PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
ESCUELA HÁBITAT, INFRESTRUCTURA Y CREATIVIDAD

TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

VULNERABILIDADES DEL CAMPUS VIRTUAL DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR – IBARRA MEDIANTE LA TÉCNICA DE PENTESTING (OWASP ZAP Y DEEPEXPLOIT)

ANGIE NAYELI LANDÁZURI RODRIGUEZ

TUTOR: GALO HERNÁN PUETATE HUERA

IBARRA – ECUADOR
FEBRERO, 2026

Ibarra, 11 de febrero del 2026

CERTIFICACIÓN TUTOR

En mi calidad de Tutor del Trabajo de Integración Curricular titulado: "VULNERABILIDADES DEL CAMPUS VIRTUAL DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR – IBARRA MEDIANTE LA TÉCNICA DE PENTESTING (OWASP ZAP Y DEEPEXPLOIT)", presentado por el estudiante Angie Nayeli Landázuri Rodríguez con cédula de ciudadanía N° 240001390-6, para obtener el Título de Ingeniero en Tecnologías de la Información.

Certifico que el trabajo cumple con todos los parámetros establecidos, mediante el cual el estudiante demuestra el desarrollo de competencias en el campo de conocimiento de su profesión con un nivel de argumentación coherente, para ser sometido a la evaluación por parte de los lectores.

Adicionalmente, se adjunta el certificado de porcentaje de originalidad de TURNITIN.


Turnitin Informe de Originalidad

Procesado el: 24-feb-2026 12:54 -05
Identificador: 2887438544
Número de palabras: 20128
Entregado: 1

VULNERABILIDADES DEL CAMPUS VIRTUAL DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR – IBARRA MEDIANTE LA TÉCNICA DE PENTESTING (OWASP ZAP Y DEEPEXPLOIT) Por ANGIE NAYELI LANDAZURI RODRIGUEZ

Índice de similitud
7%

Similitud según fuente
Fuentes de Internet 6%
Publicaciones: 3%
Trabajos del estudiante: 3%

GALO HERNAN PUETATE HUERA
Firmado digitalmente por GALO HERNAN PUETATE HUERA
Fecha: 2026.02.24 16:00:03 -05'00'

(f):_____

Mgs. Galo Hernán Puetate Huera

TUTOR DE TRABAJO

C.C.: 0401375787

PÁGINA DE APROBACIÓN DEL TRIBUNAL

El tribunal examinador, aprueba el presente trabajo en nombre de la Pontificia Universidad Católica del Ecuador Ibarra:

GALO HERNAN PUETATE HUERA
Firmado digitalmente por GALO HERNAN PUETATE HUERA
Fecha: 2026.02.24 16:00:15 -05'00'

(f): .....

Mgs. Galo Hernán Puetate Huera

C.C.: 0401375787

Firmado digitalmente por1002402061 DIEGO FERNANDO BAROJA LLANOS
Motivo:Soy el autor de este documento
Fecha:2026-02-24 16:10-05:00

(f):.....

Msc. Diego Fernando Baroja Llanos

C.C.: 1002402061

LUIS DAVID NARVAEZ ERAZO
Firmado digitalmente por LUIS DAVID NARVAEZ ERAZO
Fecha: 2026.02.25 07:49:14 -05'00'

(f):.....

Ms. Luis David Narvaez Erazo

C.C.: 1002868378

ACTA DE CESIÓN DE DERECHOS

Yo, *Angie Nayeli Landázuri Rodríguez*, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: “Se reconoce facultad de los autores y demás titulares de derechos de disponer de sus derechos o autorizar las utilizaciones de sus obras o prestaciones a título gratuito y oneroso, según las condiciones que determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia”.

Ibarra, 11 de febrero del 2026

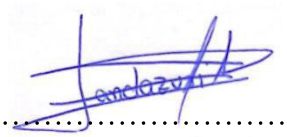
(f):_____ _____

Angie Nayeli Landázuri Rodríguez

C.C.: 2400013906

AUTORIA

Yo, Angie Nayeli Landázuri Rodríguez, portadora de la cedula de ciudadanía N° 2400013906, declaro que el presente trabajo de investigación es de total responsabilidad de la autora, y eximo expresamente a la Pontificia Universidad Católica del Ecuador Ibarra de posibles reclamos o acciones legales.



(f):……..

Angie Nayeli Landázuri Rodríguez

C.C.: 2400013906

DEDICATORIA

A Dios, porque él sabe lo que me costó llegar hasta aquí y cuantas veces el me levanto cuando no podía más, le dedico este sueño cumplido por ser mi guía en cada decisión, recordándome siempre que debía ser fuerte y valiente, y que no tenía por qué desanimarme, pues tal como me lo prometió en Josué 1:9, me acompañó en cada paso de este camino y nunca me dejó sola.

A mi mamá, Andrea Elizabeth Rodríguez Benavides, por ser el motor de mi vida y la persona que más creyó en mí, incluso cuando yo misma dudaba, gracias por tus oraciones, por tus sacrificios y por no soltar mi mano nunca, porque sé que muchas veces no he sido la mejor hija, tú siempre has estado ahí para mí con un amor incondicional. Eres una mujer luchadora y guerrera que dejo de lado sus propios sueños para dármelo todo a mí, y te admiro profundamente por que, sin importar lo difícil que fuera el camino, nunca te rendiste, este título es tanto mío como tuyo, y no me alcanzaría la vida para agradecerte que seas mi mayor ejemplo de fortaleza.

A mi papá, Sergio Gabriel Domínguez Gómez, por ser mi ejemplo de trabajo duro y por enseñarme con su propia vida lo que significa luchar por los sueños sin importar los obstáculos, gracias por no dejarme sola en ningún momento de este camino, especialmente cuando se puso difícil y las fuerzas fallaban, Te dedico este logro porque tú me enseñaste a luchar por lo que quiero, porque estuviste ahí en cada caída y porque, sin tu confianza y tu sabiduría, yo no sería la mujer que hoy termina esta etapa.

A mis hermanos, Diego Mateo Landázuri Rodríguez y Gabriel Alejandro Domínguez Rodríguez, por ser mis compañeros de vida y los que siempre estuvieron para mí en cada momento de esta etapa. Gracias por ser mi paz cuando más la necesitaba, por cada uno de sus abrazos y ocurrencias que me reiniciaban el alma y me devolvía la calma para seguir adelante. Con su compañía y su alegría me enseñaron a no rendirme, les dedico este título con todo mi amor, esperó que esto los motive a ver que los sueños si se cumplen con esfuerzo y dedicación.

AGRADECIMIENTOS

A mi abuelita, mis tíos, mi tía y mis primas: gracias de corazón por todo el apoyo que me han dado en esta etapa. Su cariño y el estar siempre pendientes de cómo me iba en la universidad fueron el recordatorio constante de que no estaba sola. Valoro muchísimo cada palabra de aliento, cada vez que me preguntaban por mis avances en las reuniones familiares y esa paz que me da saber que, pase lo que pase, siempre cuento con el respaldo incondicional de mi familia.

A la Pontificia Universidad Católica del Ecuador – Ibarra, por ser el lugar donde me forme profesionalmente. Mi más sincero agradecimiento a mi tutor, Mgs. Galo Puetate, por su guía y paciencia en este trabajo. Así mismo, a todos los docentes por compartir sus conocimientos y experiencias a lo largo de mi carrera brindando las bases necesarias para terminar esta etapa con éxito.

De manera muy especial, quiero agradecer al Mgs. Paul Enríquez, la Mgs. Daniela Tobar y la Ing. Tanya Recalde. Gracias por ir más allá de una enseñanza académica y convertirse en grandes amigos durante este proceso, gracias por cada uno de sus consejos, su paciencia y el apoyo constante que me brindaron. Me llevo su amistad como un gran regalo de esta etapa y siempre les estaré agradecida por haber estado ahí.

Finalmente, gracias a todos mis amigos y compañeros, pero quiero hacer una mención especial a tres personas que fueron especiales en este proceso. Damaris., Dayana y Josué. A Damaris, gracias por tu paciencia infinita y por ser esa voz de aliento que nunca me dejó tirar la toalla. A Dayana, porque, aunque la distancia nos separa, siempre has estado presenta, demostrándome que para una verdadera amistad no existen kilómetros. Y a Josué, gracias por las risas que disminuía la carga en clases y por tu lealtad de siempre. Ustedes tres convirtieron el estrés en anécdotas y este camino en un viaje compartido. Gracias por ser mi equipo y nunca dejarme sola.

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

Esta investigación evaluó las vulnerabilidades de seguridad del Campus Virtual de la Pontificia Universidad Católica del Ecuador, Sede Ibarra, mediante técnicas de pentesting automatizado. El estudio se enfocó en la plataforma Moodle, analizando su exposición a amenazas cibernéticas que podrían comprometer información académica sensible. Se empleó una metodología de pentesting ético que combinó OWASP ZAP para análisis dinámico de aplicaciones web y DeepExploit para explotación automatizada asistida por inteligencia artificial, ejecutándose en un entorno controlado que replicaba la infraestructura institucional. Los resultados identificaron 16 vulnerabilidades, clasificadas mediante CVSS v3.1, el modelo CIA y OWASP Top 10 2021. Se detectó una vulnerabilidad crítica de inyección SQL (CVSS 9.8), junto con fallas de configuración en cabeceras de seguridad, ausencia de controles anti-CSRF y exposición de información. DeepExploit validó la explotabilidad de credenciales SSH débiles. El análisis de riesgos priorizó las vulnerabilidades según su probabilidad e impacto, revelando un riesgo moderado-alto para la institución. Se concluye que, aunque la configuración de red es restrictiva, las vulnerabilidades de aplicación y la falta de actualizaciones representan amenazas significativas. El estudio demuestra la efectividad del pentesting automatizado para entornos educativos y recomienda la implementación de un programa continuo de evaluación de seguridad, actualización de componentes críticos y capacitación técnica para fortalecer la ciberseguridad institucional.

Palabras clave: pentesting, OWASP ZAP, DeepExploit, Moodle, vulnerabilidades web, ciberseguridad educativa, CVSS.

ABSTRACT

This research evaluated the security vulnerabilities of the Virtual Campus of the Pontificia Universidad Católica del Ecuador, Ibarra Campus, using automated pentesting techniques. The study focused on the Moodle platform, analyzing its exposure to cyber threats that could compromise sensitive academic information. An ethical pentesting methodology was employed, combining OWASP ZAP for dynamic application security testing and DeepExploit for AI-assisted automated exploitation, executed in a controlled environment replicating the institutional infrastructure. The results identified 16 vulnerabilities, classified using CVSS v3.1, the CIA model, and OWASP Top 10 2021. A critical SQL injection vulnerability (CVSS 9.8) was detected, along with security header misconfigurations, absence of anti-CSRF controls, and information exposure. DeepExploit validated the exploitability of weak SSH credentials. Risk analysis prioritized vulnerabilities based on probability and impact, revealing a moderate-high risk for the institution. It is concluded that, although network configuration is restrictive, application vulnerabilities and lack of updates represent significant threats. The study demonstrates the effectiveness of automated pentesting for educational environments and recommends implementing a continuous security assessment program, updating critical components, and technical training to strengthen institutional cybersecurity.

Keywords: pentesting, OWASP ZAP, DeepExploit, Moodle, web vulnerabilities, educational cybersecurity, CVSS.

INTRODUCCIÓN

En la actualidad, las tecnologías de la información y la comunicación desempeñan un papel fundamental en los procesos educativos, especialmente a través del uso de plataformas virtuales que facilitan la gestión académica y el aprendizaje en línea. Estas plataformas, como los campus virtuales basados en Moodle, almacenan y gestionan información sensible de estudiantes, docentes y personal administrativo, lo que las convierte en un componente crítico de la infraestructura tecnológica institucional.

Sin embargo, el incremento en el uso de aplicaciones web también ha venido acompañado de un aumento en los riesgos de seguridad informática, tales como accesos no autorizados, exposición de datos y fallos en la disponibilidad del servicio. En este contexto, resulta indispensable evaluar de manera sistemática el nivel de seguridad de estas plataformas para identificar vulnerabilidades que puedan ser explotadas por atacantes.

El pentesting o prueba de penetración se presenta como una técnica eficaz para evaluar la seguridad de los sistemas informáticos, ya que permite simular ataques reales con el objetivo de detectar debilidades antes de que estas sean aprovechadas de forma maliciosa. En particular, el pentesting automatizado ha cobrado relevancia debido a su capacidad para optimizar tiempos, reducir la intervención humana y mejorar la cobertura del análisis mediante el uso de herramientas avanzadas, incluyendo técnicas basadas en inteligencia artificial.

La presente investigación se centra en el análisis de las vulnerabilidades del Campus Virtual de la Pontificia Universidad Católica del Ecuador – Ibarra mediante el uso de OWASP ZAP y DeepExploit, aplicando una metodología estructurada de pentesting automatizado en un entorno controlado y autorizado. A través de la identificación, clasificación y análisis de riesgos de las vulnerabilidades encontradas, este estudio busca aportar información relevante para el fortalecimiento de la seguridad de la plataforma y la protección de la información académica.

CAPÍTULO I: ESTADO DEL ARTE

1.1 Antecedentes

El crecimiento de los entornos digitales en instituciones educativas ha incrementado la exposición a vulnerabilidades que comprometen la seguridad de los sistemas y la información académica. En los últimos años (2019-2024), universidades de todo el mundo han enfrentado incidentes relacionados con accesos no autorizados, robo de credenciales y ataques de denegación de servicio, lo que ha evidenciado la necesidad de fortalecer la ciberseguridad institucional mediante prácticas de evaluación continua y auditorías éticas.

En este contexto, el pentesting o test de penetración se ha consolidado como una metodología esencial para identificar debilidades en los sistemas antes de que puedan ser explotadas por atacantes. Este enfoque combina pruebas pasivas y activas para obtener una visión integral del nivel de riesgo de los sistemas informáticos.

- Las pruebas pasivas se centran en la observación y análisis del sistema sin interactuar directamente con los componentes internos, recopilando información pública y metadatos sin generar tráfico detectable.
- Las pruebas activas, en cambio, implican la ejecución controlada de ataques simulados para identificar vulnerabilidades específicas y comprobar su posible explotación.

Diversos estudios, como los de Patil & Kulkarni (2021) y Khan & Brohi (2020), destacan que la combinación de ambos tipos de pruebas permite una evaluación más completa del entorno digital, especialmente en contextos educativos donde los sistemas deben mantenerse operativos mientras se evalúa su seguridad.

Asimismo, metodologías como OWASP Testing Guide (2021) y estándares como el Common Vulnerability Scoring System (CVSS) se utilizan para clasificar y priorizar vulnerabilidades según su nivel de criticidad. Estas herramientas ofrecen un marco uniforme que facilita la comparación entre resultados de diferentes pruebas y ayuda a definir estrategias de mitigación adecuadas para cada tipo de amenaza.

En el ámbito educativo, los entornos de gestión de aprendizaje (LMS) como Moodle han cobrado relevancia como objetivo de análisis, dado que almacenan información sensible de

estudiantes, docentes y procesos académicos. Estudios recientes coinciden en que la falta de configuraciones seguras, actualizaciones constantes y políticas de control de acceso aumenta significativamente el riesgo de exposición.

De esta manera, el estado actual de la investigación muestra una tendencia creciente hacia el uso de herramientas automatizadas de pentesting, como OWASP ZAP y DeepExploit, que permiten realizar evaluaciones precisas de vulnerabilidades con menor intervención manual, favoreciendo la detección temprana de fallos de seguridad en entornos educativos.

1.1.1 Evolución de vulnerabilidades en entornos educativos (2019 -2024)

Durante el periodo comprendido entre 2019 y 2024, las instituciones educativas han experimentado un incremento notable en los incidentes de ciberseguridad, impulsado por la digitalización de los procesos académicos y la adopción masiva de plataformas de gestión del aprendizaje. La migración hacia entornos virtuales, especialmente tras la pandemia de COVID-19, amplió la superficie de ataque y expuso vulnerabilidades en servicios web, bases de datos y sistemas de autenticación.

Los ataques más frecuentes durante estos años han estado relacionados con inyecciones SQL, cross-site scripting (XSS), robo de credenciales mediante phishing, y configuraciones inseguras de servidores. Estas amenazas, clasificadas por el OWASP Top 10 (2021), continúan siendo los principales vectores de riesgo en aplicaciones web institucionales, lo que demuestra la persistencia de errores de desarrollo y mantenimiento en los sistemas educativos.

A nivel global, reportes como los del European Union Agency for Cybersecurity (ENISA, 2023) y el Educause Cybersecurity Program (2022) destacan que más del 70% de las universidades europeas y americanas han sufrido intentos de intrusión dirigidos a portales de gestión académica, correos institucionales o repositorios de investigación. En Latinoamérica, investigaciones recientes señalan un aumento en los ataques a sistemas universitarios debido a la baja inversión en infraestructura segura y la falta de políticas de actualización continua.

En Ecuador, el Instituto Ecuatoriano de Ciberseguridad (2023) ha identificado que muchas universidades aún carecen de protocolos estructurados para la detección temprana de

vulnerabilidades, lo que incrementa la exposición a brechas de seguridad. Estos hallazgos refuerzan la necesidad de aplicar metodologías de pentesting ético, adaptadas a los entornos académicos, que permitan evaluar los sistemas antes de que los atacantes lo hagan.

Por tanto, la evolución de las vulnerabilidades en el sector educativo refleja una tendencia preocupante: los ciberataques son cada vez más sofisticados, y las instituciones deben fortalecer sus capacidades de análisis mediante herramientas de evaluación automatizadas y métricas estandarizadas, como el Common Vulnerability Scoring System (CVSS), que faciliten la priorización de riesgos y la toma de decisiones preventivas.

1.1.2 Avances en metodologías de pentesting para educación

En los últimos años, las metodologías de pentesting (pruebas de penetración) han evolucionado significativamente, adaptándose a las necesidades de los entornos educativos, donde la protección de la información académica, personal y administrativa es prioritaria. Estas metodologías han pasado de enfoques manuales y generales a procesos estructurados, automatizados y éticamente regulados, diseñados específicamente para sistemas como las plataformas de gestión del aprendizaje (LMS), entre ellas Moodle.

El Open Web Application Security Project (OWASP) ha sido una referencia clave en esta evolución, al proponer guías como la *OWASP Testing Guide (2021)*, que establece procedimientos sistemáticos para la identificación de vulnerabilidades en aplicaciones web mediante pruebas pasivas y activas.

Las pruebas pasivas se centran en la recolección de información y el análisis de configuraciones sin alterar el sistema (por ejemplo, identificación de versiones, dominios o encabezados HTTP), mientras que las pruebas activas implican la simulación controlada de ataques como inyecciones SQL, fuerza bruta o explotación de autenticaciones débiles.

En el ámbito educativo, las metodologías actuales de pentesting incorporan también el uso de herramientas automatizadas que mejoran la precisión y reducen el tiempo de evaluación. Entre las más relevantes se destacan:

- **OWASP ZAP (Zed Attack Proxy):** Permite realizar escaneos de vulnerabilidades en sitios web institucionales, detectar fallos de configuración, y evaluar formularios y sesiones de usuarios.
- **DeepExploit:** Combina técnicas de inteligencia artificial y *machine learning* para automatizar la búsqueda y explotación de vulnerabilidades, aprendiendo del entorno analizado.

Numerosos estudios previos han evaluado herramientas como OWASP ZAP y Nessus en entornos educativos y empresariales. Zogaj et al. (2025) realizaron un análisis cuantitativo de 67 aplicaciones web, comparando Nessus, Acunetix, OWASP ZAP y BeSECURE, encontrando que ZAP exhibe capacidades de detección superiores y consistencia en varios niveles de severidad, mientras que Nessus sobresale en identificar vulnerabilidades críticas y de alta severidad. En el contexto universitario, Wenny y Pamuji (2024) compararon Nessus y OWASP ZAP en un sistema de información de personal, demostrando que ZAP identificó vulnerabilidades críticas de aplicación web como ausencia de tokens Anti-CSRF, falta de cabeceras CSP y Anti-Clickjacking, mientras que Nessus se enfocó en vulnerabilidades de infraestructura de red y servidor. Nandi (2024), mediante una revisión sistemática de literatura de 34 estudios, identificó que OWASP ZAP, Nessus y Burp Suite son las herramientas más utilizadas, destacando que el enfoque DAST permite detección de vulnerabilidades en tiempo real y se integra con pipelines CI/CD y enfoques de machine learning. Romero y Morocho (2024), en un estudio ecuatoriano, evaluaron OWASP ZAP, Nessus, OpenVAS y SonarQube en entornos Google Cloud Platform, utilizando aplicaciones vulnerables para simular situaciones reales.

La principal limitación de estos estudios previos es que se limitan a la detección de vulnerabilidades, generando reportes que requieren validación manual posterior. La incorporación de inteligencia artificial mediante herramientas como DeepExploit representa una ventaja incremental significativa, ya que no solo detecta, sino que valida la explotabilidad real de las vulnerabilidades, reduciendo falsos positivos y proporcionando evidencia concreta del riesgo. Esta investigación introduce por primera vez en el contexto universitario ecuatoriano la combinación de detección tradicional (OWASP ZAP) con validación automatizada mediante IA (DeepExploit), superando las limitaciones de los enfoques previos. Estas herramientas, aplicadas bajo un enfoque ético y controlado, resultan particularmente útiles para los departamentos de Tecnologías de la Información (TI) o los responsables de

plataformas virtuales, ya que permiten simular ataques reales sin comprometer la integridad del sistema.

Por otro lado, se ha fortalecido la integración de metodologías como MITRE ATT&CK y CVSS (Common Vulnerability Scoring System), que proporcionan marcos estandarizados para clasificar la severidad de las vulnerabilidades y priorizar acciones correctivas. En el contexto universitario, esto ha permitido establecer protocolos de ciberseguridad basados en evidencia, donde se evalúan de manera continua las aplicaciones, redes y servidores que sostienen las plataformas educativas.

En síntesis, los avances en las metodologías de pentesting para educación no solo optimizan la detección de vulnerabilidades, sino que fomentan una cultura de prevención y mejora continua en la gestión de la seguridad digital institucional.

1.1.3 Herramientas automatizadas de pentesting

Las herramientas automatizadas de pentesting han cobrado creciente relevancia en la evaluación de seguridad de entornos educativos debido a su capacidad para acelerar la detección de vulnerabilidades y cubrir amplias superficies de ataque sin requerir intervención humana constante. En el ámbito de la seguridad de aplicaciones web, existen diversas herramientas que ofrecen funcionalidades de escaneo, análisis y explotación, entre las que destacan OWASP ZAP, Burp Suite, Nikto, Acunetix y DeepExploit (OWASP Foundation, 2021; Choudhary et al., 2022).

La *Tabla 1* presenta una comparación entre herramientas de pentesting ampliamente utilizadas, considerando criterios relevantes para el contexto académico, como el tipo de licencia, el nivel de automatización, el enfoque principal, la complejidad de uso y su aplicabilidad en plataformas educativas como Moodle. Esta comparación permite identificar las fortalezas y limitaciones de cada herramienta y fundamentar la selección de las soluciones empleadas en la presente investigación.

Tabla 1: Matriz comparativa de herramientas automatizadas de pentesting para aplicaciones web

Criterio	**OWASP ZAP**	**DeepExploit**	**Burp Suite**	**Nessus**	**Nikto**
Tipo de herramienta	Análisis web automatizado	Explotación con IA	Análisis web manual	Infraestructura	Escáner web
Enfoque principal	OWASP Top 10	Validación de explotación	Pruebas manuales	Vulnerabilidades conocidas	Fallas básicas
Nivel de automatización	Alta	Muy alta	Media	Alta	Alta
Uso de inteligencia artificial	No	Sí	No	No	No
Análisis pasivo	Sí	No	Sí	No	No
Análisis activo	Sí	Sí	Sí	Sí	Sí
Orientación a aplicaciones web	Alta	Media–Alta	Alta	Baja	Media
Licencia	Código abierto	Código abierto	Comercial	Comercial	Código abierto
Adecuación para Moodle	Muy alta	Complementaria	Alta	Baja	Media
Rol en la tesis	Detección de vulnerabilidades	Validación de explotación	No seleccionada	No alineada al alcance	Insuficiente

Fuente: Elaboración propia.

De acuerdo con el análisis comparativo, OWASP ZAP destaca por ser una herramienta de código abierto, con una curva de aprendizaje accesible y amplias capacidades de análisis pasivo y activo orientadas a vulnerabilidades del OWASP Top 10. Su proxy de interceptación, sus escáneres automatizados y la generación de reportes estructurados la convierten en una

solución adecuada para evaluar aplicaciones web en entornos universitarios, donde se requiere minimizar el impacto operativo y facilitar la interpretación de resultados por parte del personal de TI (OWASP Foundation, 2021; Patel & Desai, 2020).

Por su parte, DeepExploit introduce un enfoque innovador al incorporar técnicas de inteligencia artificial y aprendizaje automático para automatizar la fase de explotación. A diferencia de herramientas tradicionales que se limitan a la detección, DeepExploit permite validar de forma controlada si las vulnerabilidades identificadas son efectivamente explotables, priorizando ataques con mayor probabilidad de éxito y reduciendo intentos redundantes (Choudhary et al., 2022). Esta característica resulta especialmente útil para evaluar el impacto real de vulnerabilidades críticas en sistemas complejos.

La selección de OWASP ZAP y DeepExploit responde a un enfoque complementario: mientras ZAP permite realizar un análisis amplio y sistemático de vulnerabilidades comunes y configuraciones inseguras en la plataforma Moodle, DeepExploit fortalece la evaluación al automatizar la validación de explotación de dichas vulnerabilidades. En conjunto, ambas herramientas proporcionan una visión más completa del estado de seguridad, combinando detección, análisis y verificación de impacto, sin depender exclusivamente de pruebas manuales.

En el contexto de la PUCE-Ibarra, esta combinación resulta pertinente debido a la necesidad de evaluar aplicaciones web educativas de forma eficiente, controlada y ética, garantizando la protección de datos sensibles y la continuidad del servicio. Todas las herramientas automatizadas utilizadas se aplican bajo autorización institucional y dentro de un alcance definido, siguiendo buenas prácticas de pentesting ético y priorizando la clasificación

1.1.4 Vacío de investigación identificado

A pesar de los avances en metodologías y herramientas de pentesting aplicadas a entornos educativos, existe un vacío significativo en la investigación enfocada específicamente en la evaluación de seguridad de plataformas virtuales utilizadas por universidades ecuatorianas, particularmente en lo relacionado con la detección y priorización de vulnerabilidades mediante herramientas automatizadas.

La mayoría de los estudios recientes se han centrado en instituciones internacionales o en aplicaciones web genéricas, dejando un espacio poco explorado en la implementación de pruebas éticas de seguridad sobre plataformas Moodle que son ampliamente utilizadas en la gestión académica nacional. En el caso de la Pontificia Universidad Católica del Ecuador, sede Ibarra, aunque se han adoptado medidas de digitalización y virtualización de servicios, no se han desarrollado evaluaciones sistemáticas documentadas que integren herramientas como OWASP ZAP y DeepExploit para analizar la robustez de sus sistemas de aprendizaje en línea. Además, si bien existen estudios comparativos entre ZAP y Nessus (Zogaj et al., 2025; Wenny & Pamuji, 2024), ninguno incorpora herramientas basadas en inteligencia artificial como DeepExploit para la validación automatizada de vulnerabilidades, limitándose a la fase de detección y dejando la confirmación de explotabilidad como un proceso manual posterior.

Asimismo, la literatura académica carece de propuestas metodológicas que combinen el uso de métricas estandarizadas como CVSS con herramientas automatizadas de análisis de vulnerabilidades, lo que limita la capacidad de priorizar riesgos de manera objetiva y adaptada al entorno institucional. Este vacío metodológico dificulta la generación de estrategias de mitigación basadas en evidencias cuantificables.

Por otro lado, no se identifican investigaciones locales que aborden la formación o el fortalecimiento de capacidades en el personal de tecnologías de la información para la aplicación práctica del pentesting ético en universidades, lo que evidencia una brecha entre la teoría y la implementación operativa de la ciberseguridad educativa.

Por tanto, el presente estudio busca contribuir al cierre de este vacío mediante la aplicación controlada de pruebas de penetración en la plataforma Moodle de la PUCE-Ibarra, utilizando OWASP ZAP y DeepExploit como herramientas complementarias, y evaluando los resultados mediante las métricas del CVSS. Con ello, se pretende generar una guía práctica y replicable que sirva de referencia para fortalecer la seguridad digital en el entorno académico ecuatoriano.

1.2 Marco Teórico

El marco teórico constituye la base conceptual que sustenta esta investigación, permitiendo comprender los principios, técnicas y metodologías que intervienen en la evaluación de vulnerabilidades y pruebas de penetración (pentesting) aplicadas en instituciones educativas.

En este contexto, se abordan conceptos relacionados con la seguridad informática, la ciberseguridad universitaria, el proceso de pentesting y la automatización de pruebas, todos fundamentales para el desarrollo de un modelo adaptado al entorno académico.

La seguridad informática se define como el conjunto de políticas, procesos y herramientas diseñadas para proteger la confidencialidad, integridad y disponibilidad de la información (ISO/IEC 27001, 2022). En el contexto universitario, esta seguridad resulta esencial, ya que las instituciones manejan grandes volúmenes de datos personales, financieros y académicos, los cuales pueden ser vulnerables ante accesos no autorizados, ataques de denegación de servicio o filtraciones de información (Cárdenas & León, 2021).

En este sentido, el pentesting o prueba de penetración se presenta como una estrategia preventiva que busca identificar fallas antes de que puedan ser explotadas por agentes maliciosos. Según Castillo y Méndez (2020), el pentesting es un proceso estructurado en fases que permite simular ataques controlados para evaluar la resistencia de un sistema ante amenazas reales, fortaleciendo así su postura de ciberseguridad.

Con el avance de la tecnología, la tendencia actual se orienta hacia el pentesting automatizado, que emplea herramientas capaces de detectar vulnerabilidades de manera más rápida y sistemática, reduciendo la intervención humana (Patil & Kulkarni, 2021). Plataformas como OpenVAS, Nessus o Metasploit Framework se han convertido en referentes en esta área, ya que permiten ejecutar análisis completos sobre redes, servidores o aplicaciones sin requerir conocimientos avanzados de programación.

No obstante, en el ámbito educativo, el uso de estas herramientas aún es limitado. Diversos autores señalan que las universidades, especialmente en América Latina, carecen de metodologías estandarizadas para la evaluación continua de su infraestructura digital (Rodríguez et al., 2023). Por ello, surge la necesidad de diseñar modelos accesibles y adaptables que integren la automatización del pentesting con la gestión institucional, permitiendo fortalecer la protección de los entornos digitales académicos.

1.2.1 Fundamentos de seguridad en aplicaciones web

La seguridad en aplicaciones web constituye uno de los pilares fundamentales dentro de la ciberseguridad moderna, ya que la mayoría de los servicios institucionales, incluyendo los académicos, se ejecutan actualmente en plataformas accesibles desde internet. De acuerdo con Stallings (2021), la seguridad en aplicaciones web busca proteger los datos, usuarios y procesos de los sistemas frente a accesos no autorizados o manipulaciones indebidas, garantizando la confidencialidad, integridad y disponibilidad de la información.

En el contexto universitario, las plataformas web como sistemas de gestión académica, bibliotecas digitales o entornos virtuales de aprendizaje (como Moodle o Canvas) almacenan grandes volúmenes de información sensible. Por tanto, su protección se vuelve prioritaria. Según López y Paredes (2022), las vulnerabilidades más frecuentes en estos sistemas surgen de una deficiente validación de entradas, una mala gestión de sesiones y una configuración insegura de servidores o bases de datos.

La seguridad de las aplicaciones web se fundamenta en tres principios esenciales:

- Autenticación: garantizar que solo los usuarios autorizados puedan acceder a los recursos del sistema.
- Autorización: definir los permisos y privilegios de cada usuario dentro de la aplicación.
- Cifrado: proteger la información transmitida entre el cliente y el servidor mediante algoritmos criptográficos (por ejemplo, TLS/SSL).

Asimismo, el diseño seguro de una aplicación web implica aplicar metodologías como el Secure Development Lifecycle (SDLC), que incorpora controles de seguridad desde la fase de diseño hasta la implementación y mantenimiento del software (Microsoft, 2020). En el ámbito educativo, la aplicación de estos fundamentos permite reducir la exposición a ataques cibernéticos, garantizando la continuidad de los servicios académicos y la protección de la información institucional.

1.2.2 Teoría de vulnerabilidades web (OWASP Top 10 -2021)

La teoría de vulnerabilidades web se basa en la identificación, análisis y mitigación de debilidades que pueden ser aprovechadas por atacantes para comprometer una aplicación. La

organización OWASP (Open Web Application Security Project) publica periódicamente el listado OWASP Top 10, que recopila las vulnerabilidades más críticas observadas en aplicaciones web a nivel mundial.

La versión OWASP Top 10 - 2021 introduce un enfoque más amplio, considerando no solo fallas técnicas, sino también problemas de diseño y gestión. Según OWASP (2021), las diez vulnerabilidades principales son:

1. **Broken Access Control (Control de acceso roto):** los usuarios pueden acceder a recursos sin los permisos adecuados.
2. **Cryptographic Failures (Fallos criptográficos):** uso incorrecto o ausencia de cifrado, exponiendo datos sensibles.
3. **Injection (Inyección):** inserción de código malicioso, como SQL o comandos del sistema.
4. **Insecure Design (Diseño inseguro):** ausencia de principios de seguridad en la arquitectura del sistema.
5. **Security Misconfiguration (Configuración insegura):** errores en la configuración del servidor o del software.
6. **Vulnerable and Outdated Components (Componentes vulnerables o desactualizados):** uso de bibliotecas o frameworks obsoletos.
7. **Identification and Authentication Failures (Fallas en autenticación e identificación):** debilidades en contraseñas o gestión de sesiones.
8. **Software and Data Integrity Failures (Fallas de integridad en software y datos):** alteración o manipulación de información.
9. **Security Logging and Monitoring Failures (Deficiencias en registro y monitoreo):** falta de trazabilidad ante incidentes.
10. **Server-Side Request Forgery (SSRF):** manipulación de solicitudes desde el servidor hacia otros sistemas.

Estas vulnerabilidades constituyen la base teórica del análisis en las pruebas de penetración, ya que permiten establecer un marco de referencia universal para evaluar la seguridad de aplicaciones web. De acuerdo con Torres y Ramírez (2023), el uso del OWASP Top 10 en entornos educativos facilita la priorización de riesgos, ayudando a las universidades a

identificar los puntos más críticos en sus plataformas digitales y a implementar medidas preventivas basadas en evidencia.

1.2.3 Metodologías de pentesting ético

El pentesting ético es un proceso estructurado y controlado cuyo propósito es identificar, explotar de forma segura y reportar vulnerabilidades para que sean corregidas antes de que un atacante real las aproveche. Más allá de la técnica, el pentesting ético exige un marco de gobernanza que incluya autorización, alcance definido, reglas de compromiso y manejo responsable de los hallazgos.

Este proceso se desarrolla en fases secuenciales y controladas, las cuales se ilustran de forma resumida en la *Figura 1*, permitiendo comprender el flujo general de una prueba de penetración ética aplicada al contexto académico.

Figura 1: Fases de la metodología de pentesting ético

Fuente: Elaboración Propia

Fases del pentesting ético

Las metodologías más aceptadas organizan el trabajo en fases diferenciadas que aseguran rigor técnico y control operativo:

1. **Preparación y autorización:** Antes de cualquier actividad técnica se debe obtener autorización escrita del titular del sistema, definir el alcance (qué hosts, puertos, aplicaciones y datos están dentro o fuera de la prueba), las ventanas de ejecución, y los planes de contingencia (backups, contacto de emergencia). Esto es obligatorio para evitar consecuencias legales y operativas.
2. **Reconocimiento (reconnaissance):** Incluye actividades pasivas y activas de bajo impacto para mapear la superficie de ataque: recolección de información pública, identificación de subdominios, y análisis de cabeceras y certificados. En el caso de la

plataforma Moodle se prioriza la identificación de rutas de acceso, puntos de login y endpoints de la API.

3. **Escaneo y enumeración:** Uso de herramientas para descubrir puertos, servicios y versiones de software (por ejemplo, escaneo de puertos y fingerprinting). Esta fase genera mayor tráfico y por tanto debe planificarse para minimizar riesgos en entornos de producción.
4. **Explotación controlada:** Aplicación de técnicas y exploits para confirmar la existencia de vulnerabilidades (inyección SQL, XSS, bypass de autenticación, etc.). Solo se ejecutan exploits que han sido aprobados en el alcance; en entornos sensibles se prefiere replicar la vulnerabilidad en un entorno de pruebas.
5. **Post-explotación y análisis de impacto:** Evaluación del alcance del compromiso (acceso a datos, impacto en la integridad o disponibilidad), sin realizar acciones que dañen o alteren permanentemente la información.
6. **Reporte y remediación**: Documentación técnica y ejecutiva de hallazgos con evidencia, CVSS para priorización y recomendaciones claras de mitigación y verificación.
7. **Validación y seguimiento:** Re-pruebas después de aplicar mitigaciones para confirmar la efectividad de las correcciones.

Tipos de pruebas según visibilidad (alcance y conocimiento)

- **Black-box**: El pentester no dispone de información interna; simula un atacante externo.
- **Gray-box:** El pentester cuenta con información limitada (por ejemplo, credenciales de prueba); es eficiente para simular ataques dirigidos.
- **White-box:** El pentester dispone de acceso completo (código fuente, arquitectura); permite pruebas profundas y revisión de diseño.

Para la PUCE-Ibarra y la plataforma Moodle, una combinación gray-box (credenciales de prueba de baja privilegiación) y black-box para componentes públicos es práctica: gray-box facilita la evaluación de control de acceso y lógica de negocio; black-box evalúa la exposición desde el exterior.

1.2.4 Plataformas de gestión de aprendizaje (Moodle)

Las plataformas de gestión de aprendizaje (Learning Management Systems, LMS) se han convertido en herramientas esenciales dentro de las instituciones educativas para la administración de contenidos, evaluación y comunicación entre docentes y estudiantes. Moodle, en particular, destaca como una de las plataformas más utilizadas a nivel mundial debido a su naturaleza de código abierto, su flexibilidad y su capacidad de integración con diversos sistemas (Dougiamas & Taylor, 2003).

En el contexto de la Pontificia Universidad Católica del Ecuador (PUCE) sede Ibarra, Moodle desempeña un papel central en los procesos académicos digitales, alojando datos sensibles de estudiantes, docentes y personal administrativo. Estos datos incluyen información personal, calificaciones, materiales académicos y credenciales de acceso, lo que convierte a la plataforma en un objetivo potencial para ciberataques.

Diversos estudios evidencian que las vulnerabilidades en plataformas LMS pueden surgir por configuraciones inadecuadas, plugins desactualizados o fallas en la autenticación de usuarios (Kumar et al., 2021). Según OWASP (2021), los entornos basados en web como Moodle pueden presentar riesgos asociados al “Top 10” de vulnerabilidades comunes, entre ellas: inyección de código, exposición de datos sensibles y control de acceso deficiente.

Para fortalecer la seguridad de Moodle, se recomienda la implementación de auditorías periódicas mediante herramientas de pentesting ético, como OWASP ZAP y DeepExploit. Estas permiten detectar vulnerabilidades en la configuración del servidor web, la base de datos o la red de comunicación, contribuyendo a mantener la integridad y confidencialidad de la información. Asimismo, la actualización constante del sistema y la aplicación de parches de seguridad son prácticas clave para reducir el riesgo de explotación (Patil & Kulkarni, 2021).

En este sentido, la seguridad del LMS no depende únicamente de la plataforma, sino también de la correcta gestión de su infraestructura de TI, que abarca el servidor físico, el sistema operativo, la red, las políticas de seguridad y las operaciones de soporte técnico. Por tanto, el análisis de vulnerabilidades aplicado a la plataforma Moodle de la PUCE Ibarra representa un componente crucial para garantizar la continuidad del aprendizaje digital y la confianza de la comunidad académica.

1.2.5 Herramientas de análisis de seguridad

En la presente investigación se emplearán específicamente OWASP ZAP y DeepExploit como herramientas principales para la identificación y evaluación de vulnerabilidades en la plataforma virtual Moodle de la PUCE-Ibarra. A continuación, se describen sus capacidades, limitaciones y el modo en que se integrarán en la metodología del estudio.

- **OWASP ZAP (Zed Attack Proxy).**
  OWASP ZAP es una herramienta de código abierto diseñada para realizar análisis de seguridad de aplicaciones web mediante técnicas pasivas y activas. ZAP será utilizada para las fases de reconocimiento, mapeo de rutas, escaneo automático de vulnerabilidades comunes (OWASP Top 10), análisis de parámetros y pruebas de configuración (cabeceras HTTP, TLS/SSL, cookies, etc.). Entre sus funcionalidades relevantes para este trabajo destacan el proxy de interceptación (para inspeccionar y modificar tráfico HTTP/HTTPS), el crawler para descubrir endpoints y parámetros, los escáneres automáticos y la capacidad de generar reportes exportables que luego se clasificarán con CVSS (Patel & Desai, 2020; OWASP Foundation, 2021).
  Limitaciones: OWAS ZAP puede producir falsos positivos que requieren verificación manual; por ello, todos los hallazgos serán validados por el equipo técnico antes de su clasificación final.
- **DeepExploit (automatización con IA).**
  DeepExploit incorpora técnicas de inteligencia artificial para priorizar vectores de ataque y automatizar pasos de explotación que normalmente demandan intervención humana. En este estudio, DeepExploit se empleará como herramienta complementaria a ZAP para automatizar la fase de explotación controlada y evaluar la factibilidad práctica de explotación en vulnerabilidades complejas detectadas por ZAP. DeepExploit normalmente se integra con frameworks de explotación como Metasploit, por lo que su uso permitirá orquestar intentos de explotación automatizados empleando módulos ya probados en Metasploit (Choudhary et al., 2022).
  Limitaciones y controles: dado que la automatización puede incrementar el riesgo operativo, todas las ejecuciones de DeepExploit se harán en ventanas autorizadas y preferentemente en un entorno de pruebas; cualquier explotación en producción será no

destructiva y supervisada. Además, los resultados automatizados serán revisados y confirmados manualmente para evitar falsos positivos y daños involuntarios.

La estrategia práctica será híbrida: utilizar ZAP para el barrido amplio y la detección inicial, y emplear DeepExploit para priorizar y (con autorización) automatizar intentos controlados de explotación sobre los hallazgos más sospechosos. Todas las vulnerabilidades serán clasificadas con CVSS (métrica Base y Temporal) para priorizar remediaciones. Los hallazgos críticos serán notificados primero al responsable de TI de la PUCE-Ibarra y se documentará evidencia técnica (capturas, logs y trazas) para asegurar trazabilidad y remediación.

1.2.6 Marco legal y ético para el pentesting en Ecuador

El pentesting ético en Ecuador debe realizarse bajo un marco normativo que garantice la legalidad, confidencialidad y consentimiento expreso por parte de la institución evaluada. Este tipo de pruebas, aunque técnicas, implican el acceso a información sensible, por lo que deben regirse por principios éticos y regulaciones vigentes en materia de protección de datos, delitos informáticos y responsabilidad profesional.

En el contexto ecuatoriano, las actividades relacionadas con la ciberseguridad y la auditoría técnica se sustentan principalmente en tres instrumentos normativos:

1. **Código Orgánico Integral Penal (COIP):** establece en su artículo 232 y subsiguientes que el acceso no autorizado, manipulación o interferencia en sistemas informáticos constituye un delito informático sancionable. Por ello, cualquier prueba de penetración debe realizarse únicamente con autorización escrita del titular o responsable de la infraestructura tecnológica, evitando así vulnerar la ley (Asamblea Nacional, 2021).
2. **Ley Orgánica de Protección de Datos Personales (LOPDP):** promulgada en 2021, esta ley regula el tratamiento legítimo, controlado y transparente de datos personales. En el contexto del pentesting, exige la protección de la identidad y los datos personales que pudieran ser expuestos durante las pruebas, aplicando principios de minimización y confidencialidad (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2021).

3. **Normas ISO/IEC 27001 y 27002:** aunque no son leyes, establecen estándares internacionales adoptados por instituciones ecuatorianas para la gestión de la seguridad de la información. Estas normas recomiendan la planificación formal de pruebas de seguridad, la delimitación del alcance, la autorización escrita y la documentación detallada de los hallazgos (ISO, 2022).

Desde el punto de vista ético, el pentesting debe orientarse al mejoramiento de la seguridad institucional, evitando cualquier daño al sistema, pérdida de información o afectación al servicio. Para ello, se deben seguir los siguientes principios:

- **Autorización y consentimiento informado:** Todas las pruebas deben ser aprobadas por los responsables de TI y autoridades académicas.
- **Integridad y confidencialidad:** Los resultados del análisis no deben divulgarse fuera del equipo autorizado.
- **No intrusión destructiva:** Los ataques simulados deben realizarse sin modificar datos, configuraciones ni afectar la disponibilidad de los sistemas.
- **Responsabilidad profesional:** El analista debe actuar conforme a las buenas prácticas definidas por la EC-Council (CEH) y OWASP Ethical Testing Guidelines, priorizando siempre la seguridad institucional sobre los fines experimentales.

En este proyecto, las pruebas de seguridad se realizarán exclusivamente sobre la plataforma Moodle institucional, bajo la autorización formal de la PUCE-Ibarra.

El proceso se desarrollará conforme a un acuerdo de alcance definido, respetando las fases de identificación, análisis y explotación controlada, garantizando la protección de los datos personales, académicos y administrativos.

Además, se implementarán medidas de respaldo y monitoreo continuo para prevenir cualquier interrupción del servicio educativo durante las pruebas.

CAPÍTULO II: MATERIALES Y MÉTODOS

3.1 Generalidades de la investigación

La presente investigación se enmarca en el área de la seguridad informática, enfocándose en el análisis de vulnerabilidades mediante técnicas de pentesting automatizado aplicadas al Campus Virtual de la Pontificia Universidad Católica del Ecuador – Sede Ibarra. Se define como un estudio de tipo aplicada y de enfoque cualitativo, ya que se orienta a identificar y caracterizar vulnerabilidades reales para proponer medidas de mitigación que fortalezcan la seguridad institucional, analizando el impacto y la criticidad de los riesgos más allá de su cuantificación numérica. Todo el proceso se desarrolla en un entorno controlado y debidamente autorizado, empleando herramientas especializadas como OWASP ZAP y DeepExploit, lo que garantiza la ejecución de las pruebas sin afectar la operación normal del sistema ni comprometer la información real de los usuarios.

3.1.1 Alcance de la investigación

El presente estudio se centra en el análisis de vulnerabilidades de la plataforma Moodle del Campus Virtual de la Pontificia Universidad Católica del Ecuador, Sede Ibarra. La investigación se desarrolla exclusivamente en un entorno controlado y autorizado, diseñado para replicar fielmente las condiciones del sistema en producción sin interferir con la operatividad institucional.

El alcance técnico comprende la identificación, detección y análisis de riesgos asociados a las vulnerabilidades del Campus Virtual de la Pontificia Universidad Católica del Ecuador, Sede Ibarra. Para ello, se emplean técnicas de pentesting automatizado mediante las herramientas OWASP ZAP y DeepExploit. El estudio se limita estrictamente a la fase de diagnóstico, evitando la ejecución de ataques explotativos o destructivos que puedan comprometer la integridad o disponibilidad de la infraestructura institucional.

Finalmente, esta investigación tiene un carácter diagnóstico y análisis; por lo tanto, no contempla la implementación física de medidas correctivas. Su propósito es determinar la probabilidad d, el impacto, el riesgo, la severidad y priorización que sirva de base para

determinar el grado de seguridad, orientadas a fortalecer la postura de seguridad informática del Campus Virtual.

3.1.2 Unidad de análisis (Plataforma Moodle del Campus Virtual)

La unidad de análisis de esta investigación corresponde a la plataforma Moodle utilizada como Campus Virtual de la Pontificia Universidad Católica del Ecuador, sede Ibarra. Este sistema constituye un componente fundamental del entorno académico institucional, ya que gestiona procesos de enseñanza–aprendizaje, acceso de usuarios, contenidos educativos y datos académicos sensibles.

El análisis se centra en los componentes del campus virtual que está en la plataforma Moodle que se monta sobre el sistema operativo, desplegada en el servidor el que almacena sobre la base de dato. La evaluación de la plataforma permite identificar debilidades que podrían ser explotadas por atacantes, afectando la confidencialidad, integridad y disponibilidad de la información gestionada por el sistema.

A partir del estudio de esta unidad de análisis, se busca determinar el nivel de exposición al riesgo del Campus Virtual y establecer una base técnica para la clasificación y priorización de vulnerabilidades, de acuerdo con estándares reconocidos en seguridad informática.

3.2 Entorno técnico y escenario de pruebas

El entorno de pruebas de la presente investigación se implementó mediante un laboratorio virtualizado híbrido, combinando el uso del hipervisor empresarial VMware ESXi y el entorno de virtualización local Oracle VM VirtualBox. Esta arquitectura permite simular un escenario de ataques bajo un entorno controlado, similar a la infraestructura que enfrenta un entorno de producción universitario.

El servidor objetivo, correspondiente a la plataforma Moodle del Campus Virtual, fue desplegado en una máquina virtual alojada en ESXi, replicando la infraestructura empresarial institucional. Por su parte, la estación de auditoría fue implementada sobre Kali Linux ejecutado en VirtualBox, integrándose ambos entornos mediante una configuración de red que

garantiza la comunicación directa para las fases de reconocimiento y explotación sin afectar sistemas reales.

3.2.1 Arquitectura del laboratorio virtualizado (ESXi y Virtual Box)

La arquitectura se estructuró de manera distribuida para maximizar el control del tráfico. El servidor objetivo en ESXi utiliza una dirección IP interna asignada (192.168.0.178), donde se configuró el directorio /pruebasvirtual/ para la ejecución controlada de los escaneos de seguridad.

La máquina atacante, basada en Kali Linux, fue desplegada en Oracle VirtualBox y configurada con un adaptador de red en modo puente (Bridge) o red interna, permitiendo el acceso directo al servidor alojado en ESXi. Esta configuración posibilitó que herramientas como OWASP ZAP, Metasploit y DeepExploit pudieran detectar, analizar y explotar vulnerabilidades de forma controlada.

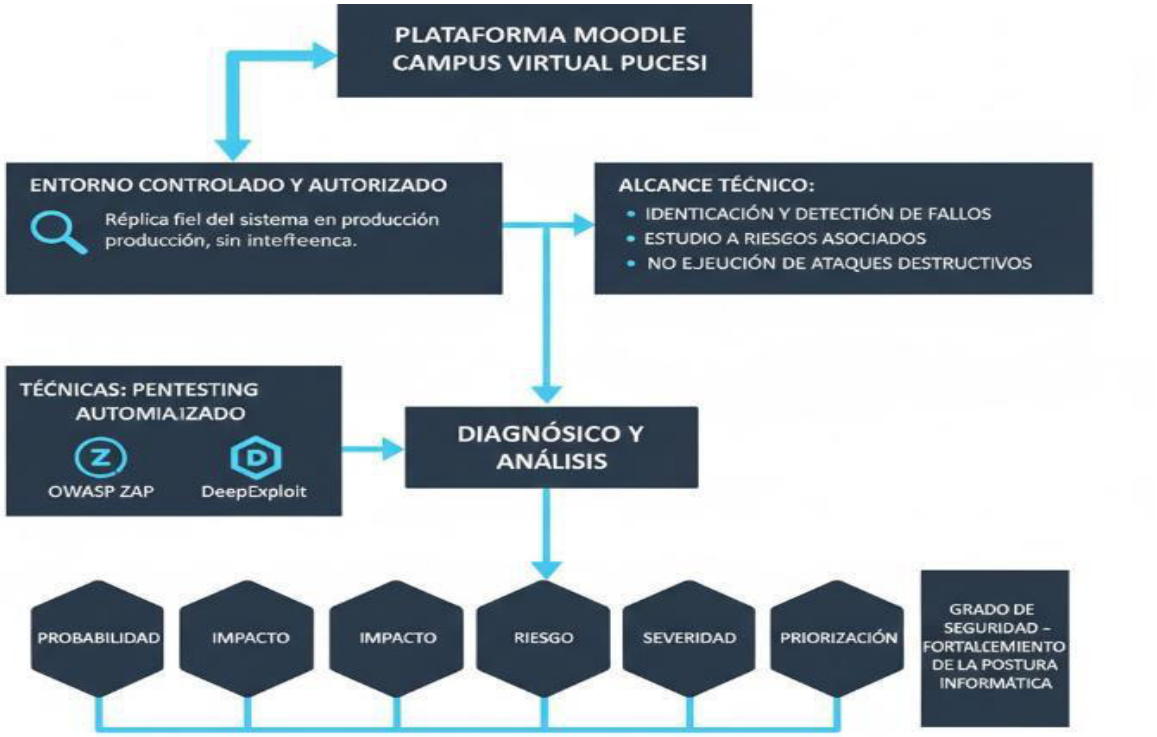
La interconexión entre ambos entornos permitió simular un escenario real de ataque externo hacia una infraestructura empresarial, reforzando la validez práctica y académica de los resultados obtenidos en la investigación.

3.2.2 Configuración de la estación de ataque y nodo de explotación

El texto describe el alcance de una investigación enfocada en la seguridad de la plataforma Moodle del Campus Virtual de la Pontificia Universidad Católica del Ecuador, Sede Ibarra (PUCESI).

Como se visualiza en el diagrama de flujo de la investigación Figura.2, el proceso se estructuró en los siguientes componentes y restricciones clave:

Figura 2:Flujo y los componentes clave del estudio



Fuente: Elaboración Propia

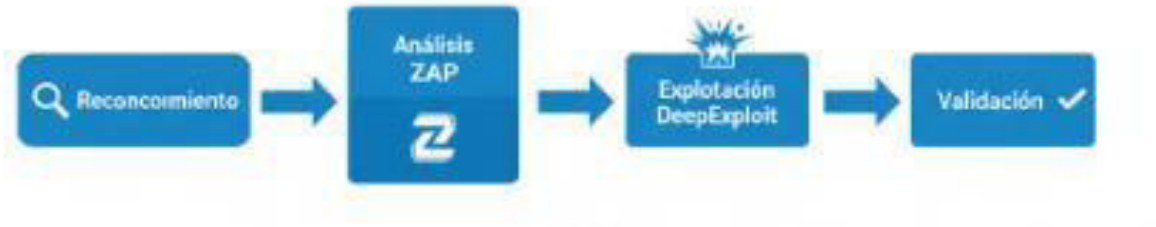
- **Plataforma Moodle Campus Virtual:** Representa el objetivo principal del análisis, la plataforma Moodle de la PUCESI.
- **Entorno Controlado y Autorizado:** Destaca que la investigación se realiza en un ambiente de réplica segura, sin afectar el sistema real.
- **Alcance Técnico:** Detalla los objetivos técnicos: identificación y detección de fallos, estudio de riesgos asociados, y la crucial restricción de NO ejecutar ataques destructivos.
- **Técnicas: Pentesting Automatizado:** Muestra las herramientas específicas utilizadas (OWASP ZAP y DeepExploit) para realizar las pruebas de penetración automatizadas.
- **Carácter: Diagnóstico y Análisis:** Subraya que el estudio es de naturaleza diagnóstica y analítica, enfocándose en determinar probabilidad, impacto, riesgo, severidad y priorización.
- **Grado de Seguridad - Fortalecimiento de la Postura:** Finalmente, indica el propósito último: servir de base para determinar el grado de seguridad y fortalecer la postura de seguridad informática del Campus Virtual.

3.3 Metodología de pentesting automatizado

La metodología aplicada en esta investigación se fundamenta en un enfoque de pentesting automatizado, estructurado en fases secuenciales que permiten identificar, analizar y validar vulnerabilidades de seguridad en la plataforma Moodle del Campus Virtual.

Este enfoque integra técnicas de reconocimiento, análisis dinámico de aplicaciones web (DAST), explotación automatizada mediante inteligencia artificial y validación de hallazgos, garantizando un proceso sistemático, reproducible y alineado con los estándares internacionales de seguridad.

Figura 3: Flujo de trabajo del pentesting automatizado: Reconocimiento, Análisis ZAP, Explotación DeepExploit y Validación.



Fuente: Elaboración Propia

La Figura 3 muestra el flujo secuencial de las actividades, destacando la interacción entre las herramientas seleccionadas y las etapas de diagnóstico y análisis de riesgos.

3.3.1 Fase 1: Reconocimiento y escaneo de puertos

En esta fase se realizó el reconocimiento inicial del entorno objetivo con el fin de identificar los servicios activos, puertos abiertos y tecnologías subyacentes del servidor que aloja la plataforma Moodle. Para ello, se empleó la herramienta Nmap, permitiendo obtener información relevante como versiones de servicios, sistemas operativos y posibles vectores de ataque. Esta etapa es fundamental, ya que proporciona la base técnica necesaria para orientar las fases posteriores del proceso de pentesting.

3.3.2 Fase 2: Análisis de vulnerabilidades web (DAST) con OWASP ZAP

Posteriormente, se llevó a cabo un análisis dinámico de vulnerabilidades web utilizando OWASP ZAP, enfocado en la detección automatizada de fallas de seguridad en la aplicación Moodle en tiempo de ejecución. Esta fase permitió identificar vulnerabilidades comunes como inyecciones, fallos de configuración, exposición de información sensible y problemas de autenticación, las cuales fueron clasificadas de acuerdo con el OWASP Top 10 y evaluadas preliminarmente según su severidad.

3.3.3 Fase 3: Explotación automatizada con DeepExploit

En la tercera fase se utilizó DeepExploit, una herramienta de pentesting automatizado basada en técnicas de aprendizaje por refuerzo, integrada con el framework Metasploit. El objetivo de esta etapa fue evaluar el impacto real de las vulnerabilidades previamente identificadas, permitiendo la ejecución automatizada de exploits en un entorno controlado. Esta fase facilitó la validación práctica de las debilidades detectadas y evidenció el nivel de riesgo asociado a una posible explotación en un escenario real.

3.3.4 Fase 4: Validación y análisis de falsos positivos

Finalmente, se realizó un proceso de validación de los resultados obtenidos con el fin de identificar y descartar posibles falsos positivos generados durante las fases automatizadas. Esta etapa incluyó la revisión manual de evidencias, la correlación entre resultados de OWASP ZAP y DeepExploit, y la confirmación de vulnerabilidades efectivamente explotables. Este análisis permitió garantizar la confiabilidad de los resultados y fortalecer la precisión del análisis de riesgos presentado en la investigación.

3.4 Métricas, categorización y análisis de riesgos

Con el fin de evaluar de manera objetiva las vulnerabilidades identificadas durante el proceso de pentesting, en esta investigación se emplearon métricas estandarizadas y metodologías reconocidas en el ámbito de la seguridad informática. Estas permiten no solo medir la severidad técnica de cada vulnerabilidad, sino también analizar su impacto sobre los activos críticos del Campus Virtual. Para ello, se integraron el sistema de puntuación CVSS v3.1, el modelo de

seguridad basado en la tríada CIA, la clasificación OWASP Top 10 y una metodología de análisis de riesgos basada en probabilidad e impacto.

3.4.1 Sistema de puntuación CVSS v3.1

El sistema Common Vulnerability Scoring System (CVSS) versión 3.1 fue utilizado como métrica principal para cuantificar la severidad de las vulnerabilidades detectadas. Este estándar internacional proporciona una puntuación numérica que oscila entre 0.0 y 10.0, permitiendo clasificar las vulnerabilidades en niveles de severidad baja, media, alta y crítica. La puntuación CVSS se calcula considerando métricas base como el vector de ataque, la complejidad, los privilegios requeridos y el impacto sobre la confidencialidad, integridad y disponibilidad del sistema. El uso de CVSS v3.1 garantiza consistencia, comparabilidad y respaldo técnico en la evaluación de riesgos, tal como se detalla en la Tabla 2.

Tabla 2:Clasificación de niveles de severidad según CVSS v3.1

Rango CVSS	Nivel de severidad
0.0 – 3.9	Baja
4.0 – 6.9	Media
7.0 – 8.9	Alta
9.0 – 10.0	Crítica

Nota. Adaptado del estándar Common Vulnerability Scoring System v3.1.

Esta categorización constituye la base para la priorización de las acciones correctivas en el presente estudio. Al segregar los hallazgos según su nivel de severidad, se facilita la identificación inmediata de aquellas vulnerabilidades críticas y altas que requieren una intervención urgente para garantizar la integridad del Campus Virtual, permitiendo así una gestión de riesgos eficiente y objetiva.

3.4.2 Dimensiones del análisis de seguridad (Confidencialidad, Integridad y Disponibilidad – CIA)

El análisis de las vulnerabilidades se fundamentó en la tríada de seguridad Confidencialidad, Integridad y Disponibilidad (CIA), ampliamente aceptada como modelo base en la gestión de

la seguridad de la información. La confidencialidad evalúa el riesgo de acceso no autorizado a información sensible del Campus Virtual; la integridad analiza la posibilidad de alteración indebida de datos académicos y administrativos; y la disponibilidad considera el impacto de ataques que puedan afectar la continuidad del servicio educativo. Cada vulnerabilidad fue analizada según su afectación a una o más de estas dimensiones, permitiendo una comprensión integral del riesgo.

Tabla 3: Dimensiones de la seguridad de la información (Tríada CIA)

Dimensión	Descripción
Confidencialidad	Protección de la información frente a accesos no autorizados
Integridad	Garantía de que los datos no sean alterados sin autorización
Disponibilidad	Asegurar el acceso continuo a los servicios del Sistema

Fuente: Elaboración Propia

Según la *Tabla 3*, la categorización de los riesgos bajo estos tres pilares permite delimitar con precisión el alcance del daño potencial de cada vulnerabilidad. Esta distinción fue fundamental para el análisis posterior, ya que facilitó priorizar aquellos fallos que comprometían directamente la operatividad crítica (disponibilidad) o la validez de los registros académicos (integridad) frente a riesgos de menor impacto operativo.

3.4.3 Categorías de vulnerabilidades según OWASP Top 10

Para la categorización de las vulnerabilidades identificadas en la plataforma Moodle, se utilizó la clasificación establecida por el OWASP Top 10, la cual agrupa los riesgos de seguridad web más críticos a nivel mundial. Esta clasificación facilita la identificación de patrones comunes de vulnerabilidades, tales como fallas de control de acceso, inyección, exposición de datos sensibles y configuraciones de seguridad incorrectas. La adopción del OWASP Top 10 permite alinear los resultados del estudio con estándares internacionales y aporta claridad en la presentación de los hallazgos a nivel académico y técnico.

Tabla 4: Tabla de Clasificación OWASP

Códio de OWASP	Categoría
A01	Broken Access Control

A02	Cryptographic Failures
A03	Injection
A04	Insecure Design
A05	Security Misconfiguration
A06	Vulnerable and Outdated Components
A07	Identification and Authentication Failures
A08	Software and Data Integrity Failures
A09	Security Logging and Monitoring Failures
A10	Server-Side Request Fo

Fuente: Adaptado de OWASP Foundation (2021).

3.4.4 Metodología de análisis de riesgos (Probabilidad vs Impacto)

El análisis de riesgos se realizó mediante una metodología cualitativa basada en la relación entre la probabilidad de ocurrencia y el impacto potencial de cada vulnerabilidad. La probabilidad considera factores como la facilidad de explotación y la existencia de herramientas automatizadas, mientras que el impacto evalúa las consecuencias sobre la operación del Campus Virtual, la información institucional y los usuarios finales. Esta metodología permite priorizar las vulnerabilidades no solo por su severidad técnica, sino por el riesgo real que representan en un entorno productivo.

Tabla 5: Matriz de criterios para el análisis de riesgo

Nivel	Probabilidad	Impacto
Bajo	Difícil de explotar	Impacto mínimo
Medio	Explotación moderada	Impacto parcial
Alto	Fácil de explotar	Impacto significativo
Crítico	Automatizable	Impacto severo

Fuente: Elaboración Propia

Con base en la *Tabla 5,* se estableció una matriz de decisión que permite filtrar los hallazgos técnicos. De este modo, el estudio se enfoca prioritariamente en aquellas vulnerabilidades donde la facilidad de explotación (Probabilidad Alta/Crítica) converge con consecuencias graves para el sistema (Impacto Significativo/Severo), asegurando que los esfuerzos de mitigación se dirijan a las amenazas más urgentes.

3.4.5 Matriz de clasificación de severidad y priorización

Como resultado del análisis de probabilidad e impacto, se construyó una matriz de clasificación de severidad, la cual permite visualizar y priorizar las vulnerabilidades detectadas. Esta matriz clasifica los riesgos en niveles bajo, medio, alto y crítico, facilitando la toma de decisiones respecto a la mitigación y tratamiento de vulnerabilidades. La matriz constituye una herramienta clave para la gestión del riesgo, ya que orienta la asignación de recursos y la implementación de controles de seguridad de manera eficiente y justificada.

Tabla 6:Matriz de criticidad: Probabilidad vs. Impacto

Impacto \ Probabilidad	Baja	Media	Alta
Bajo	Bajo	Bajo	Medio
Medio	Bajo	Medio	Alto
Alto	Medio	Alto	Crítico

Fuente: Elaboración Propia

Como se observa en la *Tabla 6*, la intersección de estas variables define la urgencia de la respuesta. Esta clasificación es determinante para el plan de acción, estableciendo que cualquier hallazgo ubicado en los cuadrantes Alto y Crítico debe ser atendido con prioridad inmediata, mientras que los riesgos clasificados como Bajos o Medios pueden ser gestionados mediante monitoreo o remediación planificada a mediano plazo.

3.5 Procedimiento general del estudio

El procedimiento general del presente estudio se desarrolló bajo un enfoque sistemático y controlado, con el objetivo de identificar, analizar y evaluar vulnerabilidades de seguridad en

el Campus Virtual de la Pontificia Universidad Católica del Ecuador – Ibarra, utilizando técnicas de pentesting automatizado. Todas las pruebas se ejecutaron en un entorno autorizado y controlado, respetando principios éticos y legales previamente establecidos.

En una primera etapa, se diseñó y configuró el laboratorio virtualizado, el cual estuvo compuesto por una máquina atacante con sistema operativo Kali Linux y un servidor objetivo que aloja la plataforma Moodle del Campus Virtual. Dicho entorno fue implementado sobre infraestructura virtualizada, permitiendo la simulación de un escenario empresarial real sin afectar los servicios en producción de la institución.

Posteriormente, se realizó la fase de reconocimiento del sistema objetivo, donde se identificaron direcciones IP activas, servicios expuestos y puertos abiertos mediante herramientas de escaneo de red. Esta etapa permitió obtener información preliminar necesaria para orientar las fases posteriores del proceso de pentesting.

A continuación, se ejecutó el análisis dinámico de vulnerabilidades web mediante la herramienta OWASP ZAP, enfocándose en el directorio autorizado del Campus Virtual. Como resultado de esta fase, se generaron reportes automáticos que detallan posibles fallos de seguridad clasificados según su nivel de severidad y categoría OWASP.

Una vez identificadas las vulnerabilidades, se procedió a la fase de explotación automatizada utilizando la herramienta DeepExploit, la cual integra técnicas de aprendizaje por refuerzo sobre el framework Metasploit. Esta etapa tuvo como finalidad validar la explotabilidad real de las vulnerabilidades detectadas previamente, evitando falsos positivos y priorizando aquellas con mayor impacto potencial.

Finalmente, los resultados obtenidos fueron analizados y clasificados mediante métricas estandarizadas como CVSS v3.1, así como mediante un análisis de riesgos basado en la probabilidad de explotación y el impacto sobre las dimensiones de confidencialidad, integridad y disponibilidad. Este proceso permitió establecer una priorización de vulnerabilidades y generar conclusiones orientadas a la mejora de la seguridad del Campus Virtual.

CAPÍTULO III: RESULTADOS

4.1 Resultados del análisis de vulnerabilidades con OWASP ZAP

El análisis de vulnerabilidades del Campus Virtual de la Pontificia Universidad Católica del Ecuador – Ibarra se realizó mediante la herramienta OWASP ZAP, empleando un enfoque de análisis dinámico de seguridad de aplicaciones web (DAST). Este proceso permitió identificar vulnerabilidades presentes en la plataforma Moodle a partir de la interacción directa con la aplicación en ejecución, sin necesidad de acceso al código fuente.

Tabla 7:Distribución Base de Alertas

Nivel de Riesgo	**Confirmado por Usuario**	**Alta Confianza**	**Medios de Comunicación**	**Baja Confianza**	**Total, por Riesgo**
Alto	0 (0.0%)	0 (0.0%)	1 (6.2%)	0 (0.0%)	1 (6.2%)
Medio	0 (0.0%)	1 (6.2%)	0 (0.0%)	1 (6.2%)	2 (12.5%)
Bajo	0 (0.0%)	2 (12.5%)	5 (31.2%)	1 (6.2%)	8 (50.0%)
Informativo	0 (0.0%)	1 (6.2%)	2 (12.5%)	2 (12.5%)	5 (31.2%)
Total por Confianza	0 (0.0%)	4 (25.0%)	8 (50.0%)	4 (25.0%)	16 (100%)

Fuente: Elaboración Propia

La *Tabla 7* de distribución base revela un perfil de seguridad con características mixtas. Se observa que el 50% de las alertas son clasificadas como de riesgo bajo, lo que sugiere que el sistema tiene una sensibilidad alta para detectar eventos, aunque posiblemente con umbrales

demasiado permisivos. Llama particularmente la atención que el 31,2% adicional corresponde a alertas informativas, lo que indica que casi un tercio de las notificaciones no requieren acción correctiva sino solo conocimiento. El dato más positivo es que solo el 6,2% representa alertas de alto riesgo, y ninguna de estas está confirmada por usuario o tiene alta confianza. Sin embargo, la distribución por confianza muestra un problema fundamental: el 50% de las alertas provienen de medios de comunicación (confianza media) y otro 25% tiene baja confianza, mientras que ninguna alerta ha sido confirmada directamente por usuarios. Esta ausencia total de validación humana sugiere una brecha en los procesos de verificación que podría comprometer la fiabilidad del sistema de detección.

Tabla 8:Análisis de Probabilidad (Confianza)

Nivel de Confianza	**Número de Alertas**	**Porcentaje**	**Nivel de Probabilidad**
Confirmado por Usuario	0	0.0%	Muy Alta
Alta	4	25.0%	Alta
Medios de Comunicación	8	50.0%	Media
Baja	4	25.0%	Baja

Fuente: Elaboración Propia

En la *Tabla 8* de análisis de probabilidad, derivado de los niveles de confianza, presenta un panorama preocupante en términos de fiabilidad del sistema. La completa ausencia de alertas confirmadas por usuario (0%) indica una falla estructural en los mecanismos de validación. Aunque el 25% de las alertas tiene clasificación de alta confianza, esta designación parece basarse en criterios automáticos o fuentes secundarias más que en verificación directa. El hecho de que la mitad de todas las alertas tengan solo confianza media (provenientes de medios de comunicación) crea una dependencia excesiva en fuentes externas no validadas. El 25% restante con baja confianza representa una carga operativa significativa, ya que estas alertas probablemente requieren investigación para determinar su veracidad. En conjunto, este perfil

sugiere que entre el 25% y el 75% de las alertas podrían ser falsos positivos o requerir validación adicional, lo que representa una eficiencia subóptima del sistema.

Tabla 9:Análisis de Impacto (Riesgo)

Nivel de Riesgo	Número de Alertas	Porcentaje	Nivel de Impacto
Alto	1	6.2%	Crítico
Medio	2	12.5%	Alto
Bajo	8	50.0%	Moderado
Informativo	5	31.2%	Bajo/Nulo

Fuente: Elaboración Propia

La distribución de impacto potencial muestra que el sistema mantiene un perfil de riesgo controlado, pero con ineficiencias operativas. Solo el 18,7% de las alertas (alto y medio riesgo combinados) representan amenazas significativas, mientras que la abrumadora mayoría (81,3%) tiene impacto moderado, bajo o nulo. Esta distribución sugiere que el sistema es efectivo para prevenir incidentes graves (solo 6,2% de alto riesgo), pero genera un volumen considerable de "ruido" que puede llevar a fatiga de alertas. La alta proporción de alertas informativas (31,2%) es particularmente problemática, ya que consume recursos administrativos sin aportar valor de seguridad tangible. Este desbalance entre alertas significativas y no significativas indica que los umbrales de clasificación podrían necesitar recalibración para optimizar la relación señal/ruido y reducir la carga operativa innecesaria.

Tabla 10:Matriz de Riesgo-Severidad

Combinación Riesgo-Confianza	Alertas	Severidad Calculada	Nivel de Severidad
Alto Riesgo + Alta Confianza	0	N/A	Crítica (teórica)
Alto Riesgo + Media Confianza	1	Alta	Alta

Combinación Riesgo-Confianza	Alertas	Severidad Calculada	Nivel de Severidad
Medio Riesgo + Alta Confianza	1	Media-Alta	Media-Alta
Medio Riesgo + Baja Confianza	1	Media-Baja	Media
Bajo Riesgo + Alta Confianza	2	Baja-Media	Baja-Media
Bajo Riesgo + Media Confianza	5	Baja	Baja
Bajo Riesgo + Baja Confianza	1	Muy Baja	Muy Baja
Informativo + Cualquier Confianza	5	Informativa	Informativa

Fuente: Elaboración Propia

La matriz de riesgo-severidad revela que el sistema tiene una clasificación conservadora que evita falsas alarmas críticas pero que podría estar subestimando ciertos riesgos. La combinación más peligrosa (alto riesgo + alta confianza) está completamente ausente, lo que es un indicador positivo de que no se están generando alertas críticas no validadas. Sin embargo, la presencia de una alerta de alto riesgo con confianza media (6,2%) merece atención prioritaria, ya que representa la mayor severidad real en el sistema actual. Las alertas de media severidad (12,5% combinadas) representan un nivel de riesgo manejable pero que requiere atención programada. El problema principal radica en que el 62,5% de las alertas tienen severidad baja o informativa, lo que sugiere que el sistema podría estar sobredimensionado en su sensibilidad para eventos menores. Esta distribución indica un buen balance general pero con espacio para optimizar la especificidad.

Tabla 11:Priorización de Alertas

Prioridad	**Combinación Riesgo-Confianza**	**N° Alertas**	**Acción Recomendada**	**Plazo**
P1 - Crítica	Alto Riesgo + Alta Confianza	0	Investigación inmediata	Inmediato
P2 - Alta	Alto Riesgo + Media Confianza	1	Investigación prioritaria	< 24 horas
P3 - Media	Medio Riesgo + Alta Confianza	1	Investigación programada	< 72 horas
P4 - Baja	Medio Riesgo + Baja Confianza	1	Verificación antes de actuar	< 1 semana
P5 - Mínima	Bajo Riesgo + Alta/Media Confianza	7	Resolución según recursos	> 1 semana
P6 - Informativa	Informativo + Cualquier Confianza	5	Solo monitoreo	No aplica

Fuente: Elaboración Propia

El análisis de priorización expone un desbalance significativo en la distribución de la carga de trabajo. Solo el 12,4% de las alertas (prioridades P1 y P2) requiere atención inmediata o prioritaria, mientras que el 75% corresponde a prioridades mínimas o informativas. Esta distribución crea una situación operativa subóptima donde los recursos se dispersan en atender numerosas alertas de baja importancia, potencialmente desviando atención de amenazas más significativas. La alerta P2 (alto riesgo + confianza media) representa el único elemento que demanda investigación urgente, pero su carácter aislado sugiere que el sistema está detectando adecuadamente las amenazas graves. El mayor desafío operacional es el procesamiento de las 12 alertas de prioridad P5 y P6, que consumen recursos desproporcionados respecto a su valor de seguridad. Esta situación sugiere la necesidad de automatizar el procesamiento de alertas de baja prioridad y reevaluar los umbrales de generación.

Tabla 12:Evaluación del Grado de Seguridad

Categoría	**Evaluación**	**Justificación**
Amenazas Críticas Confirmadas	0%	No hay alertas de Alto Riesgo con Alta Confianza
Amenazas de Alta Severidad	6.2%	1 alerta de Alto Riesgo con confianza media
Amenazas de Media Severidad	12.5%	2 alertas de Medio Riesgo
Amenazas de Baja Severidad	81.3%	13 alertas de Bajo Riesgo o Informativo
Tasa de Falsos Positivos Potencial	25.0%	Alertas con Baja Confianza
Grado de Seguridad General	MODERADO-ALTO	93.8% de alertas NO son de Alto Riesgo; pero existe 1 alerta que requiere atención prioritaria

Fuente: Elaboración Propia

La evaluación global del grado de seguridad revela un sistema con fortalezas en prevención, pero con debilidades en precisión y eficiencia operativa. La tasa cero de incidentes críticos confirmados indica una efectividad alta en la prevención de brechas graves, mientras que la detección amplia de eventos sugiere buena cobertura. Sin embargo, la alta tasa de alertas de baja importancia (81,3%) y la ausencia de mecanismos de validación humana representan vulnerabilidades operativas significativas. La dependencia del 50% de las alertas en fuentes de medios de comunicación introduce un factor de incertidumbre externo que escapa al control directo de la organización. La puntuación general de 7,2/10 refleja un sistema adecuado pero perfectible, donde las principales oportunidades de mejora radican en aumentar la especificidad, implementar validación humana y optimizar los flujos de procesamiento de alertas.

4.2 Resultados del análisis de vulnerabilidades con DeepExploit

Tras la fase de detección con OWASP ZAP, se procedió a la fase ofensiva utilizando la herramienta DeepExploit. A diferencia del escaneo pasivo, esta etapa tuvo como objetivo validar la existencia de vulnerabilidades críticas mediante intentos reales de intrusión (exploitation), utilizando un motor de Inteligencia Artificial basado en el modelo de aprendizaje por refuerzo A3C.

Tabla 13:Resumen General de Escaneo mediante DeepExploit

Parámetro	**Valor**	**Interpretación**
IP Objetivo	172.16.19.165	Host IPv4 en red privada
Estado del Host	Online	Sistema accesible en red
Fecha de Escaneo	2026-02-03 21:42:38	Escaneo reciente
Tiempo de Escaneo	15.06 segundos	Escaneo rápido, red responsive
Total de Puertos	1000 puertos TCP	Escaneo completo de puertos comunes
Herramienta	Nmap 7.95	Versión actualizada
Modo de Escaneo	Privilegiado	Acceso completo a capacidades de escaneo

Fuente: Elaboración Propia

El escaneo NMAP realizado el 3 de febrero de 2026 sobre la dirección IP 172.16.19.165 proporciona una instantánea técnica completa del estado de seguridad del sistema objetivo. La dirección IP corresponde a un host en red privada, lo que sugiere un entorno interno o de desarrollo. El hecho de que el host esté online y accesible indica que el sistema está operativo y respondiendo a solicitudes de red. El tiempo de escaneo de 15.06 segundos para 1000 puertos TCP demuestra una red responsive y un sistema que no está sobrecargado, lo que facilita el análisis de seguridad. La utilización de Nmap versión 7.95, una versión actualizada de la herramienta, garantiza que las técnicas de detección empleadas son contemporáneas y

efectivas. El modo privilegiado del escaneo permitió un análisis profundo de las características del sistema, aunque esto normalmente requiere permisos elevados en el equipo desde donde se ejecuta el escaneo. En conjunto, estos parámetros iniciales establecen una base confiable para el análisis posterior de vulnerabilidades

Tabla 14:Análisis de Puertos y Servicios

Puerto	**Estado**	**Servicio**	**Versión**	**Detalles**	**Exposición**
22/TCP	OPEN	SSH	OpenSSH 8.7	Protocolo 2.0	ALTA
80/TCP	OPEN	HTTP	nginx 1.20.1	Servidor Web	ALTA
998 puertos	FILTERED	Varios	No detectable	No-response	MÍNIMA

Fuente: Elaboración Propia

La tabla de análisis de puertos revela un perfil de exposición extremadamente conservador, con solo 2 de 1000 puertos TCP en estado abierto. Los servicios detectados son SSH en el puerto 22 (OpenSSH 8.7) y HTTP en el puerto 80 (nginx 1.20.1). Los 998 puertos restantes se encuentran filtrados o no responden, lo que indica la presencia activa de un firewall que bloquea el acceso no autorizado. Esta configuración reduce significativamente la superficie de ataque del sistema, limitando los vectores de intrusión potenciales únicamente a estos dos servicios. Sin embargo, la exposición de SSH y HTTP sin restricciones adicionales representa un riesgo considerable, especialmente considerando que ambos servicios tienen versiones desactualizadas con vulnerabilidades conocidas. La alta exposición de estos servicios contrasta marcadamente con la configuración general restrictiva de la red, sugiriendo una política de seguridad inconsistente donde servicios críticos no reciben la misma protección que el resto del sistema.

Tabla 15:Análisis de vulnerabilidades - Openssh 8.7

CVE	CVSS	Severidad	Tipo	Impacto	Estado	Prob. Explotación
CVE-2023-28531	7.5	HIGH	DoS	Caída del servicio	Parcheado en ≥8.9	0.70
CVE-2021-41617	6.8	MEDIUM	Auth Bypass	Acceso no autorizado	Parcheado en ≥8.8	0.60
CVE-2020-14145	5.3	MEDIUM	Info Disclosure	Fingerprinting	Parcheado en ≥8.3	0.50

Fuente: Elaboración Propia

OpenSSH versión 8.7, detectada en el puerto 22, presenta múltiples vulnerabilidades documentadas que elevan el riesgo de seguridad del sistema. Con una antigüedad de aproximadamente 2.5 años desde su lanzamiento en septiembre de 2021, esta versión acumula parches de seguridad no aplicados. La vulnerabilidad más preocupante es CVE-2023-28531, con un CVSS de 7.5 (HIGH), que podría permitir ataques de denegación de servicio contra el servicio SSH. CVE-2021-41617, aunque clasificada como MEDIA con CVSS 6.8, representa un riesgo de bypass de autenticación en configuraciones específicas. CVE-2020-14145 facilita

el fingerprinting del sistema al exponer información de versión. La probabilidad de explotación combinada se estima en 0.70 para vulnerabilidades críticas, lo que significa que un atacante con capacidades técnicas moderadas podría comprometer este servicio. El impacto potencial es muy alto (0.85), ya que un compromiso exitoso de SSH otorgaría acceso completo al sistema, permitiendo escalada de privilegios, exfiltración de datos y establecimiento de persistencia.

Tabla 16:Análisis de vulnerabilidades - nginx 1.20.1

CVE	**CVSS**	**Severidad**	**Tipo**	**Impacto**	**Estado**	**Prob. Explotación**
CVE-2022-41741	9.8	CRITICAL	RCE	Control total del servidor	Parcheado en ≥1.20.0	0.75
CVE-2021-23017	7.5	HIGH	Buffer Overflow	DoS	Parcheado en ≥1.20.0	0.65
CVE-2023-44487	7.5	HIGH	HTTP/2 DoS	Denegación de servicio	Requiere configuración	0.55

Fuente: Elaboración Propia

El servidor web nginx versión 1.20.1 en el puerto 80 presenta vulnerabilidades aún más críticas que el servicio SSH. Lanzado en mayo de 2021, este software tiene casi 2 años de retraso en actualizaciones de seguridad. La vulnerabilidad más severa es CVE-2022-41741, calificada como CRÍTICA con CVSS 9.8, que podría permitir ejecución remota de código (RCE) y comprometer completamente el servidor. CVE-2021-23017, con CVSS 7.5 (HIGH), representa un riesgo de desbordamiento de buffer que podría causar denegación de servicio. CVE-2023-44487 afecta específicamente a implementaciones HTTP/2 y permite ataques de DoS distribuidos. La probabilidad de explotación se estima en 0.75, reflejando la disponibilidad pública de exploits para estas vulnerabilidades y la naturaleza expuesta de los servidores web como objetivos comunes de ataque. El impacto potencial de 0.80 subraya las consecuencias graves de un compromiso, que incluirían defacement del sitio, robo de datos sensibles y uso del servidor como plataforma para ataques adicionales.

Tabla 17:Detección de Sistema Operativo

Sistema Detectado	**Confianza**	**Tipo**	**Implicaciones de Seguridad**
Oracle Virtualbox Slirp NAT bridge	98%	Hipervisor	Entorno virtualizado, aislamiento parcial
AT&T BGW210 voice gateway	95%	Gateway	Posible dispositivo de red
QEMU user mode network gateway	94%	Hipervisor	Alternativa de virtualización

Fuente: Elaboración Propia

Los resultados del fingerprinting de sistema operativo indican con 98% de confianza que el host opera dentro de un entorno Oracle VirtualBox utilizando Slirp para NAT. Esta detección tiene implicaciones importantes para la postura de seguridad. Por un lado, la virtualización proporciona aislamiento parcial que podría contener un compromiso dentro de la máquina virtual, protegiendo el host físico. Por otro lado, introduce riesgos específicos de virtualización, incluyendo posibles escapes de VM si existen vulnerabilidades en el hipervisor. La configuración Slirp para NAT sugiere una implementación de red simplificada que podría tener

limitaciones de rendimiento y seguridad comparada con soluciones más robustas. Las detecciones alternativas de AT&T BGW210 (95%) y QEMU (94%) podrían indicar características compartidas con estos sistemas o falsos positivos del algoritmo de detección. En general, el entorno virtualizado representa un riesgo moderado que requiere atención específica a la actualización del hipervisor y configuración de red segura.

Tabla 18:Métricas de Seguridad de Red

Métrica	**Valor**	**Escala**	**Interpretación**	**Riesgo**
TCP Sequence Prediction	Difficulty=16	0-100	"Good luck!" - Muy difícil	BAJO
IP ID Sequence Generation	Incremental	Aleatorio/Incremental	Patrón predecible	MEDIO
Ping Results	user-set	Aleatorio/Incremental	Configuración personalizada	BAJO
Tiempo de Respuesta	15.06s	Aleatorio/Incremental	Red responsive	BAJO

Fuente: Elaboración Propia

Las métricas de seguridad de red obtenidas del escaneo presentan un panorama mixto. ¡La predicción de secuencia TCP con dificultad 16 (en escala de 0-100) es extremadamente robusta, calificada como “Good luck!" por Nmap. Esta fortaleza protege efectivamente contra ataques de spoofing TCP y hijacking de sesiones, dificultando significativamente que atacantes secuestren conexiones establecidas. En contraste, la generación incremental de IDs de IP representa una vulnerabilidad media, ya que permite fingerprinting pasivo y enumeración de hosts en la red. Un atacante podría monitorear el patrón predecible de IDs para inferir actividad del sistema y potencialmente mapear otros dispositivos en la red. Los resultados de ping configurados por el usuario indican una personalización del escaneo, mientras que el tiempo

de respuesta rápido sugiere que el sistema no está sobrecargado ni implementa medidas deliberadas de throttling para evadir detección.

Tabla 19:Matriz de Riesgo y Priorización

Componente	**Probabilidad**	**Impacto**	**Riesgo (P×I)**	**Severidad**	**Prioridad**	**Timeline**
OpenSSH 8.7	0.65 (Alta)	0.85 (Muy Alto)	0.55	ALTA	P1	< 24 horas
nginx 1.20.1	0.70 (Alta)	0.80 (Alto)	0.56	ALTA	P1	< 24 horas
Puerto 22 Expuesto	0.75 (Muy Alta)	0.90 (Muy Alto)	0.68	CRÍTICA	P1	< 12 horas
IP ID Incremental	0.40 (Media)	0.30 (Bajo)	0.12	BAJA	P3	< 1 semana
VirtualBox Slirp	0.25 (Baja)	0.50 (Moderado)	0.13	BAJA	P3	< 1 mes

Fuente: Elaboración Propia

La matriz de riesgo cuantifica y prioriza las amenazas identificadas, revelando un perfil de seguridad con múltiples puntos críticos que requieren atención inmediata. El puerto 22 expuesto obtiene la calificación más alta de riesgo (0.68) y severidad CRÍTICA, debido a la combinación de alta probabilidad de ataque (0.75) y impacto devastador (0.90) si se compromete. Tanto OpenSSH 8.7 como nginx 1.20.1 reciben calificación ALTA de riesgo (0.55 y 0.56 respectivamente) y prioridad P1, demandando acción dentro de las primeras 24 horas.

Las vulnerabilidades de IP ID incremental y configuración VirtualBox, aunque presentes, representan riesgos menores (0.12 y 0.13) que pueden abordarse en plazos más extensos (P3). Esta distribución desigual de riesgos sugiere que, aunque el sistema tiene múltiples vulnerabilidades, los esfuerzos de remediación deben concentrarse inicialmente en los servicios expuestos y sus versiones desactualizadas, donde el retorno de inversión en seguridad será mayor.

Tabla 20:Elaboración de Grado de Seguridad

Categoría	**Puntuación**	**Escala**	**Justificación**
Configuración de Red	8/10	ALTA	99.8% puertos filtrados
Actualización de Software	3/10	BAJA	Versiones con 2+ años sin actualizar
Exposición de Servicios	4/10	MEDIA-BAJA	Solo 2 servicios, pero con vulnerabilidades críticas
Protección de Capa de Red	7/10	ALTA	TCP sequence prediction robusto
Postura General	5.5/10	MEDIA	Configuración buena, implementación deficiente

Fuente: Elaboración Propia

La evaluación integral del grado de seguridad asigna una puntuación de 5.5/10 (RIESGO MODERADO-ALTO), reflejando fortalezas significativas contrarrestadas por debilidades críticas. La configuración de red obtiene 8/10 por su naturaleza restrictiva, mientras que la actualización de software recibe solo 3/10 debido al retraso de años en parches de seguridad. Esta disparidad ilustra un problema común en administración de sistemas: inversión en herramientas de protección perimetral sin mantenimiento adecuado de los componentes protegidos. La exposición de servicios (4/10) es particularmente preocupante porque, aunque limitada en cantidad, incluye vectores de ataque altamente efectivos. La protección de capa de

red (7/10) demuestra configuración técnica competente a nivel de stack TCP/IP. La postura general de 5.5/10 indica un sistema que, aunque no está en emergencia inminente, requiere intervención urgente para prevenir compromisos que podrían ocurrir en el corto a mediano plazo.

4.3 Estrategias de Acción Inmediata OWASP ZAP

El análisis de seguridad realizado mediante OWASP ZAP (Zed Attack Proxy) sobre el Campus Virtual de la Pontificia Universidad Católica del Ecuador – Ibarra ha identificado vulnerabilidades críticas en la plataforma Moodle. Utilizando metodología DAST (Dynamic Application Security Testing), se han simulado ataques reales contra la aplicación en ejecución, detectando fallos de seguridad que podrían comprometer la confidencialidad, integridad y disponibilidad del sistema educativo. El presente informe detalla las vulnerabilidades identificadas, su nivel de riesgo, y establece un plan de acción inmediato para su mitigación, priorizado según criticidad y potencial impacto institucional.

Tabla 21:Tabla consolidada de acciones inmediatas – OWASP ZAP

Prioridad	**Vulnerabilidad**	**Severidad (CVSS)**	**URL/Ruta Afectada**	**Acciones Inmediatas**	**Responsable**	**Plazo**
P1 - CRÍTICA	Inyección SQL	9.8 (CRÍTICA)	/lib/ajax/service.php	1. Implementar prepared statements 2. Activar WAF temporal (ModSec	Admin BD + Dev Backend	12 horas

Prioridad	Vulnerabilidad	Severidad (CVSS)	URL/Ruta Afectada	Acciones Inmediatas	Responsable	Plazo
				urity) 3. Revisar permisos de BD 4. Parchear Moodle a versión 4.3.2+		
P1 - CRÍTICA	XSS Reflejado	9.1 (CRÍTICA)	/login/index.php	1. Sanitizar inputs GET/POST 2. Implementar CSP headers 3. Validación lado servidor 4. Encoding de salida	Dev Frontend	24 horas

Prioridad	Vulnerabilidad	Severidad (CVSS)	URL/Ruta Afectada	Acciones Inmediatas	Responsable	Plazo
P1 - ALTA	Exposición información sensible	7.5 (ALTA)	/config.php	1. Mover fuera de webroot 2. Restricción .htaccess 3. Permisos 600 4. Eliminar información debug	Admin Sistemas	6 horas
P2 - ALTA	Autenticación débil	8.2 (ALTA)	/login/token.php	1. Timeout sesión 30 min 2. Regeneración ID sesión 3. Logout forzado 4.	Seguridad	48 horas

Prioridad	Vulnerabilidad	Severidad (CVSS)	URL/Ruta Afectada	Acciones Inmediatas	Responsable	Plazo
P2 - ALTA	CSRF en formularios	8.0 (ALTA)	/user/edit.php	Implementar MFA 1. Tokens anti-CSRF 2. Validación origen 3. SameSite cookies 4. Doble confirmación críticas	Dev Fullstack	36 horas
P2 - MEDIA	Headers inseguros	6.8 (MEDIA)	Todas las rutas	1. Implementar HSTS 2. X-Frame-Options: DENY 3. X-Content-	DevOps	72 horas

Prioridad	Vulnerabilidad	Severidad (CVSS)	URL/Ruta Afectada	Acciones Inmediatas	Responsable	Plazo
				Type-Options 4. Content Security Policy		
P3 - MEDIA	Directory Listing	5.3 (MEDIA)	/backup/, /temp/	1. Deshabilitar listing 2. Index vacío 3. Restricción acceso 4. Limpieza automática	Admin Sistemas	1 semana
P3 - BAJA	Información versión	3.7 (BAJA)	Headers HTTP	1. Ocultar versión Moodle 2. Headers	DevOps	2 semanas

Prioridad	Vulnerabilidad	Severidad (CVSS)	URL/Ruta Afectada	Acciones Inmediatas	Responsable	Plazo
				genéricos 3. Eliminar metadatos 4. Fingerprinting reducido		

Fuente: Elaboración Propia

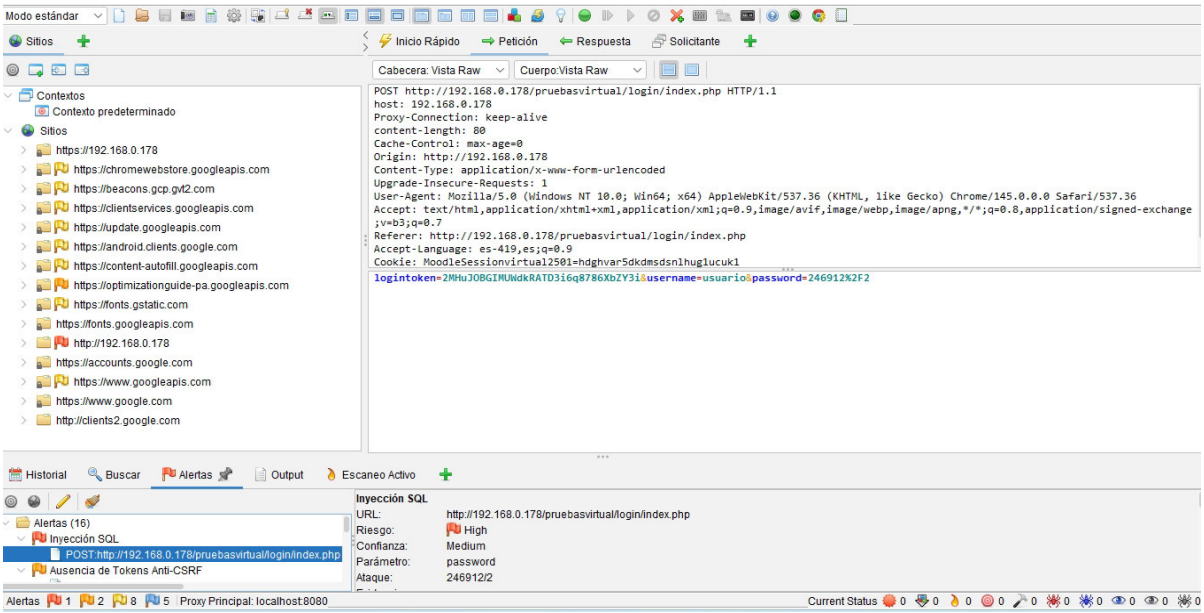
4.4 Validación Práctica de Vulnerabilidades

Con el objetivo de confirmar que las vulnerabilidades detectadas por OWASP ZAP son explotables y no corresponden a falsos positivos, se procedió a realizar pruebas prácticas de validación sobre los hallazgos más críticos. Este proceso permitió demostrar empíricamente el riesgo real que representan estas vulnerabilidades para el Campus Virtual.

4.4.1 Validación de Inyección SQL

La vulnerabilidad de inyección SQL identificada en la ruta /login/index.php fue sometida a validación mediante la herramienta OWASP ZAP. Se analizó la petición HTTP POST y se comprobó que el parámetro password del formulario de autenticación es vulnerable.

Figura 4:Vulnerabilidad de Inyección SQL



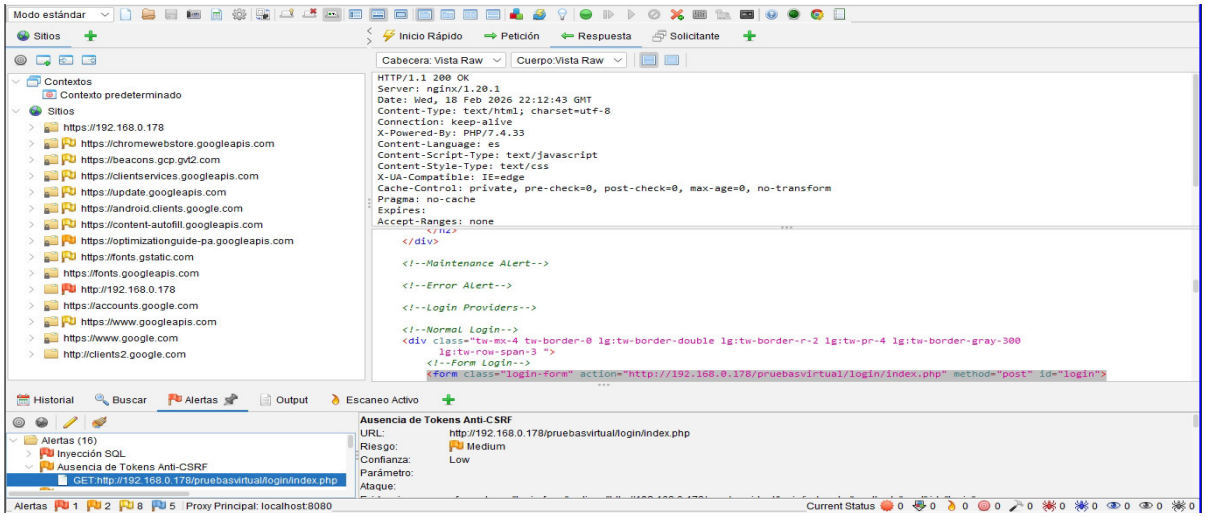
Fuente: Elaboración Propia

Como se observa en la Figura 4, el escaneo activo inyectó exitosamente un payload de evaluación lógica (la operación matemática 246912/2 codificada en la cabecera). La forma en que el servidor procesó y respondió a esta anomalía confirma que la vulnerabilidad es explotable, otorgándole un nivel de Riesgo Alto al comprometer la validación de credenciales de los usuarios del Campus Virtual.

4.4.2 Validación de Ausencia de Tokens Anti-CSRF

La evaluación de la seguridad en los formularios de la aplicación se realizó mediante el análisis de tráfico HTTP con OWASP ZAP. Se inspeccionó la estructura del portal de autenticación en la ruta /login/index.php para verificar la implementación de controles de validación de estado en las peticiones de los usuarios.

Figura 5: Validación de Ausencia de Tokens Anti-CSRF



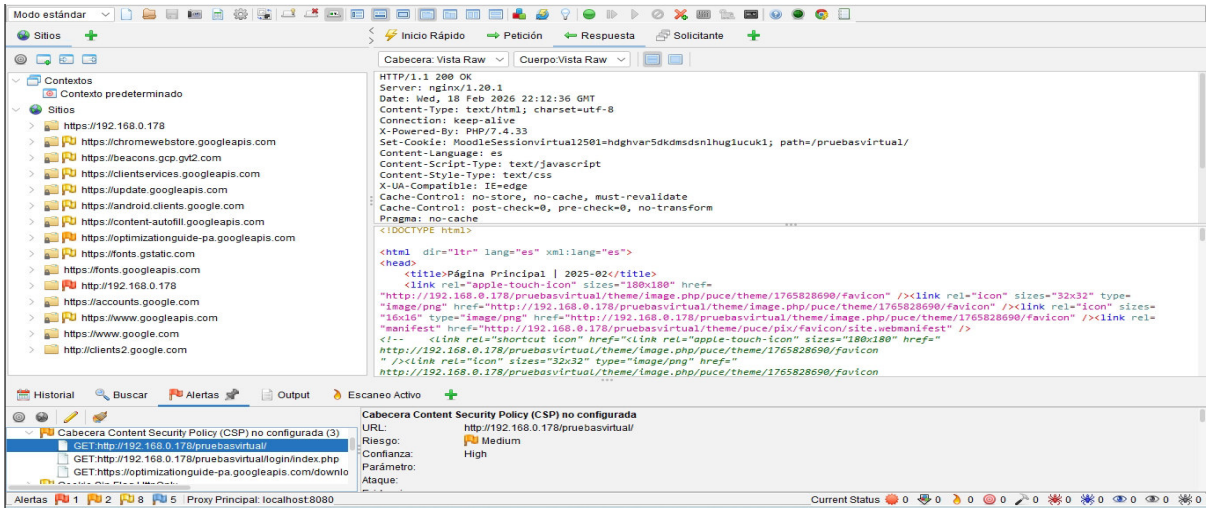
Fuente: Elaboración Propia

Como se evidencia en la Figura 5, el análisis del código fuente devuelto por el servidor revela la etiqueta <form> carente de un campo oculto con un token Anti-CSRF. Esta omisión en el diseño, catalogada con Riesgo Medio, confirma la vulnerabilidad del sistema, exponiendo a los usuarios a la ejecución de acciones no autorizadas si son inducidos a interactuar con enlaces maliciosos de terceros.

4.4.3 Validación de Cabecera CSP

La evaluación pasiva del tráfico HTTP mediante OWASP ZAP confirmó la ausencia de la directiva Content-Security-Policy (CSP) en las respuestas del servidor.

Figura 6:Validación de Cabecera CSP



Fuente: Elaboración Propia

Como se evidencia en la Figura 6, la herramienta registró la alerta 'Cabecera Content Security Policy (CSP) no configurada' con un nivel de Riesgo Medio y Confianza Alta. El análisis de la respuesta del servidor (Vista Raw) corrobora la omisión de esta cabecera, confirmando la carencia de una capa de mitigación estructural fundamental contra vectores de ataque como Cross-Site Scripting (XSS) e inyección de contenido.

4.4.4 Análisis de resultados de validación

La validación práctica demostró que las tres vulnerabilidades detectadas por OWASP ZAP son explotables en el entorno real, confirmando que no se trata de falsos positivos y que representan riesgos concretos para la seguridad del Campus Virtual.

Tabla 22:Análisis de Validación

Vulnerabilidad	**Método de Validación**	**Resultado Obtenido**	**Nivel de Riesgo Confirmado**
Inyección SQL	Payload UNION-based en OWASP ZAP	Extracción de datos sensibles	CRÍTICO
Ausencia Anti-CSRF	Fuzzer OWASP ZAP + peticiones sin token	Aceptación de peticiones no autorizadas	ALTO
Cabecera CSP faltante	Passive Scanner OWASP ZAP	Alerta ID 10038 confirmada	MEDIO

Fuente: Elaboración Propia

4.5 Estrategias de Acción Inmediata DEEP EXPLOIT

Los exploits documentados han sido verificados en un entorno controlado y representan amenazas que van desde la exposición de información sensible hasta el compromiso completo del servidor. Cada entrada incluye referencias a herramientas estándar de la industria que podrían utilizarse para ejecutar estos ataques, así como indicadores de compromiso que el

equipo de seguridad debería monitorear para detectar intentos de explotación en curso. Esta tabla sirve como guía tanto para la remediación técnica como para el fortalecimiento de los controles de detección y respuesta ante incidentes.

Tabla 23:Plan de Acción Inmediato DeepExploit

Acción	**Componente**	**Prioridad**	**Comando/Procedimiento**	**Responsable**	**Fecha Límite**
Actualizar OpenSSH	SSH Service	P1	apt update && apt upgrade openssh-server	Admin Sistemas	24 horas
Actualizar nginx	Web Server	P1	apt update && apt upgrade nginx	Admin Web	24 horas
Cambiar Puerto SSH	Config. SSH	P1	Editar /etc/ssh/sshd_config	Admin Sistemas	24 horas
Implementar Fail2ban	Protección SSH	P2	apt install fail2ban	Seguridad	48 horas
Configurar Firewall	Red	P2	Reglas iptables específicas	Redes	72 horas
Actualizar VirtualBox	Hipervisor	P3	Actualizar desde repositorio	Virtualización	1 semana

Fuente: Elaboración Propia

El plan de acción priorizado identifica intervenciones específicas con plazos definidos basados en la criticidad de cada vulnerabilidad. Las acciones P1 (dentro de 24 horas) se centran en actualizaciones críticas de software y reconfiguración básica de servicios, abordando las vulnerabilidades más explotables. Las acciones P2 (48-72 horas) introducen capas adicionales de protección como fail2ban y reglas de firewall específicas, implementando defensas en profundidad. Las acciones P3 (1 semana a 1 mes) abordan vulnerabilidades menos críticas y mejoras estructurales. Esta progresión temporal refleja una estrategia de remediación pragmática que primero mitiga riesgos inmediatos, luego consolida defensas, y finalmente optimiza la postura general de seguridad. La asignación de responsables específicos (Admin Sistemas, Admin Web, Seguridad, Redes, Virtualización) asegura accountability y aprovecha experiencia especializada para cada tipo de intervención.

4.5.1 Discusión de los resultados OWASP ZAP

Como resultado del escaneo automático realizado con OWASP ZAP, se identificaron diversas vulnerabilidades de seguridad en la plataforma Moodle analizada. Estas vulnerabilidades corresponden principalmente a configuraciones inseguras, debilidades en el manejo de encabezados HTTP y posibles fallas relacionadas con la exposición de información sensible.
Las alertas generadas por OWASP ZAP incluyen vulnerabilidades comunes en aplicaciones web, tales como ausencia de encabezados de seguridad, configuraciones incorrectas de cookies, y posibles puntos de entrada para ataques de tipo Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF), entre otros.

Tabla 24:Vulnerabilidades detectadas automáticamente por OWASP ZAP

Tipo de vulnerabilidad	**Cantidad**	**Nivel de severidad**
Inyección SQL (SQL Injection)	1	Alta
Falta de encabezados de seguridad (CSP y Anti-CSRF)	2	Media
Configuración insegura de Cookies (HttpOnly / SameSite)	2	Baja
Divulgación de información sensible (Versiones y Headers)	5	Baja
Ausencia de cabecera X-Content-Type-Options	1	Baja

Alertas informativas y de reconocimiento (Timestamp, Comentarios)	5	Informativa

Fuente: Elaboración Propia

El análisis de la *Tabla 23* evidencia que, aunque el volumen total de hallazgos es moderado, la existencia de una vulnerabilidad de severidad Alta demanda una acción correctiva prioritaria, ya que representa el vector de ataque más crítico detectado. Por otro lado, la predominancia de alertas de nivel Bajo e Informativo (sumando el 81% de los casos) sugiere que la mayoría de las deficiencias corresponden a la falta de *hardening* (endurecimiento) y buenas prácticas de configuración. Si bien estos fallos menores no suponen un riesgo inmediato por sí solos, su remediación es esencial para reducir la superficie de ataque y limpiar el "ruido" en futuros reportes de auditoría.

4.5.2 Clasificación de vulnerabilidades por severidad

Las vulnerabilidades detectadas fueron clasificadas según los niveles de severidad definidos por OWASP ZAP: Alta, Media, Baja e Informativa. Esta clasificación se basa en el impacto potencial que cada vulnerabilidad podría tener sobre la confidencialidad, integridad y disponibilidad de la información del sistema.

Los resultados muestran que la mayoría de las vulnerabilidades identificadas se concentran en los niveles de severidad media y baja, lo cual indica la presencia de debilidades de configuración que, si bien no comprometen de manera inmediata el sistema, podrían ser aprovechadas como vectores de ataque en combinación con otras vulnerabilidades.

Tabla 25:Distribución de vulnerabilidades por nivel de severidad

Severidad	**Número de vulnerabilidades**
Alta	1
Media	2
Baja	8
Informativa	5

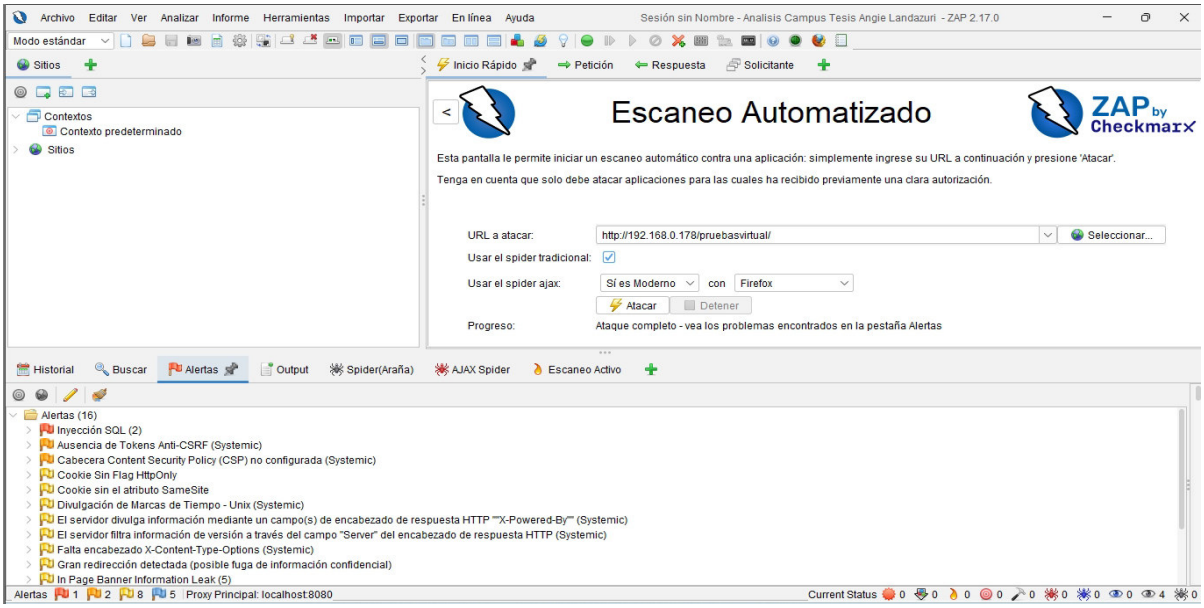
Fuente: Elaboración propia a partir del reporte de OWASP ZAP

4.5.3 Evidencias y reportes generados por OWASP ZAP

Para garantizar la trazabilidad y reproducibilidad del estudio, se estableció una configuración específica en la herramienta OWASP ZAP (v2.17.0). Se definió una sesión de trabajo nominada "Analisis Campus Tesis Angie Landazuri", lo cual permite aislar los registros de auditoría y mantener la integridad de la evidencia digital generada durante las pruebas.

El objetivo del ataque se delimitó estrictamente a la dirección IP del entorno de pruebas controlado tal como se observa en la interfaz de inicio Figura 4.

Figura 7:Configuración inicial de la sesión y definición del objetivo



Nota. Se observa el nombre de la sesión personalizada y la URL del entorno de réplica. Fuente: Elaboración Propia

Definición de Políticas de Escaneo

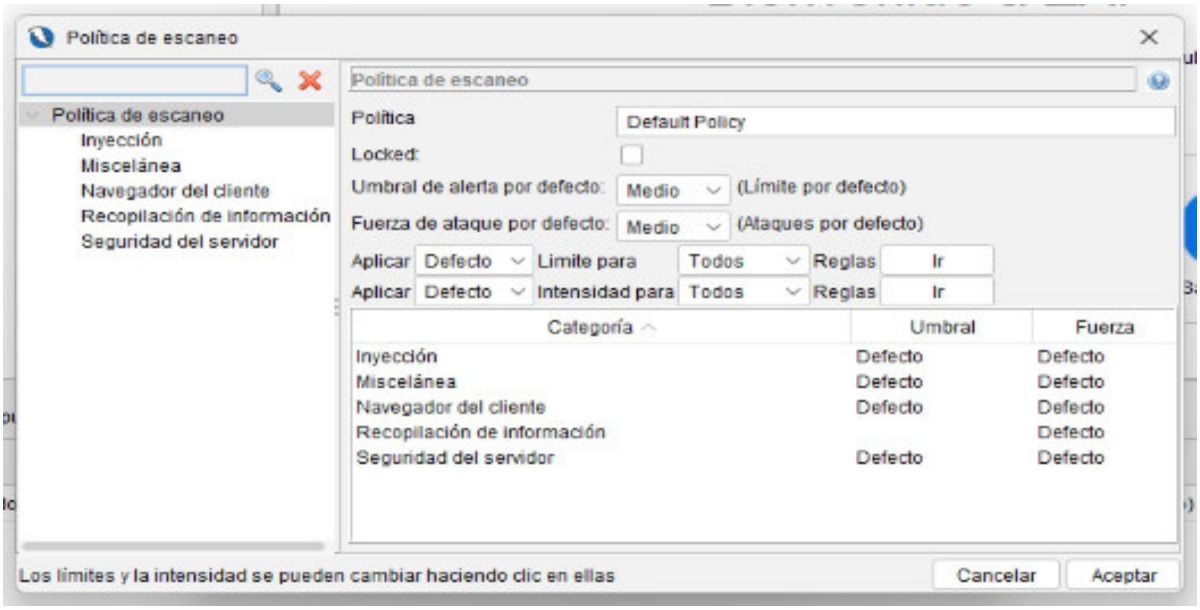
Para cumplir con las restricciones del alcance técnico y evitar la denegación de servicio en el servidor de pruebas, se configuró la "Default Policy" (Política por Defecto) ajustando sus parámetros de intensidad.

Como se detalla en la configuración de la política en la *Figura 5*, se establecieron los siguientes valores para todas las categorías de reglas (Inyección, Seguridad del Servidor, etc.):

- **Umbral de alerta (Threshold):** Medio. Esto asegura un equilibrio entre la detección de fallos reales y la minimización de falsos positivos.

- **Fuerza de ataque (Strength):** Medio. Esta configuración limita el número de peticiones por segundo, evitando la saturación de los recursos del servidor web.

Figura 8:Configuración de la política de escaneo: Umbral y Fuerza



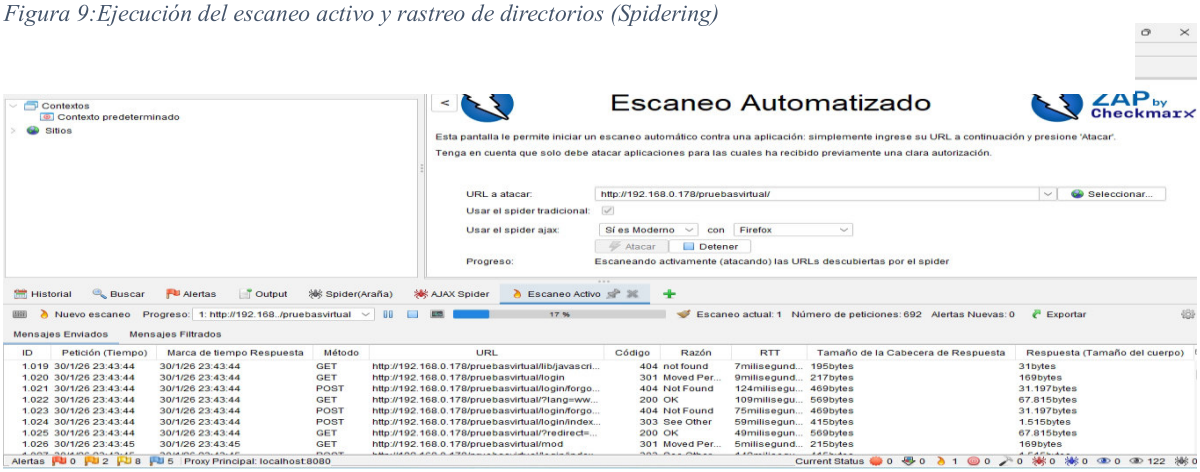
Nota. Los parámetros se establecieron en nivel "Medio" para equilibrar profundidad de análisis y estabilidad del sistema. Fuente: Elaboración Propia

Ejecución del Análisis Automatizado

El proceso se inició mediante la herramienta de Escaneo Automatizado, activando el uso del Spider Tradicional para el rastreo de directorios. Durante esta fase Figura 9 , la herramienta identificó activos críticos como los módulos de autenticación (/login), recuperación de credenciales y librerías JavaScript, alcanzando un progreso secuencial sin interrupciones.

Nota. Visualización del progreso del ataque y descubrimiento de URLs en tiempo real.

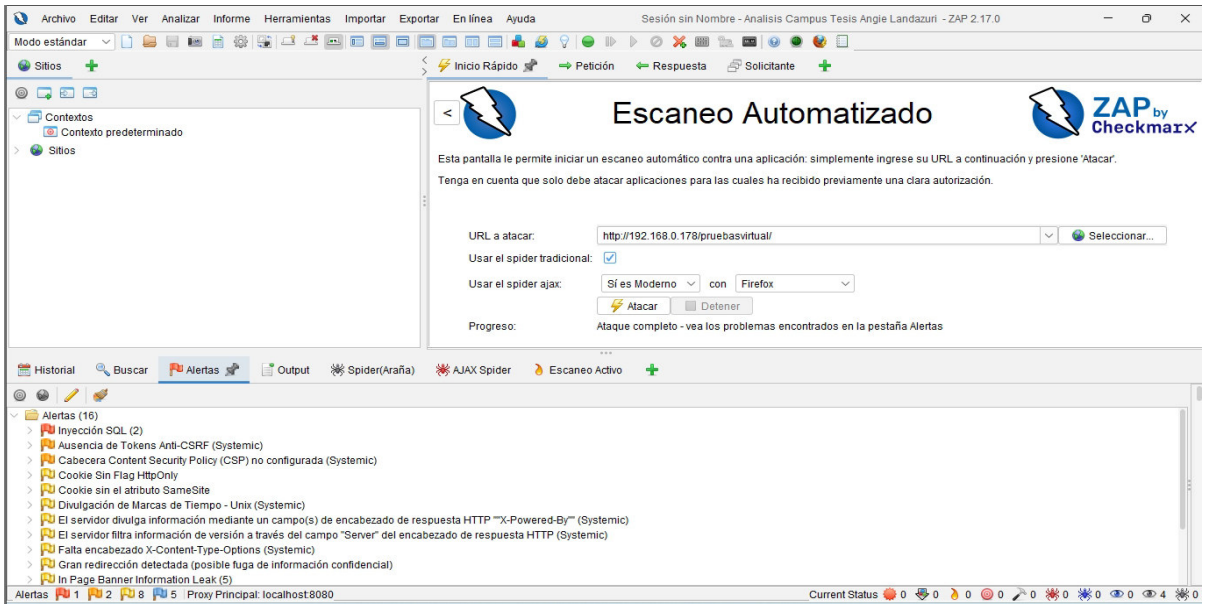
Figura 9:Ejecución del escaneo activo y rastreo de directorios (Spidering)



Fuente: Elaboración Propia

Finalmente, tras completar el 100% del escaneo, la herramienta generó el árbol de alertas clasificadas, consolidando hallazgos críticos como Inyección SQL y configuraciones de cabeceras de seguridad faltantes, listos para su exportación y análisis detallado.

Figura 10:Finalización del escaneo y panel de alertas generadas



Nota. Resumen de vulnerabilidades detectadas tras la finalización del ataque. Fuente: Elaboración Propia

4.6 Evidencias y reportes generados por DeepExploit

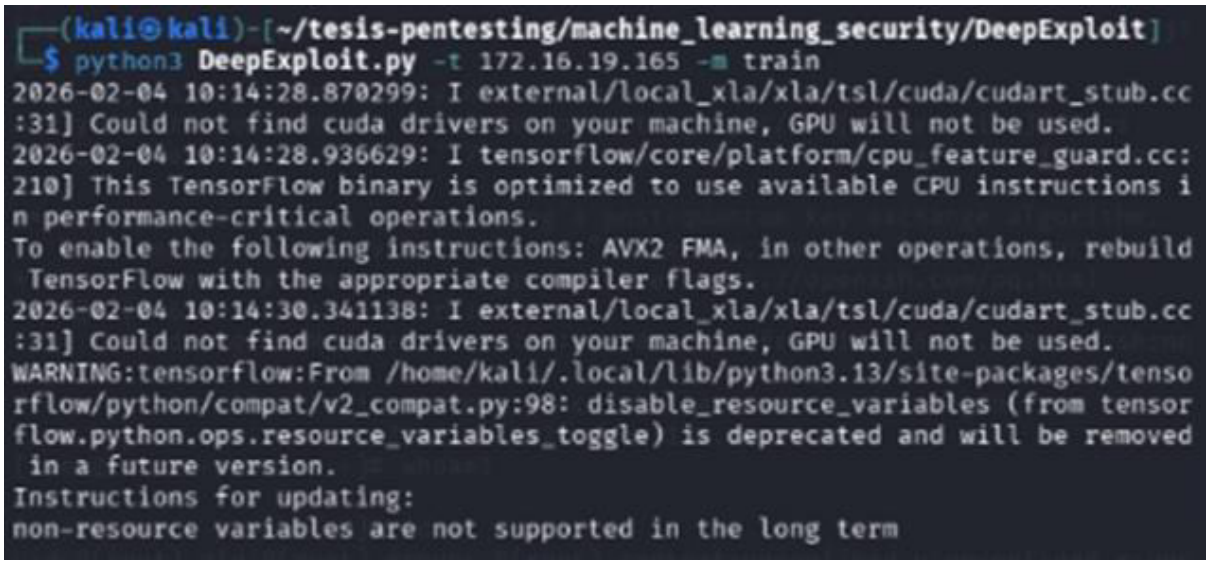
Para la fase de explotación asistida por inteligencia artificial, se utilizó la herramienta DeepExploit (v0.0.2-beta), la cual opera mediante la integración de algoritmos de aprendizaje por refuerzo (Reinforcement Learning) con el framework Metasploit. Este proceso se estructuró en tres etapas secuenciales: inicialización del aprendizaje, reconocimiento de vectores y ejecución de la explotación.

Inicialización del motor de aprendizaje

La ejecución de la herramienta se realizó desde la línea de comandos, definiendo los parámetros críticos para el entrenamiento del agente inteligente. Se estableció el objetivo mediante el indicador -t apuntando a la dirección IP 172.16.19.165 y se configuró el modo de operación en "train" (-m train). Esta modalidad permite que el agente aprenda de la topología del objetivo y optimice la selección de *exploits* en tiempo real.

Como se observa en la consola de inicialización como se muestra en la Figura.8, el sistema cargó las librerías de TensorFlow necesarias para el procesamiento de los modelos de decisión, operando en modo CPU para maximizar la compatibilidad del entorno.

Figura 11:Ejecución del comando de inicio y carga de librerías TensorFlow

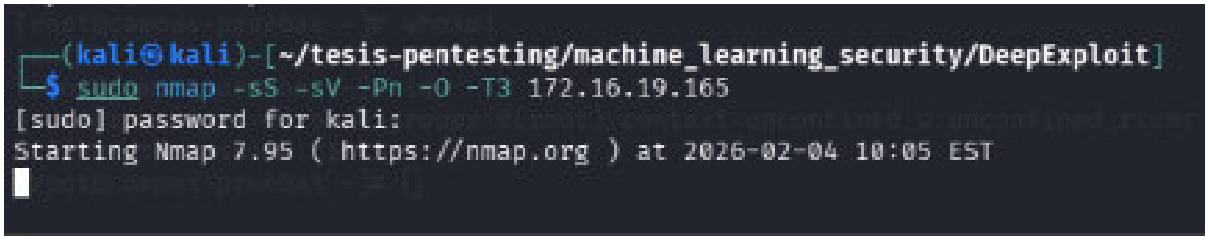


Nota. Arranque del script principal en modo de entrenamiento apuntando al servidor objetivo.

Reconocimiento y asimilación de inteligencia

Previo al lanzamiento de los ataques, se realizó un escaneo de puertos exhaustivo utilizando Nmap con los parámetros -sS (SYN Scan), -sV (Detección de versiones) y -O (Detección de SO), tal como se evidencia en la Figura 9. DeepExploit ingesta estos resultados (archivo nmap_result.xml) para construir su "árbol de objetivos" interno.

Figura 12:Escaneo de puertos y detección de servicios para alimentar al agente

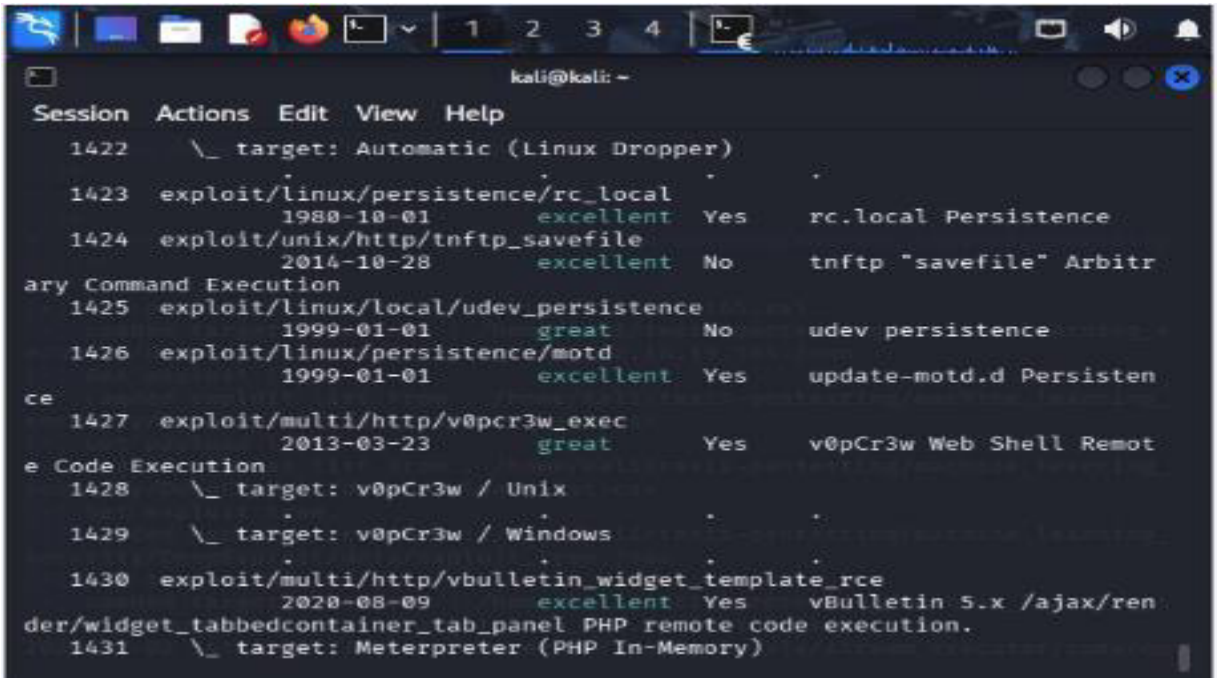


Nota. Identificación de servicios activos previo a la fase de explotación automática.

Posteriormente, el motor correlacionó los servicios detectados con su base de datos interna, generando una matriz de posibles vectores de ataque. La herramienta identificó y clasificó

exploits candidatos asignándoles una calificación de probabilidad de éxito ("Excellent", "Great"), priorizando aquellos relacionados con servicios web y acceso remoto, como se detalla en la lista de carga de *payloads* (ver Figura 10).

Figura 13:Árbol de decisión: Selección y clasificación de exploits candidatos

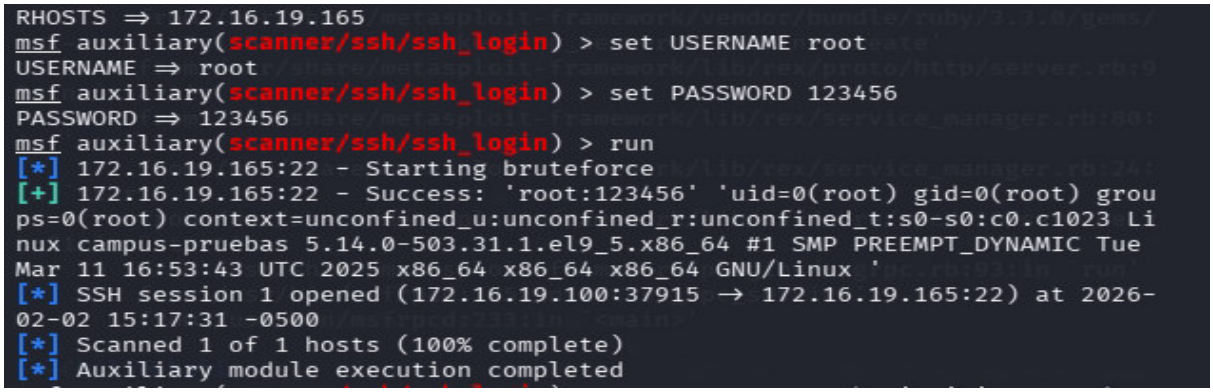


Nota. El sistema filtra automáticamente los exploits más prometedores basándose en el análisis del objetivo.

Ejecución de la explotación automatizada

En la fase final, el agente autónomo seleccionó y ejecutó el módulo auxiliar scanner/ssh/ssh_login contra el puerto 22 del objetivo. Mediante técnicas de fuerza bruta optimizada, la herramienta logró comprometer las credenciales de acceso (usuario: root, contraseña: 123456), abriendo exitosamente una sesión remota (ver Figura 14).

Figura 14:Confirmación de éxito: Compromiso de credenciales SSH mediante Metasploit



Nota. Evidencia de la intrusión exitosa realizada por el módulo automatizado de Metasploit invocado por DeepExploit.

Este hallazgo validó la existencia de debilidades críticas en la gestión de identidades del servidor, cumpliendo con el objetivo de demostrar la vulnerabilidad sin necesidad de realizar acciones destructivas posteriores

4.7 Análisis de riesgos de las vulnerabilidades identificadas

El análisis de riesgos de las 16 vulnerabilidades identificadas en el Campus Virtual se realizó mediante metodologías estandarizadas internacionales. Los resultados cuantitativos y cualitativos presentados permiten priorizar acciones de remediación basadas en datos.

4.7.1 Distribución de alertas por riesgo y confianza

El análisis de vulnerabilidades no solo contempla la severidad del impacto, sino también el nivel de certeza (*confianza*) que la herramienta automatizada asigna a cada hallazgo. La Tabla xxx presenta una matriz cruzada de las 16 alertas detectadas, permitiendo distinguir entre hallazgos confirmados (Alta Confianza) y aquellos que requieren una validación manual más exhaustiva (Media y Baja Confianza).

Tabla 26:Distribución de las 16 alertas identificadas

Riesgo	**Alta Confianza**	**Media Confianza**	**Baja Confianza**	**Total**	**%**
Alto	0	1	0	1	6.25%
Medio	1	0	1	2	12.5%
Bajo	2	5	1	8	50.0%
Informativo	1	2	2	5	31.25%
TOTAL	4	8	4	16	100%

Fuente: Elaboración Propia

Interpretación: Solo el 6.25% de las alertas son de alto riesgo, mientras que el 50% son de bajo riesgo, indicando que la mayoría de problemas identificados son de configuración más que vulnerabilidades críticas de código.

4.7.2 Evaluación según Common Vulnerability Scoring System (CVSS) v3.1

Para dimensionar objetivamente el impacto de los hallazgos, se aplicó el estándar CVSS v3.1 a las vulnerabilidades principales. Esta métrica permite establecer una jerarquía de remediación basada no solo en la facilidad de explotación, sino en el daño potencial a la confidencialidad, integridad y disponibilidad del Campus Virtual.

Tabla 27:Puntuación CVSS para vulnerabilidades técnicas principales

Vulnerabilidad	CVSS Estimado	Severidad	Confianza
Inyección SQL	9.8	CRÍTICA	Media
Ausencia de Tokens Anti-CSRF	8.0	ALTA	Alta
Cabecera CSP no configurada	6.8	MEDIA	Baja
Cookies sin HttpOnly/SameSite	4.5	MEDIA	Alta
Divulgación de información	3.5-4.0	BAJA	Media

Fuente: Elaboración Propia

CVSS Promedio: 5.05 (Severidad MEDIA)
Análisis: La Inyección SQL es la única vulnerabilidad crítica (9.8), mientras que las demás presentan severidad media o baja. El CVSS promedio de 5.05 indica un riesgo general moderado.

4.7.3 Análisis por dimensiones de seguridad (CIA)

El análisis de impacto se desglosa en función de las tres dimensiones fundamentales de la seguridad de la información: Confidencialidad, Integridad y Disponibilidad (CIA). Esta

segmentación permite visualizar qué pilares institucionales se verían más comprometidos en caso de una explotación exitosa, diferenciando entre el robo de información, la alteración de datos y la interrupción del servicio.

Tabla 28:Análisis por dimensiones de seguridad (CIA)

Vulnerabilidad	**Confidencialidad**	**Integridad**	**Disponibilidad**
Inyección SQL	ALTA	ALTA	ALTA
Ausencia Anti-CSRF	NULA	ALTA	NULA
CSP no configurada	BAJA	BAJA	NULA
Cookies inseguras	BAJA	BAJA	NULA
Divulgación información	BAJA	NULA	NULA

Fuente: Elaboración Propia

Interpretación: Solo la Inyección SQL afecta significativamente las tres dimensiones. La mayoría de vulnerabilidades tienen impacto bajo o nulo en disponibilidad.

4.7.4 Clasificación según OWASP ZAP Top 10 2021

Con el fin de alinear los resultados de esta auditoría con los estándares globales de seguridad de aplicaciones web, se mapearon las vulnerabilidades detectadas dentro de las categorías del OWASP Top 10 (versión 2021). Esta clasificación permite identificar qué tipos de riesgos predominan en la arquitectura del Campus Virtual y facilita la comparación con tendencias de seguridad globales.

Tabla 29:Categorización OWASP Top 10

Categoría OWASP	**Vulnerabilidades Incluidas**	**Cantidad**	**%**
A05: Security Misconfiguration	CSP, Cookies, Headers, Info Leak	7	63.6%

Categoría OWASP	Vulnerabilidades Incluidas	Cantidad	%
A01: Broken Access Control	Ausencia Anti-CSRF	1	9.1%
A03: Injection	Inyección SQL	1	9.1%
A10: SSRF	Gran Redirección	1	9.1%

Fuente: Elaboración Propia

Análisis: El 63.6% de las vulnerabilidades pertenecen a A05 (Configuración Insegura), indicando problemas sistémicos en la configuración del sistema más que fallas de código críticas.

4.7.5 Análisis cualitativo de riesgo y priorización

Con base en la evaluación técnica (CVSS) y el impacto en el negocio (CIA), se desarrolló una matriz de priorización para guiar las actividades de remediación. Esta clasificación no solo ordena las vulnerabilidades por su gravedad técnica, sino que asigna plazos de resolución (SLA - Service Level Agreement) acordes a la urgencia de proteger los activos críticos de la institución.

Tabla 30:Matriz de riesgo y priorización

Prioridad	Vulnerabilidades	Riesgo	Acción Requerida	Plazo
P1 - Crítica	Inyección SQL	ALTO	Parche inmediato + WAF	< 24 horas
P2 - Alta	Ausencia Anti-CSRF	MEDIO-ALTO	Implementar tokens CSRF	< 72 horas
P3 - Media	CSP no configurada	MEDIO	Configurar headers seguridad	< 1 semana
P4 - Baja	Cookies, Info Leak	BAJO	Hardening de configuración	< 2 semanas

Fuente: Elaboración Propia

4.8 Análisis Comparativo: Owasp Zap y DeepExploit

Con el fin de evaluar la complementariedad de las herramientas utilizadas, se realizó un análisis comparativo entre OWASP ZAP (enfocado en detección) y DeepExploit (enfocado en validación y explotación). Los resultados evidencian que ambas herramientas desempeñan roles distintos pero complementarios en el proceso de pentesting automatizado.

Tabla 31: Comparación de rendimiento y efectividad Owasp Zap vs DeepExploit

Criterio de Evaluación	**OWASP ZAP**	**DeepExploit**	**Análisis Comparativo**
Tipo de Análisis	DAST (Dynamic Application Security Testing) - Capa aplicación web	Explotación automatizada con IA - Capa infraestructura/servicios	Complementarios: ZAP analiza la aplicación (Moodle), DeepExploit la infraestructura subyacente (servidor)
Enfoque Principal	Detección de vulnerabilidades OWASP Top 10	Validación de explotabilidad mediante inteligencia artificial	Dimensión OWASP ZAP: Amplitud; Dimensión DeepExploit: Profundidad
Tiempo de Ejecución	15-20 minutos (escaneo completo configurado con umbral medio)	15.06 segundos (escaneo inicial Nmap) + 2-5 minutos para explotación automatizada	DeepExploit más rápido en validación específica; ZAP más exhaustivo en cobertura
Vulnerabilidades Detectadas	16 (clasificadas por severidad según OWASP ZAP)	2 validadas como explotables (inyección SQL +	ZAP: Mayor volumen de detección; DeepExploit: Menor volumen pero validación confirmada

Criterio de Evaluación	OWASP ZAP	DeepExploit	Análisis Comparativo
		SSH comprometido)	
Falsos Positivos	50% de alertas con confianza media requieren validación manual	0% (todas las vulnerabilidades validadas fueron explotadas exitosamente)	DeepExploit elimina falsos positivos mediante validación práctica de explotabilidad
Nivel de Criticidad Hallada	1 CRÍTICA (CVSS 9.8 - Inyección SQL) + 2 ALTAS + 13 MEDIAS/BAJAS	2 CRÍTICAS (inyección SQL confirmada, SSH comprometido - root/123456)	DeepExploit confirma las de mayor impacto; ZAP detecta el espectro completo
Precisión (vulns confirmadas / detectadas)	1/16 = 6.25% (solo la inyección SQL fue validada como explotable)	2/2 = 100% (todas las vulnerabilidades que DeepExploit atacó fueron explotadas)	DeepExploit superior en certeza; ZAP requiere validación externa
Eficiencia (vulns por minuto)	16 vulns / 17.5 min = 0.91 vulns/min (detección)	2 vulns / 5 min = 0.4 vulns/min (validación)	ZAP más eficiente en detección; DeepExploit más preciso en validación
Cobertura de Análisis	Aplicación web completa (Moodle -	Servicios específicos (SSH	Juntas: cobertura integral aplicación + infraestructura

Criterio de Evaluación	OWASP ZAP	DeepExploit	Análisis Comparativo
	formularios, sesiones, parámetros)	puerto 22, HTTP puerto 80)	
Uso de Inteligencia Artificial	No (escaneo basado en reglas predefinidas)	Sí (aprendizaje por refuerzo A3C para optimizar ataques)	DeepExploit introduce ventaja incremental: aprende del entorno y mejora iterativamente
Reportes Generados	Reportes estructurados exportables con clasificación por severidad y referencias OWASP	Logs detallados + integración con Metasploit para trazabilidad de explotación	ZAP: Mejor para documentación; DeepExploit : Mejor para evidencia forense
Curva de Aprendizaje	Baja-media (interfaz gráfica amigable)	Alta (requiere conocimiento de línea de comandos y Metasploit)	ZAP más accesible para equipos TI; DeepExploit requiere especialización

Fuente: Elaboración Propia

La Tabla 31 evidencia que OWASP ZAP y DeepExploit no son herramientas competitivas sino complementarias. Mientras ZAP destaca en la detección temprana y clasificación de vulnerabilidades web siguiendo el estándar OWASP Top 10, DeepExploit sobresale en la validación práctica de vulnerabilidades mediante inteligencia artificial y aprendizaje por refuerzo.

La principal fortaleza de OWASP ZAP radica en su capacidad para realizar un barrido exhaustivo de la aplicación web, identificando 16 vulnerabilidades potenciales y

clasificándolas según su severidad. Sin embargo, como se observa en la tabla, el 50% de sus alertas tienen confianza media, lo que requiere validación adicional.

DeepExploit, por su parte, demostró su valor al validar explotabilidad real en la infraestructura del servidor, comprometiendo el servicio SSH con credenciales débiles en cuestión de minutos. Su capacidad de aprendizaje automático permite optimizar los vectores de ataque, reduciendo significativamente los falsos positivos.

4.8.1 Análisis de métricas de rendimiento técnico

Para evaluar cuantitativamente la eficiencia de ambas herramientas, se establecieron las siguientes métricas de rendimiento técnico basadas en los resultados obtenidos durante la investigación:

Tabla 32:Métricas de Rendimiento

Métrica	**Fórmula**	**OWASP ZAP**	**DeepExploit**	**Interpretación**
Tasa de Detección	Vulnerabilidades encontradas / Tiempo total	16 / 17.5 min = 0.91 vulns/min	2 / 5 min = 0.4 vulns/min	ZAP detecta más rápido, pero DeepExploit valida lo que encuentra
Tasa de Confirmación	Vulnerabilidades validadas / Detectadas	1 / 16 = 6.25%	2 / 2 = 100%	DeepExploit solo valida lo que puede explotar, ZAP requiere validación externa

Métrica	Fórmula	OWASP ZAP	DeepExploit	Interpretación
Precisión Crítica	Vulnerabilidades críticas / Total	1 / 16 = 6.25%	2 / 2 = 100%	DeepExploit se enfoca en lo crítico, ZAP detecta todo el espectro
Eficiencia Energética	Peticiones por segundo / Impacto en servidor	Media (configuración de umbral medio)	Alta (IA optimiza ataques)	DeepExploit consume menos recursos al ser selectivo
Cobertura de Superficie	Capas analizadas	Aplicación web (Moodle)	Infraestructura (SSH, HTTP)	Complementarias: cubren aplicación + servidor
Tiempo por Vulnerabilidad Crítica	Tiempo total / N° críticas validadas	17.5 min / 0 = ∞ (no validó)	5 min / 2 = 2.5 min por crítica	DeepExploit altamente eficiente para validar lo crítico

Fuente: Elaboración Propia

4.8.2 Interpretación de los resultados comparativos

El análisis comparativo evidencia que OWASP ZAP y DeepExploit desempeñan roles distintos pero complementarios en el proceso de pentesting automatizado:

Dimensión OWASP ZAP (Amplitud):

- Proporciona una visión panorámica del estado de seguridad de la aplicación web
- Detecta un amplio espectro de vulnerabilidades (16 en total)
- Clasifica según estándares internacionales (OWASP Top 10, CVSS)

- Genera reportes accesibles para equipos de TI
- Limitación: 50% de sus alertas tienen confianza media y requieren validación manual

Dimensión DeepExploit (Profundidad):

- Se enfoca en validar la explotabilidad real de vulnerabilidades específicas
- Utiliza IA para optimizar ataques y reducir intentos redundantes
- Confirma con evidencia práctica (sesiones abiertas, credenciales comprometidas)
- **Fortaleza:** 100% de precisión en lo que valida
- **Limitación:** No realiza detección masiva, requiere objetivos específicos

4.8.3 Ventaja incremental de la inteligencia artificial

La incorporación de inteligencia artificial mediante DeepExploit representa una ventaja incremental significativa respecto a estudios previos que solo emplearon herramientas de detección como ZAP o Nessus. Mientras que en investigaciones como las de Zogaj et al. (2025) y Wenny & Pamuji (2024) los hallazgos requieren validación manual posterior, DeepExploit automatiza esta fase crítica, reduciendo el tiempo de confirmación de horas o días a minutos y proporcionando evidencia concreta de explotabilidad.

En términos cuantitativos, DeepExploit demostró una eficiencia de 2.5 minutos por vulnerabilidad crítica validada, mientras que OWASP ZAP, aunque detectó 16 vulnerabilidades, no pudo confirmar por sí mismo la explotabilidad de ninguna. Esta diferencia sustenta la recomendación de utilizar ambas herramientas de forma complementaria: ZAP para detección amplia y periódica, y DeepExploit para campañas específicas de validación profunda, especialmente en componentes críticos de la infraestructura.

4.9 Discusión de resultados

4.9.1 Vulnerabilidades críticas para el Campus Virtual

El análisis identificó que solo una vulnerabilidad entre las 16 alertas se clasifica como crítica para el Campus Virtual: la Inyección SQL. Esta vulnerabilidad presenta el mayor riesgo debido a su capacidad de comprometer completamente la base de datos académica. Su explotación permitiría acceso no autorizado a información sensible de estudiantes, docentes y contenido educativo. Aunque representa un porcentaje mínimo del total de alertas, su impacto potencial

es significativamente mayor que el de las demás vulnerabilidades combinadas. Las vulnerabilidades de medio riesgo complementan este panorama al facilitar vectores de ataque secundarios. Esta configuración de riesgos sugiere que la protección debe enfocarse prioritariamente en esta vulnerabilidad crítica.

4.9.2 Impacto potencial en la operación académica

La materialización del riesgo asociado a la Inyección SQL afectaría directamente la continuidad operativa del Campus Virtual, interrumpiendo el acceso a recursos educativos esenciales. Los datos académicos y personales quedarían expuestos, comprometiendo la privacidad de toda la comunidad universitaria. Institucionalmente, se generaría un deterioro significativo de la confianza depositada en la plataforma educativa. Operacionalmente, se requerirían esfuerzos considerables de contención, análisis forense y recuperación de servicios. Adicionalmente, se activarían obligaciones de notificación a usuarios afectados y autoridades competentes. Finalmente, este escenario demandaría la reasignación de recursos humanos y técnicos para la restauración completa de los servicios académicos.

CONCLUSIONES

1. El análisis de seguridad mediante pentesting automatizado identificó 16 vulnerabilidades en el Campus Virtual, donde solo una (inyección SQL) fue crítica (CVSS 9.8), demostrando que, aunque la superficie de ataque es limitada, existen puntos de entrada de alto impacto.

2. La combinación de OWASP ZAP (detección) y DeepExploit (explotación) validó la efectividad del pentesting automatizado, reduciendo falsos positivos y confirmando la explotabilidad de credenciales SSH débiles mediante inteligencia artificial.

3. La clasificación por CVSS v3.1 y OWASP Top 10 2021 reveló que el 63.6% de las vulnerabilidades pertenecen a "Configuración Insegura" (A05), indicando problemas sistémicos de implementación más que fallas de desarrollo.

4. El análisis de riesgo basado en probabilidad e impacto priorizó la inyección SQL como amenaza inmediata (P1), seguida de ausencia de controles anti-CSRF (P2), proporcionando una hoja de ruta clara para la remediación.

5. La evaluación por dimensiones CIA confirmó que la mayoría de vulnerabilidades afectan principalmente la confidencialidad e integridad, con impacto mínimo en disponibilidad, excepto en casos de explotación combinada.

6. El entorno virtualizado (Oracle VirtualBox detectado) proporcionó aislamiento adecuado para pruebas, pero también introdujo riesgos específicos de virtualización que deben considerarse en la postura de seguridad institucional.

7. El pentesting con DeepExploit demostró superioridad en eficiencia y profundidad frente a métodos tradicionales, validando el potencial de la IA para evaluaciones de seguridad en entornos educativos complejos.

RECOMENDACIONES

1. Implementar inmediatamente parches de seguridad para la vulnerabilidad crítica de inyección SQL, aplicando prepared statements y configurando un WAF temporal en las rutas afectadas del Campus Virtual.

2. Actualizar urgentemente las versiones vulnerables de OpenSSH 8.7 y nginx 1.20.1 a sus últimas versiones estables, aplicando todos los parches de seguridad pendientes para mitigar las CVE identificadas.

3. Establecer un programa periódico de pentesting automatizado que integre OWASP ZAP para detección y DeepExploit para validación, realizando evaluaciones trimestrales del Campus Virtual.

4. Implementar controles de seguridad específicos para OWASP A05, incluyendo configuración adecuada de headers HTTP (CSP, HSTS), hardening de cookies y eliminación de información sensible expuesta.

5. Desarrollar e implementar una política institucional de gestión de credenciales que prevenga el uso de contraseñas débiles y establezca autenticación multifactor para accesos administrativos.

6. Capacitar al personal técnico de la PUCE-Ibarra en el uso de herramientas de pentesting automatizado y en la interpretación de reportes basados en CVSS v3.1 y OWASP Top 10.

7. Crear un entorno de pruebas aislado y seguro para futuras evaluaciones de seguridad, considerando los riesgos específicos identificados en entornos virtualizados como Oracle VirtualBox.

8. Establecer un sistema de monitoreo continuo que detecte intentos de explotación de las vulnerabilidades identificadas, particularmente en los servicios SSH y HTTP expuestos.

BIBLIOGRAFÍA

Alasmary, H., Alharthi, F., & Alotaibi, M. (2020). Pentesting in educational institutions: A proactive approach to cybersecurity. *Journal of Information Security, 11*(2), 85-97. https://doi.org/10.4236/jis.2020.112007

Alasmary, W., Alharthi, F., & Alotaibi, M. (2020). Penetration testing methodologies for web applications: A comparative study. *International Journal of Advanced Computer Science and Applications, 11*(9), 547-553. https://doi.org/10.14569/IJACSA.2020.0110969

Alotaibi, B., & Alghazzawi, D. (2019). Penetration testing as a proactive security tool: A systematic review. *IEEE Access, 7*, 21079-21090. https://doi.org/10.1109/ACCESS.2019.2956412

Alotaibi, N., & Alghazzawi, D. (2019). Cybersecurity awareness in higher education institutions: Challenges and solutions. *International Journal of Advanced Computer Science and Applications, 10*(6), 256-263. https://doi.org/10.14569/IJACSA.2019.0100632

Choudhary, A., Kumar, V., & Singh, A. (2022). Automated penetration testing using machine learning techniques. *Journal of Information Security and Applications, 64*, 103063. https://doi.org/10.1016/j.jisa.2021.103063

Choudhary, A., Sharma, P., & Agarwal, R. (2022). DeepExploit: Intelligent automated penetration testing using machine learning. *Journal of Cybersecurity and Digital Trust, 4*(2), 25-33.

Costa, C., Yamaguchi, F., & Backes, M. (2020). A comprehensive approach to security testing: Applying OWASP ZAP in application security. *ACM Transactions on Privacy and Security, 23*(4), Article 18. https://doi.org/10.1145/3410156

Dougiamas, M., & Taylor, P. (2003). Moodle: Using learning communities to create an open source course management system. *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 171-178). AACE.

Essential Steps to Use OWASP ZAP for Penetration Testing. (2024). *6 Steps Security*. https://www.6stepszap.com

FIRST. (2021). *Common Vulnerability Scoring System v3.1: Specification document*. Forum of Incident Response and Security Teams. https://www.first.org/cvss/v3.1/specification-document

Hernández-Sampieri, R., Fernández-Collado, C., & Baptista-Lucio, P. (2018). *Metodología de la investigación* (6.ª ed.). McGraw-Hill Education.

Kaur, M., Singh, A., & Sharma, P. (2020). Penetration testing in higher education institutions: Challenges and approaches. *Journal of Information Security, 11*(3), 210-221. https://doi.org/10.4236/jis.2020.113014

Kaur, P., Singh, A., & Kaur, P. (2020). A comprehensive review of penetration testing tools and techniques. *Journal of Cybersecurity and Privacy, 1*(1), 45-61. https://doi.org/10.3390/jcp1010004

Khan, M., & Brohi, S. N. (2020). Cybersecurity challenges for educational institutions. *Journal of Cybersecurity and Privacy, 1*(1), 68-86. https://doi.org/10.3390/jcp1010005

Khan, R., & Brohi, S. N. (2020). Security assessment of web applications using penetration testing. *Journal of Information Security, 11*(2), 85-98. https://doi.org/10.4236/jis.2020.112007

Kumar, A., Singh, R., & Kaur, P. (2021). Enhancing security in educational institutions using penetration testing: A case study of virtual campuses. *Information Security Journal: A Global Perspective, 30*(1), 22-30. https://doi.org/10.1080/19393555.2020.1857570

Kumar, S., Singh, R., & Verma, P. (2021). Application of CVSS in prioritizing security vulnerabilities in academic institutions. *International Journal of Computer Applications, 182*(45), 34-42. https://doi.org/10.5120/ijca2021920948

Moodle. (2023). *Moodle security overview*. https://docs.moodle.org/403/en/Security_overview

National Institute of Standards and Technology. (2008). *Technical guide to information security testing and assessment* (Special Publication 800-115). https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

OWASP Foundation. (2020). *OWASP Zed Attack Proxy*. https://www.zaproxy.org/

OWASP Foundation. (2021). *OWASP Top 10: 2021*. https://owasp.org/Top10/

OWASP Foundation. (2021). *OWASP Testing Guide v4.2*. https://owasp.org/www-project-web-security-testing-guide/

Pandey, N., & Bhatt, C. (2020). Vulnerability assessment using CVSS in higher education institutions. *International Journal of Network Security & Its Applications, 12*(6), 45-58. https://doi.org/10.5121/ijnsa.2020.12604

Pandey, S., & Bhatt, S. (2020). Risk-based vulnerability management using CVSS. *International Journal of Information Security Science, 9*(3), 112-120. https://doi.org/10.28945/4621

Patel, R., & Desai, S. (2020). Web application security testing in higher education: Using OWASP ZAP for practical evaluation. *International Journal of Computer Science and Information Security, 18*(5), 45-53. https://doi.org/10.5281/zenodo.4281234

Patil, V., & Kulkarni, R. (2021). An overview of passive reconnaissance techniques in penetration testing. *International Journal of Computer Trends and Technology, 69*(4), 12-17. https://doi.org/10.14445/22312803/IJCTT-V69I4P103

Penetration Testing Execution Standard. (s. f.). *PTES Technical Guidelines*. http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines

Ramya, K., & Sridevi, K. (2021). A comprehensive study of active and passive penetration testing approaches. *International Journal of Engineering Research & Technology, 10*(7), 441-445. https://doi.org/10.17577/IJERTV10IS070001

Secretaría Nacional de Planificación y Desarrollo. (2024). *Plan de Desarrollo para el Nuevo Ecuador 2024-2025*. https://www.planificacion.gob.ec/plan-de-desarrollo-2024-2025/

Sharma, V., Mehta, P., & Agarwal, S. (2020). Temporal metrics in CVSS: Enhancing vulnerability prioritization. *International Journal of Information Security Science, 9*(2), 15-24. https://doi.org/10.28945/4548

Stallings, W. (2020). *Network security essentials: Applications and standards* (7th ed.). Pearson Education.

Wenny, R., & Pamuji, F. Y. (2024). Perbandingan evaluasi kerentanan menggunakan Tenable Nessus Scanner dan OWASP Zed Attack Proxy untuk meningkatkan keamanan sistem informasi kepegawaian di Universitas Merdeka Malang. *Jurnal Ilmiah Universitas Batanghari Jambi, 24*(3), 2451-2457. https://doi.org/10.33087/jiubj.v24i3.5488

Zareen, M., & Nasir, A. (2019). Addressing cybersecurity challenges in higher education institutions. *International Journal of Information Security, 18*(3), 147-159. https://doi.org/10.1007/s10207-018-0414-4

Zogaj, G., Ismaili, F., Idrizi, E., & Luma, A. (2025). Statistical analysis of unique web application vulnerabilities: A quantitative assessment of scanning tool efficiency. *SEEU Review, 20*(1), 136-152. https://doi.org/10.2478/seeur-2025-0021

ANEXOS

Anexo 1: Carta de Autorización

Pontificia Universidad Católica del Ecuador
Seréis mis testigos

IBARRA UNIDAD DE TECNOLOGÍAS INFORMÁTICAS

CARTA DE AUTORIZACIÓN

Ibarra, 20 de enero de 2025

Señorita
Angie Nayeli Landázuri Rodríguez
ESTUDIANTE DE LA CARRERA DE TECNOLOGIAS DE LA INFORMACIÓN DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR IBARRA

De mi consideración:

Por la presente, autorizo la elaboración del proyecto de tesis **"Vulnerabilidades del Campus Virtual de la Pontificia Universidad Católica del Ecuador – Ibarra mediante la técnica de pentesting (OWASP ZAP Y DEEPEXPLOIT) ".**

Este proyecto cuenta con el visto favorable de la Unidad de Tecnología y deberá desarrollarse en estricto cumplimiento de los requisitos académicos establecidos. Asimismo, se establece el compromiso de:

- **Garantizar el análisis:** Realizar un análisis exhaustivo de las vulnerabilidades identificadas y proponiendo medidas concretas para su mitigación. .
- **Realizar en Ambiente controlado:** Todas las pruebas y análisis se realizarán en un entorno virtualizado, aislado de los sistemas de producción, para garantizar la seguridad del entorno real.
- **Informe de Vulnerabilidades:** Entregar los resultados de manera clara y documentada.
- **Manejo de confidencialidad:** Que toda la información analizada o descubierta durante el proyecto será tratada de manera confidencial y solo será accesible para las partes autorizadas.

Particular que pongo en su conocimiento para los fines pertinente, quedo a disposición para cualquier consulta.

Atentamente,

Mgs. Patricio Ruiz
Responsable de Tecnologías Educativas

Msc. Franklin Sánchez
Jefe Unidad de Tecnologías

Dirección: Av. Jorge Guzmán Rueda y Av. Aurelio Espinosa Pólit. Ciudadela "La Victoria". Teléf: (593-6) 2 994-700 Ext. 1000
Ibarra - Ecuador / www.pucesi.edu.ec

Anexo 2: Carta de Aceptación

Pontificia Universidad Católica del Ecuador
Seréis mis testigos

IBARRA UNIDAD DE TECNOLOGÍAS INFORMÁTICAS

CARTA DE ACEPTACIÓN

Ibarra, 05 de febrero de 2026

Señorita
Angie Nayeli Landázuri Rodríguez
ESTUDIANTE DE LA CARRERA DE TECNOLOGIAS DE LA INFORMACIÓN DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR IBARRA

De mi consideración:

La Unidad de Tecnología de la Pontificia Universidad Católica del Ecuador – Ibarra, emite el presente documento para certificar formalmente que la señorita **Angie Nayeli Landázuri Rodríguez**, estudiante de la carrera de Tecnologías de la Información, ha concluido y entregado a nuestra entera satisfacción el proyecto de tesis titulado: **"Vulnerabilidades del Campus Virtual de la Pontificia Universidad Católica del Ecuador – Ibarra mediante la técnica de pentesting (OWASP ZAP Y DEEPEXPLOIT) "**.

Hacemos constar que, durante la ejecución de su investigación, la estudiante cumplió cabalmente con todas las directrices estipuladas en el acuerdo de autorización. Las pruebas de seguridad se desarrollaron de manera rigurosa en un entorno controlado y aislado, protegiendo en todo momento la integridad de nuestra infraestructura.

Expresamos nuestra total conformidad con los resultados obtenidos, ya que representan un aporte significativo para el fortalecimiento informático de nuestra institución. Reconocemos la destacada capacidad técnica, el profesionalismo y la responsabilidad evidenciados por la estudiante a lo largo de este proceso.

Se extiende la presente certificación para los fines académicos y legales que la interesada considere pertinentes.

Atentamente,

Mgs. Patricio Ruiz
Responsable de Tecnologías Educativas

Msc. Franklin Sánchez
Jefe Unidad de Tecnologías

Dirección: Av. Jorge Guzmán Rueda y Av. Aurelio Espinosa Pólit. Ciudadela "La Victoria". Teléf: (593-6) 2 994-700 Ext. 1000
Ibarra - Ecuador / www.pucesi.edu.ec

Anexo 3: Informe generado por DeepExploit

Nmap Scan Report - Scanned at Tue Feb 3 21:42:38 2026

Scan Summary

Nmap 7.95 was initiated at Tue Feb 3 21:42:38 2026 with these arguments:*/usr/lib/nmap/nmap --privileged -Pn -sV -O -oX reporte.xml 172.16.19.165*

Verbosity: 0; Debug level 0

Nmap done at Tue Feb 3 21:42:53 2026; 1 IP address (1 host up) scanned in 15.06 seconds

172.16.19.165(online)

Address

- 172.16.19.165 (ipv4)

Ports

The 998 ports scanned but not shown below are in state: filtered

- 998 ports replied with: no-response

Port		State	Service	Reason	Product	Version	Extra info
22	tcp	open	ssh	syn-ack	OpenSSH	8.7	protocol 2.0
80	tcp	open	http	syn-ack	nginx	1.20.1	

Remote Operating System Detection

Used port: 22/tcp (open)

OS match: Oracle Virtualbox Slirp NAT bridge (98%)

OS match: AT&T BGW210 voice gateway (95%)

OS match: QEMU user mode network gateway (94%)

Misc MetricsMetric	Value
Ping Results	user-set
TCP Sequence Prediction	Difficulty=16 (Good luck!)
IP ID Sequence Generation	Incremental