base en cada nodo del swarm de docker.

La definición de los escenarios, o ciberejercicios complejos, se realizará mediante el panel de control de la plataforma. Además, debido a limitaciones diseño de la propia plataforma no se pueden utilizar herramientas como docker-compose ni stacks ni servicios de docker.

VI. Casos de uso

En esta sección se expondrá una serie de casos de uso de esta plataforma en forma de propuestas de ciberejercicios básicos para la modalidad de Jeopardy de varias categorías clave del *hacking* ético.

El objetivo de estos ciberejercicios es educar a los participantes en los aspectos básicos de cada categoría, fomentando la participación y el trabajo en equipo para superar una serie de retos complejos. Cada uno de ellos tiene asociada una puntuación que está integrada en un sistema de puntos que engloba toda la plataforma. Mediante el conteo de estos puntos, se puede organizar una competición entre los distintos puntos de usuario.

VI-A. Criptografía

Este tipo de retos son simples problemas de criptografía propuestos para ser resueltos por los usuarios de la plataforma. Un ejemplo de este tipo de ejercicios es el que se representa en la figura 5. En dicha figura se presenta una captura de la interfaz de la plataforma propuesta en este documento. En este caso se trata de un reto bastante simple que consta de un enunciado escrito y un campo con la respuesta pertinente. Resulta evidente que para este tipo de ejercicios no se necesita de la infraestructura de Docker para su ejecución.

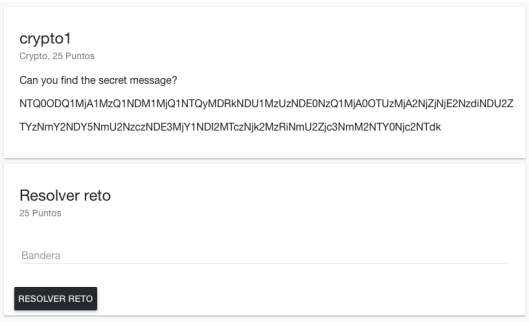

Figura 5. Ejemplo de reto criptográfico

VI-B. Exploiting

En la figura 6 se muestra un ejemplo de ejercicio sobre *exploiting*. Como se puede ver, este ejercicio sí que requiere un entorno virtual, de modo que sí se hace uso de la infraestructura de Docker.

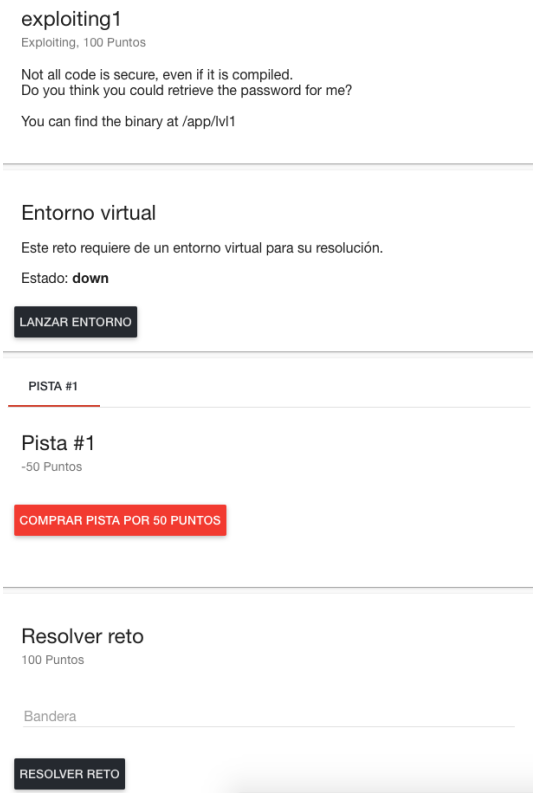

Figura 6. Ejemplo de reto sobre exploiting

Una vez lanzado el entorno virtual, la plataforma proporciona los datos de acceso al mismo como se muestra en la figura 7

Figura 7. Datos de acceso para entorno virtual

El objetivo de este ejercicio es obtener nociones básicas de las diferentes herramientas que existen para analizar datos binarios, obteniendo información a partir de las cadenas de texto potencialmente reconocibles en un bloque binario utilizando la herramienta *strings*.

Como se puede ver en la figura 6, es posible obtener pistas para resolver el ejercicio a costa de disminuir la puntuación conseguida en el ejercicio. Esto aporta otro matiz más de gamificación a la plataforma.

VI-C. Análisis Web

Dentro de la disciplina del análisis web, este ejemplo versa sobre inyección SQL. Como se puede ver en la figura 8, en este caso, cuando se lanza el entorno virtual, se despliega un contenedor que aloja un servidor web que el participante tendrá que atacar mediante una inyección SQL. Como se muestra en la figura, la plataforma proporciona una URL que apunta hacia la aplicación web objetivo.

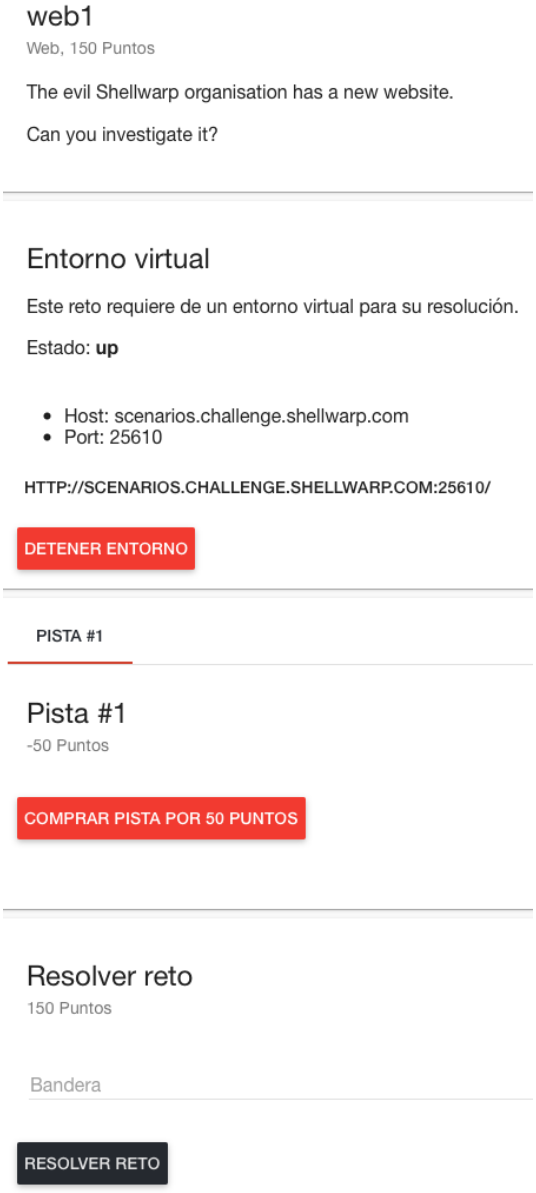

Figura 8. Ejemplo de reto sobre inyección SQL

VI-D. Pivoting

Usando la plataforma propuesta también es posible desplegar escenarios complejos como el de la figura 9.

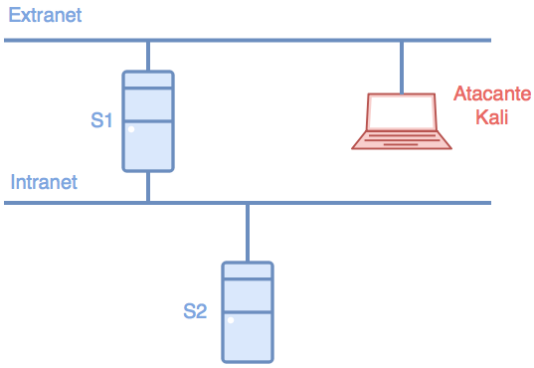

Figura 9. Escenario para ejercicio de Pivoting

Este ejercicio consiste en acceder al servidor S2 desde la máquina atacante usando el servidor S1 como pivote. Una vez

más, la plataforma desplegaría este escenario con dos redes y tres máquinas, proporcionaría los credenciales de acceso de la máquina atacante y esperaría un flag que se encontrase en la máquina objetivo (en este caso S2)

VII. Trabajo futuro

Uno de los trabajos futuros relacionados con esta plataforma sería incluirlo en un proyecto de innovación educativa donde se implante como herramienta de trabajo en una asignatura relacionada con la ciberseguridad de titulación oficial y medir el aumento de la productividad y satisfacción de los alumnos con respecto a las herramientas usadas en la actualidad en este tipo de asignaturas.

Por supuesto, otra línea de trabajo interesante sería aumentar la colección de ciberejercicios contenidos en la plataforma e integrarlo en los temarios de distintas asignaturas que se impartan en las diversas titulaciones relacionadas de alguna manera con la ciberseguridad.

Actualmente, la aplicación web está dividida en dos SPAs independientes (una para los "administradores" y otra para los "estudiantes").Una continuación del trabajo relacionado con esta plataforma podría ser integrar estas dos aplicaciones en una sola con los roles mencionados bien diferenciados.

Otra mejora a la herramienta propuesta sería integrarla con una interfaz gráfica que permita la visualización de escenarios complejos. Un ejemplo de dicha interfaz se puede ver en [11]

Además de Docker, se podrían desarrollar adaptadores a otros proveedores de virtualización o compartimentalización, o simplemente a entornos hardware dedicados, ya sean pequeños arduinos, drones, dispositivos fabricados a medida o grandes sistemas SCADA. Las principales casuísticas serían de adaptación de dispositivos *IoT* controlados por radio, ya sea *WiFi*, *ZigBee*, *BlueTooth* u otras tecnologías comunes del mercado.

VIII. Conclusiones

Una de las conclusiones finales de este documento es que la concienciación es fundamental a hora de combatir las amenazas y riesgos de internet, pero que no sólo basta con concienciar a la población, sino que es necesario un entrenamiento específico de los profesionales que se dedicarán al sector de la ciberseguridad.

Para facilitar el entrenamiento, o en su caso el aprendizaje básico, los escenarios virtualizados resultan especialmente útiles y prácticos, aunque repercutirán directamente en el coste de infraestructura y operaciones de la solución tecnológica que se decida utilizar para la realización de los ejercicios. Es por esto que la propuesta presentada en este documento se ha focalizado en utilizar nuevas tecnologías para optimizar dicho coste reduciendo drásticamente los requisitos de hardware y de operaciones respecto a otras soluciones que puedan estar actualmente en el mercado.

Referencias

[1] Muntean, Cristina Ioana."Raising engagement in e-learning through gamification." en Proc. 6th International Conference on Virtual Learning ICVL. vol. 1. sn, 2011.

[2] Merkel, Dirk. "Docker: lightweight linux containers for consistent development and deployment" en Linux Journal. vol. 2014. n. 239., 2014

[3] Nake, Niraj B., and Pushpanjali Chouragade. "Review on Virtual Laboratory in Network Security Education." en IJCA Proc. on National Conference on Recent Trends in Computer Science and Engineering, MEDHA 2015, n.1, pp. 40-28, 2015.

[4] Willems, Christian, and Christoph Meinel. "Practical network security teaching in an online virtual laboratory." en Proc. 2011 Intl. Conference on Security and Management (SAM 2011). 2011.

[5] Kunnath, Tony B., and Ashok Babu. "Cloud Based SDN-Lab for Network Security Education." en International Journal of Computer and Mathematical Sciences vol. 4, n. 12, 2015.

[6] xu, Le; huang, Dijiang; tsai, Wei-Tek. "Cloud-based virtual laboratory for network security education". IEEE Transactions on Education, 2014, vol. 57, no 3, p. 145-150, 2014

[7] Nakiami/mellivora. GitHub. 2018. Available at: https://github.com/Nakiami/mellivora. Accessed March 7, 2018.

[8] CTFd/CTFd. GitHub. 2018. Available at: https://github.com/CTFd/CTFd. Accessed March 7, 2018.

[9] facebook/fbctf. GitHub. 2018. Available at: https://github.com/facebook/fbctf. Accessed March 7, 2018.

[10] Bernstein, Philip A., en "Middleware: A Model for Distributed System Services",vol. 39, n. 2, pp. 86-98, 1996.

[11] Ruiz, Ricardo; Barea, J. Francisco; Álvarez-Campana, Manuel; Vázquez, Enrique "Herramienta gráfica de configuración de escenarios virtuales para ejercicios de ciberdefensa" en III Jornadas Nacionales de Investigación en Ciberseguridad (pp 134-141), 2017