# UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

**FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES**

**CARRERA DE COMPUTACIÓN**

**Tema: "Aplicación de técnicas de pentesting para la evaluación de vulnerabilidades"**

Trabajo de Integración Curricular previo a la obtención del título de Ingeniero en Ciencias de la Computación

AUTOR: Changuan Rodríguez Roberth Guillermo

TUTOR: Ing. Del Hierro Mosquera Milton Gabriel MSc

Tulcán, 2026.

**CERTIFICADO DEL TUTOR**

Certifico que el estudiante Changuan Rodríguez Roberth Guillermo con el número de cédula 1002672614 ha desarrollado el Trabajo de Integración Curricular: "Aplicación de técnicas de pentesting para la evaluación de vulnerabilidades"

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de la Unidad de Integración Curricular, Titulación e Incorporación de la UPEC, por lo tanto, autorizo la presentación de la sustentación para la calificación respectiva


Ing. Del Hierro Mosquera Milton Gabriel MSc

**TUTOR**

Tulcán, enero de 2026

**AUTORÍA DE TRABAJO**

El presente Trabajo de Integración Curricular constituye un requisito previo para la obtención del título de Ingeniero en la Carrera de computación de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Yo, Changuan Rodríguez Roberth Guillermo con cédula de identidad número 1002672614 declaro que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

_____

Changuan Rodríguez Roberth Guillermo

**AUTOR**

Tulcán, enero de 2026

**ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE INTEGRACIÓN CURRICULAR**

Yo, Changuan Rodríguez Roberth Guillermo declaro ser autor de los criterios emitidos en el Trabajo de Integración Curricular: "Aplicación de técnicas de pentesting para la evaluación de vulnerabilidades" y eximo expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes de posibles reclamos o acciones legales.

_____

Changuan Rodríguez Roberth Guillermo

**AUTOR**

Tulcán, enero de 2026

## AGRADECIMIENTO

Agradezco, en primer lugar, a la Universidad Politécnica Estatal del Carchi, por brindar el espacio académico y los recursos necesarios para el desarrollo de esta investigación.

Un reconocimiento especial al docente tutor, por su guía, criterio técnico y observaciones oportunas, que permitieron orientar el trabajo con rigor metodológico y enfoque profesional.

A las autoridades y personal del área de Tecnologías de la Información, por facilitar el acceso a la infraestructura y por la colaboración brindada durante la ejecución de las pruebas y el levantamiento de información.

Finalmente, a mi familia, por el apoyo constante, la paciencia y la confianza depositada a lo largo de este proceso académico, que fue clave para culminar esta etapa de formación profesional.

## DEDICATORIA

El presente trabajo está dedicado, a mi familia, por el apoyo incondicional brindado a lo largo de mi formación académica. Su esfuerzo, paciencia y confianza fueron el pilar fundamental para superar cada etapa de este proceso, incluso en los momentos de mayor exigencia.

Asimismo, esta dedicatoria se extiende a todas las personas que, de manera directa o indirecta, contribuyeron a mi crecimiento personal y profesional, brindando palabras de aliento, orientación y motivación.

Este logro representa no solo la culminación de una etapa académica, sino también el resultado de la constancia, el compromiso y el aprendizaje adquirido durante el camino recorrido.

## ÍNDICE

## ÍNDICE DE TABLAS



## ÍNDICE DE FIGURAS

## ÍNDICE DE ANEXOS

**RESUMEN**

La presente investigación tiene como objetivo evaluar las vulnerabilidades presentes en los sistemas informáticos y la infraestructura tecnológica de la Universidad Politécnica Estatal del Carchi (UPEC), mediante la aplicación de técnicas estructuradas de pentesting. Para ello, se utilizó la metodología PTES, compuesta por fases como reconocimiento, modelado de amenazas, análisis de vulnerabilidades, explotación controlada y documentación de resultados. Durante el proceso se identificaron activos expuestos, subdominios vulnerables, versiones desactualizadas de tecnologías y encabezados HTTP mal configurados. Además, se aplicaron marcos de análisis como STRIDE y MITRE ATTCCK para categorizar riesgos y priorizar vectores de amenaza. Las herramientas empleadas incluyeron Nmap, OpenVAS, Metasploit, Burp Suite y Wireshark, lo cual permitió realizar un análisis técnico y económico viable dentro del entorno universitario. Los resultados muestran que la UPEC presenta falencias significativas en su postura de seguridad, careciendo de una estrategia proactiva de evaluación de riesgos. Se concluye que el pentesting estructurado es una práctica eficiente para identificar y mitigar vulnerabilidades, por lo que se recomienda integrarlo en las auditorías de seguridad regulares y promover la capacitación continua del personal técnico en ciberseguridad.

**Palabras Claves:** Pentesting, vulnerabilidades, seguridad informática, PTES, MITRE ATTCCK.

**ABSTRACT**

The present research aims to evaluate the vulnerabilities in the computer systems and technological infrastructure of the Universidad Politécnica Estatal del Carchi (UPEC), through the application of structured penetration testing techniques. For this purpose, the PTES methodology was used, which comprises phases such as reconnaissance, threat modeling, vulnerability analysis, controlled exploitation, and documentation of results. During the process, exposed assets, vulnerable subdomains, outdated technology versions, and misconfigured HTTP headers were identified. In addition, analytical frameworks such as STRIDE and MITRE ATT&CK were applied to categorize risks and prioritize threat vectors. The tools used included Nmap, OpenVAS, Metasploit, Burp Suite, and Wireshark, which allowed for a viable technical and economic analysis within the university environment. The results show that UPEC has significant shortcomings in its security posture, including a lack of a proactive risk assessment strategy. It is concluded that structured pentesting is an efficient practice for identifying and mitigating vulnerabilities, and it is recommended to integrate it into regular security audits and to promote continuous cybersecurity training for technical staff.

**Keywords:** Pentesting, vulnerabilities, cybersecurity, PTES, MITRE ATT&CK

# INTRODUCCIÓN

En la actual era digital, la protección de la información se ha transformado en un elemento esencial para asegurar la integridad, disponibilidad y privacidad de los datos en cualquier entidad. Este reto es aún más significativo en entidades educativas como las universidades, donde la infraestructura tecnológica resguarda datos delicados vinculados a alumnos, profesores, investigaciones y procedimientos administrativos. En estas circunstancias, la Universidad Politécnica Estatal del Carchi (UPEC), situada en la frontera norte de Ecuador, se enfrenta a amenazas cibernéticas en aumento debido a la ausencia de tácticas proactivas en términos de seguridad informática.

A pesar de contar con sistemas tecnológicos operativos, la UPEC no tiene implementado un proceso institucionalizado de evaluación de vulnerabilidades ni personal especializado exclusivamente en ciberseguridad. Esto ha generado una postura de defensa pasiva frente a posibles ataques cibernéticos, exponiendo activos críticos como servidores académicos, aplicaciones web internas y servicios administrativos. Esta situación refleja tendencias observadas en otras universidades ecuatorianas, donde la seguridad informática suele ser relegada a un segundo plano hasta que ocurre un incidente grave.

Ante este escenario, surge la necesidad de adoptar prácticas proactivas que permitan identificar y mitigar riesgos antes de que sean explotados por actores malintencionados. Es aquí donde el pentesting, o pruebas de penetración controladas, se presenta como una solución viable y eficiente. Esta metodología permite simular ataques reales sobre la infraestructura informática de la universidad, identificando puntos débiles y ofreciendo recomendaciones técnicas para su fortalecimiento. Su aplicación no solo ayuda a mejorar la postura de seguridad, sino que también fomenta la capacitación técnica del personal encargado de la gestión de redes y sistemas.

La presente investigación busca aplicar técnicas de pentesting para evaluar las vulnerabilidades presentes en los sistemas informáticos de la UPEC, siguiendo la metodología PTES (Penetration Testing Execution Standard). Se espera demostrar que esta práctica es efectiva para detectar debilidades en la infraestructura tecnológica universitaria, incluso bajo recursos limitados y utilizando herramientas de código abierto.

Como hipótesis principal, se plantea que la infraestructura tecnológica de la UPEC contiene vulnerabilidades significativas que podrían ser explotadas por atacantes externos e internos, debido a la falta de políticas estructuradas de evaluación de seguridad. Como objetivos específicos, se pretende:

- Identificar activos informáticos críticos dentro de la infraestructura universitaria.
- Aplicar técnicas de reconocimiento pasivo y activo para mapear la superficie de ataque.
- Realizar un análisis de vulnerabilidades mediante herramientas automatizadas y verificación manual.
- Simular escenarios de ataque controlados para evaluar el impacto potencial.
- Proponer medidas de reducción y buenas prácticas para mejorar la seguridad institucional.

Esta investigación no solo busca contribuir al fortalecimiento de la seguridad informática en la UPEC, sino también generar conocimiento aplicable a otras instituciones educativas en Ecuador y América Latina.

# I. EL PROBLEMA

## 1.1. PLANTEAMIENTO DEL PROBLEMA

En el mundo digital actual, las organizaciones dependen cada vez más de sus sistemas informáticos para realizar sus operaciones diarias. Sin embargo, la creciente complejidad de estos sistemas y la constante evolución de las amenazas cibernéticas han incrementado significativamente el riesgo de vulnerabilidades de seguridad. Las configuraciones incorrectas, el uso de software desactualizado y la falta de medidas de seguridad adecuadas pueden exponer a las organizaciones a ataques cibernéticos que pueden comprometer la integridad, confidencialidad y disponibilidad de la información.

A nivel global, la ciberseguridad se ha convertido en una preocupación crítica para gobiernos, empresas y organizaciones de todos los tamaños. Según el Foro Económico Mundial, los ciberataques y el robo de datos se encuentran entre los cinco principales riesgos globales en términos de probabilidad (Schwab, 2022). La pandemia de COVID-19 ha acelerado la digitalización en todo el mundo, exponiendo aún más las vulnerabilidades de los sistemas informáticos y aumentando la superficie de ataque para los ciberdelincuentes (Interpol, 2023).

En América Latina, la situación de la ciberseguridad presenta desafíos particulares. La región ha experimentado un aumento significativo en el número y sofisticación de los ciberataques en los últimos años. Factores como la posible falta de profesionales calificados en ciberseguridad y la potencial insuficiente inversión en tecnologías de protección podrían contribuir a la vulnerabilidad de la región.

En Ecuador, la problemática de la ciberseguridad refleja tendencias regionales, pero con matices propios.

El país ha experimentado un incremento en los incidentes de seguridad informática, especialmente en sectores críticos como el financiero y el gubernamental. Según un estudio realizado por la Escuela Politécnica Nacional, más del 60% de las empresas

ecuatorianas han sufrido algún tipo de ciberataque en los últimos dos años, siendo el phishing y el malware las amenazas más comunes (Escuela Politécnica Nacional, 2023.

La Agencia de Regulación y Control de las Telecomunicaciones (ARCOTEL) de Ecuador ha identificado que muchas organizaciones en el país carecen de políticas de seguridad informática adecuadas y no realizan evaluaciones regulares de sus sistemas, lo que las hace más vulnerables a ataques (ARCOTEL, 2023). Además, la falta de conciencia sobre la importancia de la ciberseguridad entre los empleados y la alta dirección de las empresas ecuatorianas agrava el problema.

El marco legal y regulatorio en Ecuador relacionado con la ciberseguridad está en desarrollo, pero aún presenta brechas significativas. La Ley Orgánica de Protección de Datos Personales, aprobada en 2021, representa un avance importante, pero su implementación efectiva sigue siendo un desafío para muchas organizaciones (Asamblea Nacional del Ecuador, 2021).

En el ámbito educativo y de formación profesional, Ecuador enfrenta una escasez de especialistas en ciberseguridad. Un informe del Ministerio de Telecomunicaciones y de la Sociedad de la Información (MINTEL) señala que existe una brecha significativa entre la demanda de profesionales de ciberseguridad y la oferta disponible en el mercado laboral ecuatoriano (MINTEL, 2023).

Este panorama subraya la necesidad urgente de que las organizaciones ecuatorianas adopten un enfoque más proactivo y estratégico hacia la ciberseguridad. Esto implica no solo la implementación de soluciones tecnológicas avanzadas, sino también la formación continua del personal, la actualización de políticas y procedimientos de seguridad, y la colaboración estrecha con entidades gubernamentales y del sector privado para fortalecer la resiliencia cibernética del país.

La Universidad Politécnica Estatal del Carchi (UPEC) depende de diversos servicios informáticos para la gestión académica, administrativa y operativa. Sin embargo, la infraestructura tecnológica existente no cuenta con procesos sistemáticos de evaluación de seguridad, lo que genera un panorama incierto respecto al nivel real de exposición de sus sistemas internos.

Aunque la universidad dispone de herramientas básicas de protección, como firewalls y configuraciones estándares, no se ha realizado un análisis técnico que permita determinar si estas medidas son suficientes para proteger la red ante amenazas internas o externas. Esta ausencia de evaluaciones periódicas limita la capacidad institucional para anticipar riesgos, aplicar mejoras o responder de forma adecuada frente a vulnerabilidades que puedan comprometer servicios críticos como plataformas académicas, sistemas administrativos, portales web y servidores internos.

## 1.2. FORMULACIÓN DEL PROBLEMA

¿De qué manera efectiva se puede aplicar el pentesting para evaluar las vulnerabilidades en las redes y sistemas informáticos de la Universidad Politécnica Estatal del Carchi y mejorar la seguridad de la información?

## 1.3. JUSTIFICACIÓN

Hoy en día, casi todo lo que hacemos depende de sistemas informáticos conectados a internet. En las universidades no es diferente: se manejan notas, registros, datos personales, plataformas virtuales, correos institucionales, etc. Pero muchas veces se da por hecho que todo está "seguro", cuando en realidad no se han hecho pruebas reales para comprobarlo. Ahí es donde entra el pentesting.

Esta investigación tiene como propósito poner a prueba la seguridad real de los sistemas públicos de la UPEC usando herramientas y metodologías profesionales, pero adaptadas a un entorno como el nuestro. No se trata solo de ver si hay fallas, sino de demostrar cómo se podrían aprovechar esas debilidades, y, sobre todo, cómo se pueden prevenir incidentes antes de que sucedan.

El principal beneficiario será la Universidad Politécnica Estatal del Carchi. Con esta investigación no solo se entregará un informe con las vulnerabilidades encontradas, sino también una guía con soluciones concretas. Esto puede servirle a la Dirección de TIC como punto de partida para mejorar su estrategia de seguridad. A la larga, también se beneficia toda la comunidad universitaria: estudiantes, docentes y administrativos, porque se protege la información que todos manejamos.

Más allá de lo técnico, esta investigación puede abrir una conversación dentro de la UPEC sobre la importancia de realizar evaluaciones de seguridad de forma regular, algo que muchas instituciones aún no hacen por desconocimiento o falta de

recursos. Justamente por eso usé herramientas de código abierto y enfoques prácticos que pueden ser replicados fácilmente. Es decir, esto no se queda en la teoría, sino que se puede aplicar en la vida real.

Además, esta investigación busca llenar un espacio que no está muy explorado: cómo implementar una metodología internacional como PTES en una universidad pública ecuatoriana, con lo que tenemos a mano. Creo que eso le da valor tanto desde el lado técnico como desde el lado académico, porque puede servir de referencia para otros estudiantes, docentes o equipos TIC que quieran hacer algo parecido.

En cuanto a la viabilidad, el proyecto ya está encaminado. La universidad me dio el visto bueno, las herramientas que usé son libres y no hubo necesidad de inversiones grandes. Todo se hizo bajo un enfoque ético y con cuidado de no afectar ningún servicio.

## 1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

### 1.4.1. Objetivo General

Evaluar técnicas de pentesting para la evaluación de vulnerabilidades en la Universidad Politécnica Estatal del Carchi.

### 1.4.2. Objetivos Específicos

Fundamentar bibliográficamente la evaluación de vulnerabilidades de una red, utilizando fuentes académicas y técnicas confiables para la sustentación de la investigación.

Estimar la efectividad de los controles de seguridad implementados mediante la realización de pruebas de intrusión controladas, utilizando herramientas especializadas para identificación y explotación de vulnerabilidades de manera efectiva en la red de la Universidad Politécnica Estatal del Carchi.

Desarrollar un informe que documente los procesos, herramientas y mejores prácticas para la evaluación de la seguridad de la red.

### 1.4.3. Preguntas de Investigación

¿Cuáles son las metodologías y enfoques utilizados en la evaluación de posibles vulnerabilidades de una red?

¿Qué herramientas especializadas de pentesting están disponibles y cómo se utilizan para identificar y explotar vulnerabilidades en una red?

¿Cuál es la efectividad de las pruebas de intrusión controladas en la evaluación de los controles de seguridad implementados en una red?

# II. FUNDAMENTACIÓN TEÓRICA

## 2.1. ANTECEDENTES DE LA INVESTIGACIÓN

En una tesis de la Universidad de las Américas, se investigó la vulnerabilidad de los portales web en instituciones educativas ecuatorianas. El estudio abordó la creciente preocupación por el cibercrimen y la necesidad de proteger la información en sitios web institucionales. Se emplearon metodologías como OWASP y recomendaciones de NIST para evaluar y mejorar la seguridad informática (Añazco & Ortiz, 2020).

Otra investigación, realizada en la Universidad Estatal del Sur de Manabí, se centró en la aplicación del hacking ético para identificar amenazas y vulnerabilidades en la red universitaria. Se destacó la importancia de la capacitación del personal administrativo en seguridad informática y se identificaron herramientas efectivas para evaluar la seguridad de la red (Briones, 2020).

Un trabajo de grado de la Universidad Piloto de Colombia exploró el hacking ético como herramienta para la seguridad informática. Se detalló un proceso sistemático para realizar pruebas de vulnerabilidad, enfatizando la importancia de la planificación y el reconocimiento pasivo (Medina, 2021).

En Cuba, un estudio publicado en la Revista Cubana de Informática Médica analizó las herramientas fundamentales para el hacking ético. Se destacaron software como Nmap y OpenVas, subrayando la importancia del conocimiento y la interpretación en la seguridad informática (Rodríguez, 2020).

Por otro lado, Alarcón (2022) hizo pentesting en la red de Salud Valle del Mantaro, que hace uso de 3 sistemas administrados por el área de informática, utilizando el sistema operativo Parrot con sus herramientas integradas y scripts creados por la comunidad de GitHub, con su prueba logro reducir las vulnerabilidades en un 68%, mejorando la seguridad de los sistemas.

Una tesis de la Universidad Estatal de Manabí propuso la aplicación del hacking ético para mejorar la seguridad en la red de una organización. Se implementaron tres

etapas fundamentales del hacking ético y se recomendó realizar pruebas periódicas y capacitar al personal técnico (Chilán, 2022).

Un proyecto de tesis de la Universidad Técnica de Ambato aplicó el hacking ético para evaluar la seguridad informática en una empresa industrial. Se identificaron debilidades en equipos con Linux y se recomendó realizar auditorías de hacking ético de manera regular (Rojas Buenaño, 2020).

Por último, Acosta (2022) hizo una auditoría de seguridad en un entorno controlado, las pruebas las hizo en Windows y Ubuntu, usando herramientas de código abierto, utilizó la metodología OWASP para comparar y analizar los resultados de su auditoría, con posibles soluciones y recomendaciones.

## 2.2. MARCO TEÓRICO

### 2.2.1. Seguridad Informática

La seguridad informática se refiere al conjunto de medidas y procedimientos implementados para proteger la información digital y los sistemas informáticos contra amenazas, ataques y accesos no autorizados (Gómez Vieites, 2020). Este campo multidisciplinario abarca diversas áreas, incluyendo tecnología, procesos y factor humano, con el objetivo de salvaguardar los activos informáticos de una organización o individuo.

Según Costas Santos (2022), los conceptos básicos de la seguridad informática incluyen:

- Amenaza: Cualquier elemento o acción capaz de atentar contra la seguridad informática.
- Vulnerabilidad: Debilidad en un sistema que puede ser explotada por una amenaza.
- Riesgo: Probabilidad de que una amenaza se materialice, aprovechando una vulnerabilidad.
- Control: Medida de protección para mitigar riesgos y reducir vulnerabilidades.

En la era digital actual, la seguridad informática ha adquirido una importancia crítica. Díaz Orueta et al. (2020) señalan que la creciente dependencia de las tecnologías de la información en todos los ámbitos de la sociedad ha amplificado la necesidad de proteger los datos y sistemas informáticos. Algunos factores que subrayan su relevancia son:

- Aumento de ciberataques: El número y sofisticación de los ataques informáticos han crecido exponencialmente en los últimos años (Gómez Vieites, 2020).
- Regulaciones y cumplimiento: Nuevas leyes y regulaciones, como el Reglamento General de Protección de Datos (RGPD) en Europa, exigen a las organizaciones implementar medidas de seguridad robustas (Costas Santos, 2022).
- Transformación digital: La adopción masiva de tecnologías como la nube, el Internet de las Cosas (IoT) y la inteligencia artificial ha ampliado la superficie de ataque, requiriendo nuevas estrategias de seguridad (Díaz Orueta et al., 2020).
- Impacto económico: Las brechas de seguridad pueden resultar en pérdidas financieras significativas, daños reputacionales y sanciones legales (Gómez Vieites, 2020).

Los pilares de la seguridad informática, conocidos como la tríada CIA (por sus siglas en inglés), son fundamentales para comprender y aplicar las medidas de protección adecuadas. Estos principios, según Costas Santos (2022), son:

- Confidencialidad: Garantiza que la información solo sea accesible para las personas o sistemas autorizados. Como señala Gómez Vieites (2020), "la confidencialidad implica la protección de datos y comunicaciones frente a accesos no autorizados" (p. 45).
- Integridad: Asegura que la información no sea alterada de manera no autorizada, manteniendo su exactitud y completitud. Díaz Orueta et al. (2020) enfatizan que "la integridad de los datos es crucial para mantener la confianza en los sistemas de información y garantizar la toma de decisiones basada en datos precisos" (p. 78).
- Disponibilidad: Garantiza que la información y los recursos estén accesibles para los usuarios autorizados cuando los necesiten. Costas Santos (2022) afirma que "la disponibilidad es esencial para mantener la continuidad del negocio y evitar interrupciones en los servicios críticos" (p. 62).

Estos principios fundamentales son interdependientes y deben considerarse en conjunto al diseñar e implementar estrategias de seguridad informática. Como concluye Gómez Vieites (2020), "el equilibrio entre confidencialidad, integridad y

disponibilidad es clave para una seguridad informática efectiva y adaptada a las necesidades específicas de cada organización" (p. 103).

### 2.2.2. Vulnerabilidades en Sistemas Informáticos

En el contexto de la seguridad informática, una vulnerabilidad se refiere a una debilidad o fallo en un sistema de información que puede ser explotado por una amenaza, comprometiendo la seguridad del sistema (Gómez Vieites, 2020). Estas debilidades pueden existir en diversos componentes del sistema, incluyendo hardware, software, procesos o incluso en el comportamiento de los usuarios.

Según Costas Santos (2022), "una vulnerabilidad es cualquier característica o circunstancia de un sistema informático que pueda ser aprovechada por un atacante para causar un daño o perjuicio en la confidencialidad, integridad o disponibilidad de la información" (p. 87).

Las vulnerabilidades en sistemas informáticos pueden clasificarse en tres categorías principales:

- Vulnerabilidades de software:
  Estas son las más comunes y pueden surgir debido a errores de programación, configuraciones incorrectas o falta de actualizaciones. Díaz Orueta et al. (2020) señalan que "las vulnerabilidades de software pueden incluir desbordamientos de búfer, inyecciones SQL, cross-site scripting (XSS), entre otras" (p. 124).
- Vulnerabilidades de hardware:
  Se refieren a debilidades en los componentes físicos de un sistema. Gómez Vieites (2020) menciona que "las vulnerabilidades de hardware pueden incluir fallos en el diseño de chips, puertas traseras implementadas por fabricantes o debilidades en los protocolos de comunicación de los dispositivos" (p. 156).
- Vulnerabilidades humanas:
  Estas vulnerabilidades están relacionadas con el factor humano en la seguridad informática. Costas Santos (2022) afirma que "las vulnerabilidades humanas pueden ser el resultado de la falta de concienciación en seguridad, el incumplimiento de políticas o la susceptibilidad a ataques de ingeniería social" (p. 92).

El ciclo de vida de una vulnerabilidad describe las etapas por las que pasa desde su creación hasta su resolución. Según Díaz Orueta et al. (2020), este ciclo típicamente incluye las siguientes fases:

- Introducción: La vulnerabilidad se introduce en el sistema, generalmente durante el desarrollo o la implementación.
- Descubrimiento: La vulnerabilidad es identificada, ya sea por investigadores de seguridad, atacantes o por accidente.
- Divulgación: La vulnerabilidad se comunica al fabricante o desarrollador del software/hardware afectado.
- Publicación: Se hace pública la existencia de la vulnerabilidad, a menudo acompañada de detalles técnicos.
- Explotación: Los atacantes pueden comenzar a aprovechar la vulnerabilidad conocida.
- Mitigación: Se desarrollan y aplican parches o soluciones para abordar la vulnerabilidad.
- Resolución: La vulnerabilidad se considera resuelta una vez que se han aplicado las correcciones necesarias.

Gómez Vieites (2020) subraya la importancia de "gestionar adecuadamente el ciclo de vida de las vulnerabilidades para minimizar la ventana de exposición y reducir el riesgo de explotación" (p. 203).

El sistema Common Vulnerabilities and Exposures (CVE) es un estándar internacional para la identificación y catalogación de vulnerabilidades de seguridad conocidas. Costas Santos (2022) explica que "el CVE proporciona un método estandarizado para identificar y referenciar vulnerabilidades específicas, facilitando el intercambio de información sobre seguridad entre diferentes herramientas, bases de datos y servicios" (p. 115).

Características clave del sistema CVE:

- Identificadores únicos: Cada vulnerabilidad recibe un identificador CVE único (por ejemplo, CVE-2021-44228 para la vulnerabilidad Log4Shell).
- Descripción estandarizada: Cada entrada CVE incluye una breve descripción de la vulnerabilidad y sus posibles impactos.

- Referencias: Se proporcionan enlaces a fuentes adicionales de información sobre la vulnerabilidad.
- Puntuación de gravedad: Aunque no es parte directa del CVE, a menudo se asocia con el sistema CVSS (Common Vulnerability Scoring System) para evaluar la gravedad de las vulnerabilidades.

Díaz Orueta et al. (2020) destacan que "el uso del sistema CVE facilita la gestión de vulnerabilidades, permitiendo a las organizaciones priorizar sus esfuerzos de mitigación y mantener sus sistemas actualizados de manera más eficiente" (p. 187).

### 2.2.3. Pentesting

El pentesting, abreviatura de "pruebas de penetración" (penetration testing en inglés), es una práctica de seguridad informática que consiste en evaluar la seguridad de un sistema, red o aplicación mediante la simulación de ataques controlados. Según Gómez Vieites (2020), "el pentesting es un método proactivo para identificar vulnerabilidades y debilidades en los sistemas informáticos antes de que puedan ser explotadas por atacantes malintencionados" (p. 278).

Los principales objetivos del pentesting, como señala Costas Santos (2022), son:

- Identificar vulnerabilidades y debilidades en los sistemas.
- Evaluar la eficacia de las medidas de seguridad existentes.
- Proporcionar una evaluación realista de los riesgos de seguridad.
- Ayudar a priorizar las acciones de mitigación y mejora de la seguridad.
- Cumplir con requisitos regulatorios y de cumplimiento normativo.

La historia del pentesting está estrechamente ligada a la evolución de la seguridad informática. Díaz Orueta et al. (2020) trazan sus orígenes a la década de 1960:

- Años 60-70: Surgimiento de los "phreakers", precursores de los hackers modernos, que exploraban vulnerabilidades en sistemas telefónicos.
- Años 80: Aparición de los primeros hackers informáticos y desarrollo de técnicas de intrusión en sistemas.
- Años 90: Reconocimiento formal del pentesting como práctica de seguridad. Empresas comienzan a contratar "hackers éticos" para evaluar sus sistemas.
- 2000 en adelante: Profesionalización del pentesting, desarrollo de metodologías estandarizadas y herramientas especializadas.

Gómez Vieites (2020) señala que "la evolución del pentesting ha sido impulsada por la creciente sofisticación de las amenazas cibernéticas y la necesidad de adoptar un enfoque proactivo en la seguridad informática" (p. 281).

Costas Santos (2022) describe los tres tipos principales de pentesting:

- Pentesting de caja negra:
  El pentester no tiene conocimiento previo del sistema objetivo. "Este enfoque simula un ataque real de un hacker externo y evalúa la seguridad desde la perspectiva de un atacante sin información interna" (p. 156).
- Pentesting de caja blanca:
  El pentester tiene acceso completo a la información del sistema, incluyendo código fuente, diagramas de red y configuraciones. "Este método permite una evaluación exhaustiva y detallada de la seguridad interna del sistema" (p. 157).
- Pentesting de caja gris:
  Es un enfoque intermedio donde el pentester tiene un conocimiento limitado del sistema. Díaz Orueta et al. (2020) afirman que "el pentesting de caja gris equilibra la simulación de amenazas externas con un conocimiento parcial del sistema, proporcionando una evaluación más completa que la caja negra, pero más realista que la caja blanca" (p. 213).

El proceso de pentesting se divide generalmente en seis fases principales, según Gómez Vieites (2020):

- Planificación:
  Se definen los objetivos, alcance y metodología del pentesting. "Esta fase es crucial para establecer las expectativas y límites de la prueba" (p. 285).
- Reconocimiento:
  Recopilación de información sobre el objetivo. Costas Santos (2022) señala que "esta fase puede incluir técnicas como OSINT (Open Source Intelligence) para obtener datos públicamente disponibles" (p. 160).
- Escaneo:
  Identificación de sistemas activos, puertos abiertos y servicios en ejecución. "El escaneo proporciona una visión detallada de la superficie de ataque potencial" (Díaz Orueta et al., 2020, p. 215).
- Explotación:

Intento de aprovechar las vulnerabilidades identificadas. Gómez Vieites (2020) advierte que "esta fase debe realizarse con extremo cuidado para evitar daños no intencionados en los sistemas objetivo" (p. 287).

- Post-explotación:
  Evaluación del impacto potencial de las vulnerabilidades explotadas. "En esta fase, se determina el alcance del acceso obtenido y se identifican posibles rutas de escalada de privilegios" (Costas Santos, 2022, p. 162).
- Informes:
  Documentación detallada de los hallazgos, vulnerabilidades y recomendaciones. Díaz Orueta et al. (2020) enfatizan que "un informe claro y accionable es esencial para que las organizaciones puedan abordar eficazmente las vulnerabilidades identificadas" (p. 217).

Costas Santos (2022) concluye que "la ejecución metódica de estas fases asegura una evaluación comprehensiva de la seguridad del sistema, proporcionando una base sólida para la mejora continua de las defensas cibernéticas" (p. 163).

### 2.2.4. Metodologías de Pentesting

Las metodologías de pentesting proporcionan marcos estructurados para realizar pruebas de penetración de manera sistemática y exhaustiva. Según Roa Buendía (2023), "estas metodologías aseguran que las pruebas de penetración se realicen de manera consistente, reproducible y efectiva, cubriendo todos los aspectos críticos de la seguridad del sistema" (p. 189). A continuación, se describen tres de las metodologías más reconocidas en el campo del pentesting:

El OSSTMM, desarrollado por el Instituto para la Seguridad y las Metodologías Abiertas (ISECOM), es una metodología completa para probar la seguridad operacional de ubicaciones físicas, interacciones humanas y todas las formas de comunicaciones, ya sean inalámbricas, cableadas o analógicas.

Gavilán Fontecha (2020) destaca que "el OSSTMM se centra en la seguridad operacional real, medible y cuantificable, proporcionando un marco para la verificación científica de la seguridad" (p. 112). Las principales características del OSSTMM incluyen:

- Enfoque holístico: Abarca seguridad física, humana y de tecnología de la información.

- Métricas de seguridad: Introduce el concepto de RAV (Valor de Ataque Real) para cuantificar la seguridad.
- Independencia de las herramientas: Se centra en la metodología más que en herramientas específicas.

El PTES es un estándar desarrollado por un grupo de expertos en seguridad para proporcionar un lenguaje común y un marco de referencia para las pruebas de penetración. López Neira y Ruiz Spohr (2022) señalan que "el PTES ofrece una guía detallada para cada fase del proceso de pentesting, desde la interacción inicial con el cliente hasta la presentación de informes" (p. 245).

Las siete fases principales del PTES son:

- Interacciones previas al compromiso
- Recolección de información
- Modelado de amenazas
- Análisis de vulnerabilidades
- Explotación
- Post-explotación
- Informes

Areitio Bertolín (2021) añade que "el PTES es particularmente valioso por su enfoque en el modelado de amenazas y la fase de post-explotación, aspectos a menudo subestimados en otras metodologías" (p. 301).

La Guía de Pruebas de OWASP se centra específicamente en la seguridad de aplicaciones web. Aceituno Canal (2020) describe esta guía como "un recurso invaluable para los profesionales de seguridad que se especializan en la evaluación de aplicaciones web, proporcionando un enfoque sistemático para identificar vulnerabilidades comunes" (p. 178).

Características clave de la Guía de Pruebas de OWASP:

- Enfoque en aplicaciones web: Aborda vulnerabilidades específicas de entornos web.
- Categorización de pruebas: Organiza las pruebas en categorías como autenticación, autorización, validación de entrada, etc.

- Actualización continua: Se revisa regularmente para incluir nuevas amenazas y técnicas de prueba.

Roa Buendía (2023) destaca que "la Guía de Pruebas de OWASP no solo proporciona una metodología de prueba, sino también información detallada sobre cómo explotar y mitigar vulnerabilidades específicas, lo que la convierte en una herramienta educativa además de una guía práctica" (p. 213).

Estas metodologías proporcionan marcos robustos para la realización de pruebas de penetración, cada una con sus propias fortalezas y enfoques. La elección de la metodología dependerá del contexto específico, los objetivos de la prueba y el tipo de sistema o aplicación que se esté evaluando. Como concluye Gavilán Fontecha (2020), "la combinación de elementos de estas metodologías a menudo resulta en un enfoque más completo y adaptado a las necesidades específicas de cada organización" (p. 120).

### 2.2.5. Herramientas de Pentesting

Las herramientas de pentesting son esenciales para llevar a cabo pruebas de penetración efectivas y eficientes. Según Gavilán Fontecha (2020), "la selección y uso adecuado de herramientas de pentesting puede marcar la diferencia entre una evaluación de seguridad superficial y una exhaustiva" (p. 150). A continuación, se describen algunas de las herramientas más relevantes en diferentes categorías del proceso de pentesting:

Las herramientas de Open Source Intelligence (OSINT) se utilizan para recopilar información públicamente disponible sobre el objetivo. Roa Buendía (2023) señala que "el reconocimiento es la base de cualquier prueba de penetración exitosa, y las herramientas OSINT son fundamentales en esta fase" (p. 230).

Algunas herramientas OSINT populares incluyen:

- Maltego: Permite visualizar relaciones entre piezas de información recopiladas de diversas fuentes en línea.
- Shodan: Un motor de búsqueda para dispositivos conectados a Internet.
- theHarvester: Recopila correos electrónicos, subdominios, hosts, nombres de empleados, puertos abiertos y banners de múltiples fuentes públicas.

López Neira y Ruiz Spohr (2022) destacan que "el uso ético de herramientas OSINT requiere un cuidadoso equilibrio entre la recopilación de información valiosa y el respeto a la privacidad" (p. 278).

Los escáneres de vulnerabilidades son herramientas diseñadas para identificar debilidades potenciales en sistemas y redes.

Nmap (Network Mapper):

Nmap es una herramienta de código abierto utilizada para descubrimiento de redes y auditoría de seguridad. Aceituno Canal (2020) describe Nmap como "una herramienta versátil que puede utilizarse para inventariar redes, monitorear el tiempo de actividad de hosts y detectar servicios en ejecución" (p. 201).

Características clave de Nmap:

- Descubrimiento de hosts activos
- Escaneo de puertos
- Detección de servicios y sistemas operativos
- Scripting para pruebas avanzadas

OpenVAS (Open Vulnerability Assessment System):

OpenVAS es un escáner de vulnerabilidades completo y de código abierto. Areitio Bertolín (2021) señala que "OpenVAS ofrece una solución integral para la detección y gestión de vulnerabilidades, con una base de datos de pruebas en constante actualización" (p. 315).

Características principales de OpenVAS:

- Escaneo de vulnerabilidades en múltiples plataformas
- Interfaz web para gestión y visualización de resultados
- Integración con otras herramientas de seguridad

Las herramientas de explotación se utilizan para aprovechar las vulnerabilidades identificadas y ganar acceso a los sistemas objetivo.

Metasploit Framework:

Es una de las herramientas de explotación más populares y potentes en el campo del pentesting. Gavilán Fontecha (2020) describe Metasploit como "un marco de trabajo

extensible que permite a los profesionales de seguridad desarrollar, probar y ejecutar exploits" (p. 180).

Características clave de Metasploit:

- Amplia biblioteca de exploits y payloads
- Capacidad para desarrollar y probar nuevos exploits
- Integración con otras herramientas de pentesting
- Módulos para post-explotación y pivoteo en redes

Roa Buendía (2023) advierte que "aunque Metasploit es una herramienta poderosa para los pentesters, también es utilizada por actores maliciosos, lo que subraya la importancia de su uso ético y controlado" (p. 245).

Las herramientas de análisis de tráfico permiten examinar en detalle las comunicaciones de red, lo cual es crucial para identificar vulnerabilidades y comprender el comportamiento de los sistemas.

Wireshark:

Wireshark es un analizador de protocolos de red ampliamente utilizado. López Neira y Ruiz Spohr (2022) describen Wireshark como "una herramienta indispensable para cualquier profesional de seguridad, permitiendo una inspección detallada del tráfico de red a nivel de paquete" (p. 300).

Características principales de Wireshark:

- Captura y análisis de tráfico en tiempo real
- Soporte para cientos de protocolos
- Potentes capacidades de filtrado y búsqueda
- Visualización gráfica de flujos de comunicación

Aceituno Canal (2020) añade que "Wireshark no solo es útil para el pentesting, sino también para la resolución de problemas de red y el análisis forense digital" (p. 225).

Es importante destacar que, como señala Areitio Bertolín (2021), "el uso de estas herramientas requiere no solo habilidades técnicas, sino también una sólida comprensión de los aspectos éticos y legales del pentesting" (p. 330). Los profesionales de la seguridad deben asegurarse de tener los permisos adecuados y operar dentro de los límites acordados al utilizar estas potentes herramientas.

# III. METODOLOGÍA

## 3.1. ENFOQUE METODOLÓGICO

### 3.1.1. Enfoque

Cualitativo y cuantitativo.

El presente estudio adopta un enfoque mixto: cualitativo y cuantitativo, debido a la naturaleza técnica y contextual de la evaluación de seguridad informática. Desde el enfoque cuantitativo, se recopilaron datos objetivos relacionados con servicios expuestos, versiones de software, número de vulnerabilidades detectadas, niveles de criticidad (según estándares como CVSS, CVE y CWE) y estadísticas generadas por herramientas de análisis como Nmap, WPScan, Nikto, entre otras.

Pero también el enfoque cualitativo permitió analizar el comportamiento de los sistemas evaluados frente a escenarios simulados de ataque, interpretar las configuraciones encontradas y observar el entorno institucional en el que operan. A través de entrevistas informales con personal del área TIC, también se identificaron criterios y limitaciones en la gestión actual de la seguridad.

La combinación de ambos enfoques permitió no solo cuantificar los riesgos técnicos, sino también comprender el contexto operativo en el que surgen y persisten dichas vulnerabilidades. Esta perspectiva integral resultó clave para proponer soluciones viables y adaptadas al entorno institucional.

### 3.1.2. Tipo de Investigación

Investigación experimental.

Se realizaron pruebas de intrusión controladas en una red específica. Al manipular y controlar las variables, se evalúo la efectividad de los controles de seguridad implementados y se determinó cómo responden ante intentos de intrusión.

La investigación permitió obtener resultados empíricos y cuantificables que respalden conclusiones y recomendaciones. Además, proporcionó la oportunidad de analizar

el impacto de las técnicas de pentesting en la identificación y explotación de vulnerabilidades.

Investigación de campo

Al ejecutar las pruebas en un entorno real y no simulado, este tipo de investigación permitió obtener resultados empíricos y cuantificables que respaldan las conclusiones formuladas, y valorar el impacto de las técnicas de pentesting en entornos productivos bajo condiciones reales de operación.

### 3.2. IDEA A DEFENDER

La aplicación de la metodología PTES permite identificar de manera estructurada y efectiva las vulnerabilidades presentes en una red universitaria, facilitando así el diseño de medidas de mitigación que fortalezcan la postura de seguridad institucional.

### 3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES

Definición de las variables

Técnicas de Pentesting: Métodos utilizados para probar la seguridad de un sistema.

Evaluación de Vulnerabilidades: Proceso mediante el cual se identifican, analizan y clasifican fallos de seguridad en los sistemas.

### 3.3.1. Operacionalización de las variable

**Tabla 1.** Variable Independiente

| | Variable | Definición | Dimensión | Indicador | Técnica | Instrumento |
|---|---|---|---|---|---|---|
| Variable independiente | Técnicas de Pentesting | Conjunto de métodos y procedimientos sistemáticos utilizados para evaluar la seguridad de sistemas informáticos mediante pruebas de penetración controladas. Se aplicó la metodología PTES como marco estructural. | Percepción de expertos | Nivel de efectividad percibida<br>Complejidad de Implementación | Entrevista semiestructurada | Guía de entrevista |
| | | | Reconocimiento | Número de hosts identificados<br>Cantidad de servicios detectados<br>Porcentaje de puertos analizados | Escaneo de red<br>Enumeración de servicios | Nmap<br>Logs de reconocimiento |
| | | | Análisis | Número de pruebas realizadas<br>Tiempo de ejecución<br>Cobertura de análisis | Análisis automatizado<br>Verificación manual<br>Pruebas de configuración | Scanners de vulnerabilidades |
| | | | Documentación | Cantidad de hallazgos<br>Precisión de los reportes | Registro sistemático<br>Documentación técnica | Plantillas de reporte<br>Herramientas de documentación |

**Tabla 2.** Variable Dependiente

| | Variable | Definición | Dimensión | Indicador | Técnica | Instrumento |
|---|---|---|---|---|---|---|
| Variable dependiente | Evaluación de Vulnerabilidades | Proceso técnico para identificar, clasificar y analizar fallos de seguridad en sistemas informáticos, mediante el uso de pruebas controladas de intrusión. | Identificación | -Número de vulnerabilidades detectadas<br>-Tasa de falsos positivos | Entrevista semiestructurada | Guía de entrevista |
| | | | Clasificación | -Nivel de severidad(CVSS)<br>-Tipo de vulnerabilidad<br>-Impacto potencial | Escaneo de red<br>Enumeración de servicios<br>Análisis automatizado | Nmap<br>Logs de reconocimiento |
| | | | Mitigación | -Efectividad de controles<br>-Costo de Implementación | Verificación manual<br>Pruebas de configuración | Scanners de vulnerabilidades |

## 3.4. MÉTODOS UTILIZADOS

**Tabla 3.** Comparación de Metodologías de Pentesting

| Criterio | PTES | OWASP | NIST SP 800-115 |
|---|---|---|---|
| Enfoque principal | Estandariza todas las fases del pentesting, desde la planificación hasta el reporte final, con énfasis en procesos técnicos y estratégicos. | Evaluación de aplicaciones web, centrada en pruebas para vulnerabilidades en software y componentes web. | Guía técnica para pruebas de seguridad, con enfoque general para redes, sistemas y aplicaciones. |
| Cobertura de fases | Completa: planificación, reconocimiento, modelado de amenazas, análisis, explotación, post-explotación y reporte. | Parcial: principalmente enfoque en pruebas manuales (black-box) para aplicaciones. | Parcial: orientada a entornos corporativos, pero con menor profundidad en explotación o post análisis. |
| Nivel de detalle técnico | Alto: establece herramientas, técnicas y procedimientos específicos por fase. | Medio: se centra en escenarios de ataque comunes, pero no detalla cómo explotarlos en entornos reales. | Medio: estructura general sin bajar a nivel de procedimientos detallados. |
| Tipo de objetivos que abarca | Infraestructura de red, servicios web, dispositivos, usuarios, entornos híbridos. | Aplicaciones web principalmente. | Redes, dispositivos y algunos aspectos de software. |
| Facilidad de adopción | Requiere conocimiento técnico y planificación, pero es modular y escalable según recursos. | Más sencilla para entornos web, pero limitada fuera de ellos. | Adecuada para entornos organizativos amplios con procedimientos formales. |
| Aplicabilidad en universidades | Alta: se adapta bien a infraestructuras mixtas con limitaciones de personal o presupuesto. | Media: útil para revisar portales institucionales, pero no evalúa redes ni servicios críticos. | Media: sugiere evaluaciones, pero no detalla cómo ejecutarlas con herramientas accesibles. |
| Beneficio en esta investigación | Permite aplicar un enfoque estructurado, ético y completo, que cubre desde la recolección hasta la presentación del informe técnico. | Sirve como complemento para validar aplicaciones web (como WordPress), pero no cubre el resto del sistema. | Útil como marco organizativo, pero insuficiente para pruebas prácticas detalladas. |
| Motivo de elección | Fue seleccionada por su capacidad de ajustarse a entornos reales, su claridad metodológica, su enfoque profesional y su utilidad académica. | Se utilizó como referencia para pruebas web puntuales. | Referenciada como estándar institucional, pero no adoptada por su poca aplicabilidad directa. |

La metodología PTES fue seleccionada para esta investigación por su capacidad de estructurar de forma completa y práctica todas las fases de un pentest profesional. A diferencia de OWASP, que se centra en aplicaciones web, o de NIST, que es más

normativo y menos operativo, PTES proporciona una guía concreta y flexible, útil tanto en entornos corporativos como en escenarios educativos con recursos limitados.

Su enfoque paso a paso, desde la planificación hasta la post-explotación y el reporte, permitió una ejecución ordenada, con resultados tangibles y aplicables. Además, su compatibilidad con herramientas de código abierto facilita su adopción en contextos donde no se dispone de soluciones comerciales. Estas características convierten a PTES en una alternativa ideal para trabajos de investigación aplicada como el presente estudio.

## 3.5. ANÁLISIS ESTADÍSTICO

**Entrevista**

**1. ¿Cómo describiría su experiencia personal en seguridad informática?**

**Respuesta:**
Mi formación inicial incluyó aspectos básicos de seguridad, como la implementación de firewalls, IPS e IDS. Sin embargo, no me he especializado en seguridad informática como tal. Mi experiencia ha estado más enfocada en redes e infraestructura. Conozco los conceptos generales del pentesting y su importancia, pero no he tenido la oportunidad de aplicar herramientas específicas ni liderar proyectos de pruebas de penetración.

**Análisis:**
Esta respuesta muestra una base conceptual en seguridad, pero sin aplicación práctica reciente. Refleja una brecha común en roles de infraestructura donde la seguridad se considera un complemento y no un área de especialización. Puede representar una oportunidad para fortalecer el área con personal o proyectos que aborden directamente esta necesidad.

**2. ¿Considera que la universidad ha implementado estrategias concretas de pentesting o análisis de vulnerabilidades?**

**Respuesta:**
No de forma institucional. Usualmente, este tipo de actividades se realizan en el contexto de proyectos de titulación de estudiantes de carreras afines. Desde el área de TIC, damos soporte a esos proyectos, pero no hemos ejecutado pruebas de penetración de manera formal ni periódica desde el departamento.

**Análisis:**

Aquí se evidencia una ausencia de políticas o estrategias estructuradas en torno al pentesting. Delegar estas tareas exclusivamente a estudiantes refleja una falta de planificación organizacional en seguridad. Sería clave desarrollar protocolos institucionales que incluyan estas prácticas como parte del ciclo de seguridad de TI.

**3. ¿Qué limitaciones enfrentan al intentar fortalecer la seguridad de los sistemas?**

**Respuesta:**

Una de las principales limitaciones es la falta de personal especializado exclusivamente en seguridad informática. Otro factor es el presupuesto, ya que muchas herramientas avanzadas requieren licencias costosas. Además, no contamos con entornos de pruebas aislados, lo que limita realizar simulaciones sin afectar la operación.

**Análisis:**

Este tipo de limitaciones son comunes en organizaciones educativas. Subraya la necesidad de apostar por soluciones de código abierto, formación continua y, si es posible, colaboraciones con instituciones externas que permitan la transferencia de conocimientos y herramientas.

**4. ¿Qué importancia cree que tiene el pentesting dentro del contexto universitario?**

**Respuesta:**

Permite a los estudiantes aplicar sus conocimientos en entornos reales y detectar vulnerabilidades que podrían pasar desapercibidas. Sin embargo, desde el área administrativa aún no le hemos dado el peso operativo que merece.

**Análisis:**

La visión positiva hacia el pentesting académico abre la puerta para futuras colaboraciones entre el área técnica y la comunidad estudiantil. Formalizar estos ejercicios dentro de un marco institucional, incluso como parte de auditorías internas, fortalecería el entorno de seguridad de la universidad.

**5. ¿Qué medidas básicas de seguridad aplica actualmente el departamento de TIC?**

**Respuesta:**

Nos enfocamos en mantener los sistemas actualizados, realizar respaldos periódicos y configurar adecuadamente los firewalls. También vigilamos el tráfico de red y

limitamos el acceso a ciertos servicios, aunque no utilizamos herramientas de monitoreo avanzadas.

**Análisis:**

Estas medidas representan una base sólida, pero insuficiente ante amenazas más sofisticadas. Sería beneficioso avanzar hacia soluciones más completas, como la segmentación de red, análisis de logs, autenticación multifactor y detección temprana de anomalías.

**6. ¿Considera que sería útil capacitar al personal del área en herramientas de pentesting?**

**Respuesta:**

Sí, definitivamente. Aunque tenemos una noción general, hace falta una formación más específica y práctica. Esto permitiría responder mejor ante incidentes, entender los reportes de vulnerabilidades y mejorar la prevención.

**Análisis:**

Reconocer la necesidad de capacitación es el primer paso para fortalecer el equipo. Invertir en formación técnica puede ser más sostenible a largo plazo que depender exclusivamente de proyectos estudiantiles, y permite que la seguridad se integre de forma transversal en la gestión tecnológica.

# IV. RESULTADOS Y DISCUSIÓN

## 4.1. RESULTADOS

## 4.2. PROPUESTA

La seguridad informática es un pilar fundamental en la protección de datos y sistemas dentro de las organizaciones. En el caso de las instituciones de educación superior, como la Universidad Politécnica Estatal del Carchi (UPEC), la integridad de la red y la información es crucial para garantizar la continuidad de sus actividades académicas y administrativas.

El proyecto tiene como objetivo aplicar técnicas de pentesting para evaluar vulnerabilidades dentro de la infraestructura tecnológica de la universidad, permitiendo identificar riesgos potenciales y proponer estrategias de mitigación.

La ejecución del proyecto comprende diversas fases que incluyen la investigación teórica, el análisis de la red, la ejecución de pruebas de penetración y la generación de informes de vulnerabilidad. Durante el proceso se abordarán aspectos clave como:

- Identificación de activos críticos dentro de la infraestructura de la universidad.
- Evaluación de configuraciones y seguridad.
- Simulación de ataques controlados para medir el nivel de exposición a amenazas.
- Recomendaciones para fortalecer la seguridad de la red.

Este enfoque permitirá a la Universidad Politécnica Estatal del Carchi mejorar sus mecanismos de defensa ante posibles ciberataques y fortalecer la seguridad de sus sistemas.

### 4.2.1. Estudio de Factibilidad

**Información sobre la Institución**

La Universidad Politécnica Estatal del Carchi (UPEC) es una institución de educación superior. Su enfoque académico está orientado a la formación de profesionales en diversas áreas del conocimiento, con énfasis en tecnología e innovación.

**Ubicación**

La UPEC se encuentra en la ciudad de Tulcán, capital de la provincia de Carchi, en la región norte de Ecuador, cerca de la frontera con Colombia. Su ubicación estratégica la convierte en un punto de referencia académico en la zona.

**Área de Aplicación**

Este estudio se enfocará en evaluar la seguridad de la infraestructura de red y los sistemas informáticos de la universidad. Las pruebas se centrarán en identificar vulnerabilidades en:

Redes internas y externas de la institución.

Servidores utilizados para la gestión académica y administrativa.

Aplicaciones web internas de la UPEC.

**¿Qué se va a hacer?**

El proyecto consistió en la ejecución de un pentesting controlado en la red universitaria, siguiendo protocolos de seguridad que permitió evaluar su resistencia ante posibles ataques. Se desarrollará un informe con los hallazgos y se presentarán recomendaciones para mejorar la ciberseguridad institucional.

**Misión y Visión de la UPEC**

Misión: "Formar profesionales competentes, con principios éticos y valores, que contribuyan al desarrollo sostenible de la sociedad mediante la generación y aplicación del conocimiento."

Visión: "Ser una institución de educación superior reconocida por su excelencia académica, innovación tecnológica e impacto en el desarrollo regional y nacional."

**Justificación Técnica**

El proyecto es técnicamente viable, considerando:

Infraestructura Disponible

Para llevar a cabo las pruebas de seguridad, se utilizarán los siguientes recursos:

Equipos de cómputo y servidores universitarios: La UPEC proporciona servidores y acceso a la red institucional, lo que permite realizar evaluaciones de vulnerabilidades en un entorno real.

Herramientas de código abierto: Se emplearán software de pentesting como Nmap, Metasploit, OpenVAS, Burp Suite Community Edition, Wireshark, entre otros. Esto garantiza que las pruebas sean efectivas sin incurrir en costos adicionales de licencias.

**Compatibilidad con los Sistemas de la Universidad**

El pentesting se ejecutará sin afectar la operatividad de la institución, respetando las restricciones y políticas de acceso a información sensible.

**Segmentación de pruebas:** Se realizarán auditorías en redes específicas, sin comprometer la estabilidad de la infraestructura general.

**Experiencia del Investigador**

El autor cuenta con formación en Ciencias de la Computación, lo cual respalda la capacidad técnica para realizar las pruebas de penetración. El proyecto se fundamenta en metodologías reconocidas como:

**PTES** (Estándar para pruebas de penetración)

**OWASP Testing Guide** (para análisis de aplicaciones web).

**NIST SP 800-115** (para metodologías de pruebas de seguridad en sistemas).

**MITRE ATT&CK** (para identificar vectores de ataque utilizados por ciberdelincuentes).

**Alcance Técnico del Pentesting**

El análisis de seguridad incluirá los siguientes aspectos técnicos:

Reconocimiento y análisis de infraestructura: Identificación de servicios expuestos y mapeo de la red.

Escaneo de vulnerabilidades: Evaluación de fallos en servidores, aplicaciones web y bases de datos.

Pruebas de explotación controladas: Simulación de ataques éticos para comprobar el impacto de las vulnerabilidades encontradas.

Evaluación de políticas de seguridad: Análisis del uso de contraseñas, segmentación de red y accesos.

**Justificación Económica**

La inversión es razonable en comparación con los beneficios:

Costos mínimos: Solo implica gastos de transporte, electricidad e internet.

Ahorro en licencias: Uso de herramientas gratuitas en lugar de software comercial.

Prevención de pérdidas: La identificación temprana de vulnerabilidades evita costos por ataques cibernéticos en el futuro.

**Tabla 4.** Costos Operativos (4meses)

| Concepto | Mensual (USD) | Total 4 meses (USD) |
|---|---|---|
| Transporte | $40 | $160 |
| Electricidad | $8 | $32 |
| Internet | $21 | $84 |
| Total, Operativo | $69 | $276 |

**Costo Total del Proyecto (4 meses)**

Valor de tu tiempo: $4,000 USD.

Costos operativos: $276 USD.

Costo total estimado: $4,276 USD.

**Justificación Legal y Ética**

Autorización institucional: Se requirió un permiso formal de la UPEC para realizar las pruebas.

Respeto a la privacidad y protección de datos: No se extrajo información personal ni se modificó datos críticos.

Cumplimiento normativo: Se siguieron estándares como la Ley Orgánica de Protección de Datos Personales de Ecuador y normativas internacionales de ciberseguridad.

**Beneficios del Proyecto**

Mejora de la seguridad informática: La universidad conocerá sus puntos débiles y podrá reforzarlos.

Fortalecimiento académico: El proyecto puede servir como referencia para futuras investigaciones en ciberseguridad.

Reducción del riesgo de ataques: Al corregir vulnerabilidades antes de que sean explotadas.

Optimización de políticas de seguridad: La UPEC podrá actualizar sus protocolos de protección de datos.

**Posibles Riesgos y Limitaciones**

Riesgos:

Interrupción del servicio: Las pruebas mal ejecutadas pueden afectar la operatividad de los sistemas.

Errores en la detección de vulnerabilidades: Algunas fallas podrían no identificarse si las pruebas no se realizan correctamente.

Limitaciones:

Acceso restringido a ciertos sistemas: La universidad podría limitar las pruebas en servidores críticos.

### 4.2.2. Resultados por fase de PTES

La fase de Interacciones previas al compromiso constituye el punto de partida en la metodología PTES. En esta etapa se definieron de forma clara los objetivos de la evaluación, el alcance permitido por la institución, los límites técnicos del entorno de pruebas, así como los responsables del monitoreo y acompañamiento durante el proceso. Además, se establecieron los acuerdos legales y éticos necesarios, incluyendo una carta de autorización formal por parte de la Universidad Politécnica Estatal del Carchi (UPEC). Esta planificación inicial garantiza que el pentesting se realice de forma controlada, segura y conforme con los lineamientos institucionales y normativos vigentes.

### 4.2.3. Definición del Alcance

Para delimitar la evaluación de seguridad, se han identificado los siguientes componentes como objetivos del pentesting:

Sistemas críticos de información: Plataformas de gestión académica y bases de datos que contienen información sensible de estudiantes y docentes.

Infraestructura de red: Segmentos de red internos y externos, dispositivos de comunicación y servidores conectados a internet.

Endpoints y dispositivos físicos: Computadoras de laboratorios y servidores internos utilizados en procesos administrativos y académicos.

Servicios en la nube: Aplicaciones alojadas en plataformas externas utilizadas por la universidad para gestión académica y administrativa.

Además, se han definido los límites del pentesting para evitar interrupciones en los servicios críticos y garantizar el cumplimiento de las normativas institucionales.

### 4.2.4. Requerimientos Técnicos

Para llevar a cabo las pruebas de penetración, se han identificado las siguientes herramientas y configuraciones:

**Herramientas Para Utilizar**

Las herramientas seleccionadas para cada fase del pentesting son:

Escaneo y enumeración:

Nmap (detección de servicios y puertos abiertos).

Netdiscover (identificación de dispositivos en la red).

**Tabla 5.** Comparación de herramientas de escaneo

| Criterio | Nmap (Elegida) | Zenmap | Masscan |
|---|---|---|---|
| Tipo de herramienta | Escáner avanzado de puertos y servicios. | GUI de Nmap | Escaneo ultrarrápido |
| Nivel de detalle | Muy alto (servicios, versiones, scripts NSE). | Medio – Alto depende de Nmap | Bajo (solo puertos) |
| Precisión | Alta | Alta | Media |
| Funciones adicionales | Detección SO, NSE scripting, escaneo evasivo. | Misma que Nmap | Enorme velocidad |
| Curva de aprendizaje | Media | Baja | Media |
| Casos de uso ideales | Pentesting profesional y auditorias completas. | Uso educativo | Escaneos masivos |
| Motivo de elección | Ofrece información profunda, flexible y exacta; es estándar en pentesting | No añade valor técnico real | Solo velocidad, no sirve para análisis detallado |

**Tabla 6.** Comparación de herramientas ARP

| Criterio | Netdiscover (Elegida) | ARP-Scan | Fing |
|---|---|---|---|
| Método | ARP scanning | ARP scanning | Multiplataforma |
| Velocidad | Alta | Alta | Media |
| Nivel de detalle | IP, MAC y vendor | Similar, pero más técnico | Básico |
| Requisitos | Bajo | Medio | App externa |
| Motivo de elección | Ligero, rápido y suficiente para redes locales | Requiere más configuración | No es apropiado para pentesting profesional |

Análisis de vulnerabilidades:

OpenVAS (detección de vulnerabilidades en servidores y aplicaciones).

Nikto (análisis de seguridad en servidores web).

**Tabla 7.** Comparación de escáneres de vulnerabilidades

| Criterio | OpenVAS (Elegida) | Nessus | QualysGuard |
|---|---|---|---|
| Licencia | Libre / open source | Comercial | Comercial |
| Base de datos | Amplia, actualizada | Muy Amplia | Muy Amplia |
| Nivel de detalle | Alto | Muy Alto | Muy Alto |
| Costo | Gratis | Alto | Muy Alto |
| Requisitos | Medio | Bajo | Nube |
| Motivo de elección | Permite evaluación completa sin costos | Requiere licencias | Servicio empresarial fuera del alcance. |

**Tabla 8.** Comparación de herramientas de análisis HTTP

| Criterio | Nikto (Elegida) | Wappalyzer | Skipfish |
|---|---|---|---|
| Tipo | Escáner de vulnerabilidades web | Fingerprinting | Fuzzer web |
| Nivel de profundidad | Alto (cabeceras, fallos comunes, CVEs) | Bajo (solo tecnologías) | Medio |
| Errores típicos detectados | Malas configuraciones, versiones, archivos expuestos | Ninguno | Limitado |
| Motivo de elección | Encuentra vulnerabilidades reales y configuraciones inseguras | Solo reconocimiento | Muy ruidoso y menos preciso |

Explotación de vulnerabilidades:

Metasploit Framework (automatización de exploits).

**Tabla 9.** Comparación de frameworks de explotación

| Criterio | Metasploit (Elegida) | Core Impact | Immunity Canvas |
|---|---|---|---|
| Licencia | Libre | Comercial | Comercial |
| Módulos disponibles | Extensa librería actualizada | Muy Alta | Alta |
| Popularidad | Estándar mundial | Empresarial | Empresarial |
| Costos | Gratis | Muy Alto | Alto |
| Motivo de elección | La herramienta profesional más completa disponible sin costo | Licencias prohibitivas | No accesible para universidades |

Intercepción y manipulación de tráfico:

Burp Suite (análisis de tráfico web y ataques a aplicaciones).

**Tabla 10.** Comparación de herramientas proxy

| Criterio | Burp Suite (Elegida) | OWASP ZAP | Fiddler |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Tipo | Proxy interceptador para pentesting web | Proxy y escáner web | Analizador HTTP |
| Nivel profesional | Muy Alto | Alto | Medio |
| Funcionalidades | Intruder, repeater, decoder, sequencer | Escáner automático | Solo debugging |
| Motivo de elección | Estándar profesional y más completa incluse en versión gratuita | Muy buena, pero menos precisa en ataque manuales | No apta para pentesting serio |

Ataques a credenciales y autenticación:

Hydra (ataques de fuerza bruta a contraseñas).

John the Ripper (crackeo de hashes de contraseña).

**Tabla 11.** Comparación de herramientas de cracking

| Criterio | Hydra / Jhon | Hashcat | Medusa |
|---|---|---|---|
| Tipo | Fuerza bruta ( Hydra) / Cracking hashes (Jhon) | GPU-cracking | Fuerza bruta |
| Alcance | Amplio | Muy Alto | Medio |
| Requisitos | Bajos | GPU | Bajos |
| Motivo de elección | Suficientes para pruebas controladas y acorde al alcance académico | Necesita hardware adicional | Menos potente que Hydra |

#### 4.2.5.2 Infraestructura y Equipos Necesarios

Para la ejecución de las pruebas se hizo uso de la siguiente infraestructura:

Computadora con capacidad de procesamiento suficiente para ejecutar herramientas de pentesting.

- Procesador - Intel i5 – 10300h 4 núcleos 8 hilos.
- GPU – Nvidia GTX 1650.
- RAM – 8 Gigabytes.

Acceso a la red objetivo con los permisos correspondientes.

- Conexión a internet vía wifi de 100 megabits

Entornos de prueba en máquinas virtuales para evitar afectar sistemas en producción.

#### 4.2.5.3 Planificación del Pentesting

Con base en los requerimientos definidos, se ha estructurado un plan de trabajo que considera los siguientes aspectos:

Asignación de responsabilidades, identificando el rol de cada integrante del equipo de evaluación.

Plan de mitigación de riesgos, estableciendo medidas para prevenir impactos negativos en la operación de los sistemas evaluados.

Criterios de éxito, determinando qué resultados se esperan y cómo se evaluará la efectividad del proceso.

### 4.3. FASE1: INTERACCIONES PREVIAS AL COMPROMISO.

Fue establecido el contacto con los responsables del área de Tecnologías de la Información de la UPEC, quienes autorizaron formalmente la ejecución de pruebas de penetración. Se delimitó el alcance del pentesting a redes internas, servicios web institucionales y servidores académicos. Se elaboró un documento de autorización y se definieron los términos para garantizar que las pruebas no afectaran la continuidad operativa.

### 4.4. FASE 2: RECOLECCIÓN DE INFORMACIÓN.

Mediante herramientas OSINT (theHarvester, DNSDumpster, WHOIS), fueron recolectados los siguientes datos:

Subdominios y servidores web visibles.

Direcciones IP públicas asociadas a la institución.

Información sobre certificados SSL y encabezados HTTP.

Tecnologías utilizadas en aplicaciones web (detectadas con WhatWeb).


**Figura 1.** Herramienta theHarvester
**Fuente:** Mensaje de presentación de la herramienta, de Ignacio Pérez, 2024


**Figura 2.** Herramienta DNSDumpster
**Fuente:** DNSDumpster, de VulneraLabs, 2022

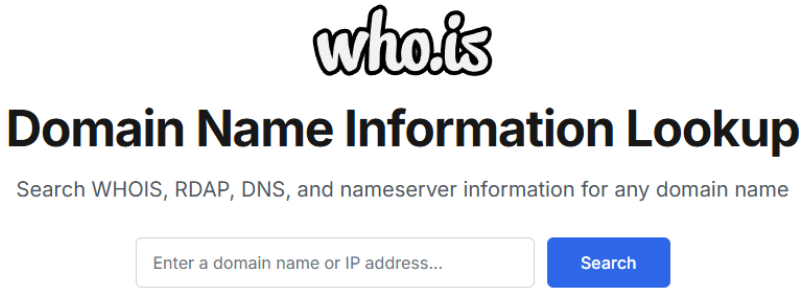



**Figura 3.** Herramienta WHOIS
**Fuente:** Whois, de Who.is, 2025

### 4.4.1. Reconocimiento Pasivo

Es la primera fase del proceso de pentesting y consiste en recopilar información sobre el objetivo sin interactuar directamente con sus sistemas. Su propósito es obtener datos que permitan comprender la infraestructura y detectar posibles vectores de ataque sin alertar a los sistemas de seguridad.

WhatWeb is a next generation web scanner.

WhatWeb recognises web technologies including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices.

WhatWeb has over 1800 plugins, each to recognise something different. WhatWeb also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more.

Enter a domain to analyze:
upec.edu.ec Go

Download

Wiki

```
http://upec.edu.ec [301 Moved Permanently] Country[ECUADOR][EC],
 IP[190.15.133.60],
 RedirectLocation[https://upec.edu.ec:443/],
 Title[Object moved permanently]
https://upec.edu.ec/ [200 OK] Cookies[PHPSESSID,cookiesession1],
 Country[ECUADOR][EC],
 Email[info@upec.edu.ec],
 Frame, HTML5, HTTPServer[nginx],
 HttpOnly[cookiesession1],
 IP[190.15.133.60],
 JQuery[3.7.1],
 MetaGenerator[Elementor 3.27.6; features: additional_custom_breakpoints; settings: css_print_me
 Script[text/javascript,text/template],
 Title[UPEC],
 UncommonHeaders[link],
 Vimeo, WordPress[6.7.2],
 nginx
```

**Figura 4.** Reporte de WhatWeb

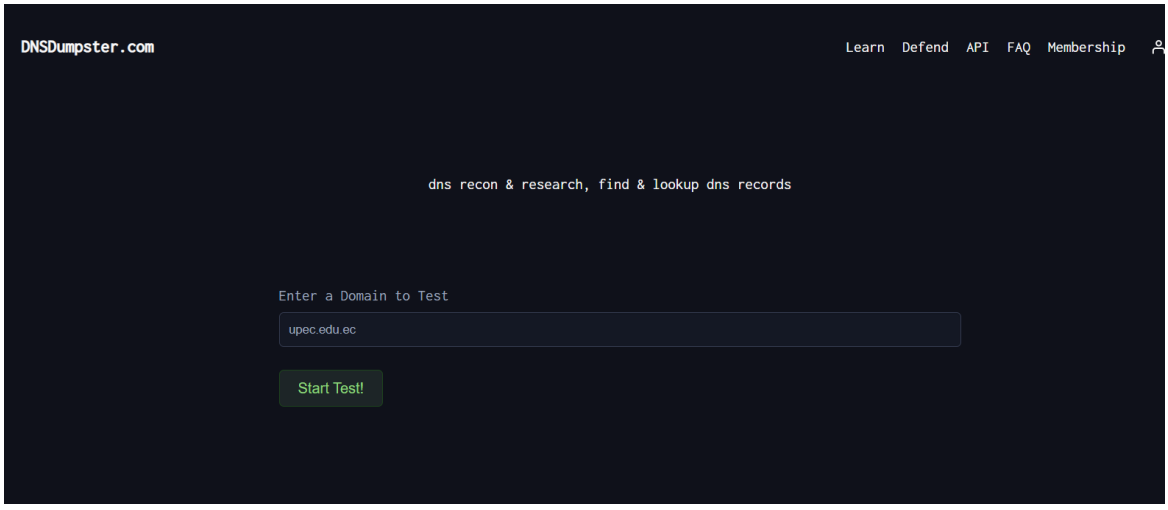



**Figura 5.** Reconocimiento en DNSDumspter


**Figura 6.** Mapa de Subdominios de DNSDumspter

## 4.5. FASE 3: MODELADO DE AMENAZAS

Durante la fase de Modelado de amenazas se hizo un análisis estructurado de los posibles vectores de ataque y escenarios de explotación que podrían comprometer los activos críticos de la Universidad Politécnica Estatal del Carchi (UPEC). Se identificaron los sistemas más sensibles, como servidores académicos, aplicaciones web internas y servicios administrativos. Con base en la información recolectada previamente, se construyeron perfiles de atacantes (actores de amenaza) y se categorizaron sus posibles acciones utilizando marcos como STRIDE y MITRE ATT&CK. Este modelado permitió anticipar técnicas potenciales de acceso, escalamiento y persistencia, y sirvió como guía estratégica para orientar el análisis de vulnerabilidades y priorizar los objetivos de explotación durante las fases siguientes.

En esta fase se construyó un modelo de amenazas a partir de la información recolectada durante el reconocimiento pasivo y activo, utilizando herramientas como dnsdumpster y WhatWeb. El análisis permitió identificar diversos subdominios asociados al dominio principal upec.edu.ec, así como sus respectivas direcciones IP

públicas, rangos de red, servidores DNS, MX y proxys. Esta información fue clave para visualizar la superficie de ataque expuesta por la infraestructura de la Universidad Politécnica Estatal del Carchi (UPEC).

Entre los activos críticos detectados se encuentran portales institucionales como midispace.upec.edu.ec, biblioteca.upec.edu.ec, virtualgrado.upec.edu.ec, transparencia.upec.edu.ec, entre otros. Algunos de estos están desplegados sobre tecnologías web populares como WordPress 6.7.2, lo cual representa un posible vector de ataque conocido por sus vulnerabilidades históricas si no está debidamente actualizado. Además, se identificaron encabezados HTTP que podrían revelar información sensible como X-Powered-By, Server: nginx, y cookies de sesión activas, las cuales podrían estar expuestas a ataques de secuestro de sesión si no se aplican correctamente atributos de seguridad como HttpOnly y Secure.

A través del uso de WhatWeb, se observó también el uso del plugin Elementor 3.27.6 en el CMS, lo cual representa un punto de interés para un posible atacante, dado el historial de vulnerabilidades encontradas en versiones anteriores de este constructor visual. Asimismo, se identificó el uso de tecnologías JavaScript como jQuery 3.7.1, lo cual podría abrir la puerta a ataques del tipo XSS (Cross-Site Scripting) si no se valida adecuadamente el input del usuario.

El modelado de amenazas contempló actores como atacantes externos no autenticados, atacantes internos (usuarios maliciosos con acceso limitado a servicios institucionales) y atacantes persistentes que podrían utilizar técnicas avanzadas para evadir la detección. Se consideraron escenarios de ataque como explotación de vulnerabilidades conocidas en los CMS, escaneo de puertos expuestos y ataques a través de DNS mal configurados.

Este análisis fue fundamental para priorizar los objetivos y diseñar las rutas de ataque más probables a ser comprobadas en las siguientes fases del pentesting, enfocándose en aquellas superficies expuestas con mayor valor para un atacante y mayor riesgo para la organización.

### 4.5.1. Aplicación del modelo STRIDE

El modelo STRIDE clasifica las amenazas en seis categorías. A continuación, se detalla su aplicación sobre los sistemas y servicios detectados en los dominios y subdominios de la UPEC:

**Tabla 12.** Modelo STRIDE aplicado a la universidad

| Categoría STRIDE | Descripción | Ejemplo en UPEC |
|---|---|---|
| Spoofing | Suplantación de identidad | Posibles ataques por fuerza bruta a paneles de login sin autenticación multifactor. |
| Tampering | Alteración de datos | Subida de archivos maliciosos o modificación de contenido en WordPress mal configurado. |
| Repudiation | Negación de acciones | Ausencia de logs de acceso en algunos subdominios puede permitir que acciones maliciosas pasen desapercibidas. |
| Information Disclosure | Fuga de Información | Exposición de versiones, directorios listados y cabeceras HTTP informativas en nginx. |
| Denial of Service | Interrupción de servicios | Subdominios sin WAF podrían ser vulnerables a ataques de denegación de servicio. |
| Elevation of Privilege | Escalada de privilegios | Plugins WordPress vulnerables podrían permitir escalar privilegios hasta administradores. |

### 4.5.2. Aplicación de MITRE ATT&CK

El marco MITRE ATT&CK permitió mapear las posibles técnicas que un atacante podría usar, basadas en las debilidades observadas en los sistemas. Algunas técnicas relevantes al contexto fueron:

**Tabla 13.** Modelo STRIDE aplicado a la universidad

| Técnica ATT&CK ID | Técnica | Descripción aplicada a UPEC |
|---|---|---|
| T1190 | Exploit Public-Facing Application | Servicios WordPress y nginx expuestos podrían ser atacados con exploits conocidos. |
| T1046 | Network Service Scanning | La recolección de información mediante escaneo de puertos es viable en rangos IP sin restricciones. |
| T1110.001 | Brute Force: Password Guessing | Paneles de login WordPress sin CAPTCHA o rate-limiting permiten ataques por diccionario. |
| T1552.001 | Unsecured Credentials: Files | Si existen archivos como .env o .git, podrían contener contraseñas expuestas. |
| T1566.002 | Phishing via Service | Un atacante podría usar subdominios con apariencia legítima para phishing dirigido a estudiantes o docentes. |
| T1203 | Exploitation for Client Execution | Exploits en plugins o plantillas maliciosas permitirían ejecución de código al visitar páginas comprometidas. |

Este modelado permitió priorizar las amenazas más relevantes, sirviendo como base para el análisis de vulnerabilidades y la planeación de la fase de explotación.

## 4.6. FASE 4: ANÁLISIS DE VULNERABILIDADES

### 4.6.1. Reconocimiento pasivo.

Para hacer el reconocimiento pasivo se empleó herramientas como whois, dnsrecon, amass y theHarvester, estas permiten recolectar información pública del objetivo. Luego, para el reconocimiento activo se utilizó nmap, con los parámetros -sV, -T4 y -Pn, a fin de detectar versiones de servicios expuestos sin hacer ping, optimizando el escaneo. En el análisis web, se aplicó nikto para identificar vulnerabilidades conocidas y whatweb para identificar tecnologías implementadas en el servidor web.

Recolección de registros DNS con DNSRecon

Se hizo uso de la herramienta dnsrecon para obtener información del dominio upec.edu.ec. La herramienta permite hacer una enumeración general de registros asociados al dominio

Comando utilizado:

dnsrecon -d upec.edu.ec

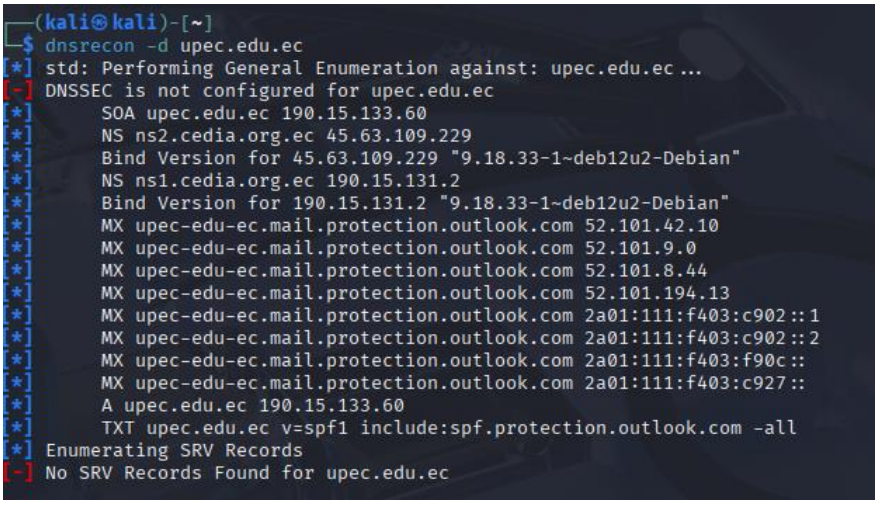



**Figura 7.** Reporte de DNSRecon del portal principal UPEC

Resultados obtenidos

DNSSEC no configurado.

El dominio no cuenta con mecanismos de validación de integridad DNS, lo cual puede representar una oportunidad para ciertos tipos de ataques como cache poisoning (envenenamiento de cache).

SOA (Start of Authority).

IP: 190.15.133.60

Indica el servidor que contiene la zona principal de autoridad para el dominio.

Servidores NS (Name Servers).

ns1.cedia.org.ec – IP: 190.15.131.2

ns2.cedia.org.ec – IP: 45.63.109.229

Los dos DNS están administrados por la Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia (CEDIA). Se identificó que ambos utilizan el software BIND versión 9.18.33 sobre Debian, lo cual puede ser útil para búsqueda de vulnerabilidades en versiones específicas.

El escaneo muestra que el dominio depende externamente de Microsoft para correo, los dns se ubican en servidores externos (CEDIA), implicando que, si la infraestructura se ve comprometida, también afectaría a la universidad. Al final se muestra el objetivo principal la ip 190.15.133.160 para el análisis en la siguiente fase.

Escaneo del servidor web con Nikto

Para detectar configuraciones inseguras en el servidor web que atiende el dominio http://upec.edu.ec, se hizo uso de la herramienta Nikto, que permite encontrar problemas comunes de seguridad en servidores HTTP como cabeceras mal configuradas, scripts expuestos o configuraciones inseguras.

Comando utilizado:

nikto -h http://upec.edu.ec


**Figura 8.** Escaneo del portal principal con la herramienta Nikto

Resultados obtenidos

Redirección HTTP a HTTPS.

El sitio redirige automáticamente de http:// a https://, indicando una buena práctica de seguridad.

Cabecera X-Frame-Options no configurada.

Esto podría permitir que el sitio sea cargado en un <iframe> desde otros sitios, abriendo la posibilidad a ataques de clickjacking.

Cabecera X-Content-Type-Options ausente.

La falta de esta cabecera permite al navegador adivinar (y potencialmente malinterpretar) el tipo de contenido de una respuesta. Esto podría permitir ataques de MIME-sniffing.

No se detectaron directorios CGI activos:

Aunque Nikto buscó directorios comunes de scripts CGI, no encontró ninguno expuesto.

Estas cabeceras faltantes no representan una vulnerabilidad crítica por sí solas, pero sí una superficie de ataque innecesaria, especialmente si se combinan con otros errores de configuración.

Evaluación del servidor HTTPS con Nikto

Se escaneó el servicio HTTPS del dominio objetivo mediante nikto, lo que permitió detectar posibles malas prácticas de configuración y vulnerabilidades pasivas.

Comando utilizado:

nikto -h https://upec.edu.ec

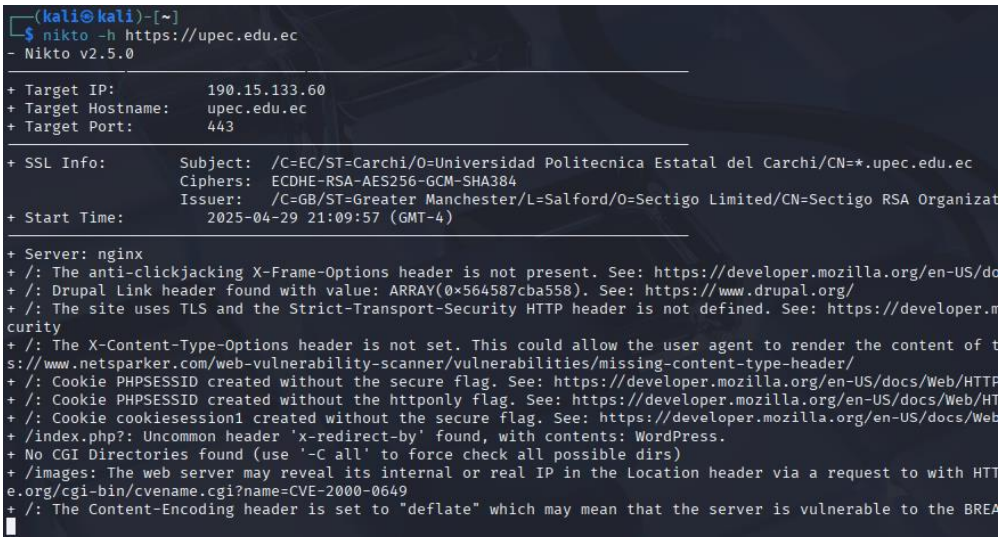



**Figura 9.** Reporte de vulnerabilidades de Nikto del portal principal

Resultados obtenidos

Información del certificado ssl.

CN: *.upec.edu.ec (certificado wildcard, cubre todos los subdominios)

Emisor: Sectigo RSA OV Secure Server CA

Cifrado activo: ECDHE-RSA-AES256-GCM-SHA384 (robusto, considerado seguro)

El uso de TLS moderno y certificado válido emitido por CA reconocida son buenas prácticas de seguridad.

Servidor identificado.

Nginx es un popular servidor web, objetivo frecuente de ataques si está mal configurado.

**Tabla 14.** Resumen de Vulnerabilidades y Riesgo del portal principal UPEC

| Elemento | Descripción Técnica | Riesgo |
|---|---|---|
| X-Frame-Options ausente | Puede permitir ataques de clickjacking | Medio |
| Strict-Transport-Security ausente | No obliga a los navegadores a usar HTTPS siempre | Medio |
| X-Content-Type-Options ausente | Riesgo de MIME sniffing | Medio |
| Cookies sin Secure y HttpOnly | Pueden ser interceptadas o leídas por scripts maliciosos | Alto |
| Drupal Header | Señal de uso de CMS Drupal | Bajo |
| Encabezado x-redirect-by: WordPress | Muestra coexistencia con WordPress | Bajo |
| BREACH | Usa compresión deflate, puede ser vulnerable a BREACH | Medio |
| IP interna expuesta | /images revela IP 10.100.100.126 | Alto |

Este servidor muestra una mezcla de tecnologías (Drupal y WordPress), con cabeceras faltantes que aumentan la superficie de ataque. Las cookies inseguras y la revelación de IP interna son hallazgos críticos.

### 4.7. FASE 5: EXPLOTACIÓN.

Dirección IP analizada: 190.15.129.70 tiene redireccionamiento a http://turismo.upec.edu.ec/turismolab/


**Figura 10.** Portal de la Carrera de Turismo (redireccionamiento de ip analizada)

Herramientas utilizadas: nmap, wpscan, gobuster, metasploit, nikto, dnsrecon, WPScan, curl, Whois, etc.

Resumen de hallazgos

Se identificaron un total de 19 vulnerabilidades asociadas a los servicios expuestos en la IP analizada:

- 14 vulnerabilidades críticas y altas en los servicios Apache, PHP y mod_perl en los puertos 80 y 443.

```
CVE-2024-38476  9.8     https://vulners.com/cve/CVE-2024-38476
CVE-2024-38474  9.8     https://vulners.com/cve/CVE-2024-38474
A5425A79-9D81-513A-9CC5-549D6321897C    9.8     https://vulners.com/githubexploit/A5425A79-9D81-513A-9CC5-549D6321897C  *EXPLOIT*
CVE-2024-38475  9.1     https://vulners.com/cve/CVE-2024-38475
5418A85B-F4B7-5BBD-B106-0800AC961C7A    9.1     https://vulners.com/githubexploit/5418A85B-F4B7-5BBD-B106-0800AC961C7A  *EXPLOIT*
2EF14600-503F-53AF-BA24-683481265D30    9.1     https://vulners.com/githubexploit/2EF14600-503F-53AF-BA24-683481265D30  *EXPLOIT*
0486EBEE-F207-570A-9AD8-33269E72220A    9.1     https://vulners.com/githubexploit/0486EBEE-F207-570A-9AD8-33269E72220A  *EXPLOIT*
B0A9E5E8-7CCC-5984-9922-A89F11D6BF38    8.2     https://vulners.com/githubexploit/B0A9E5E8-7CCC-5984-9922-A89F11D6BF38  *EXPLOIT*
CVE-2024-38473  8.1     https://vulners.com/cve/CVE-2024-38473
249A954E-0189-5182-AE95-31C866A057E1    8.1     https://vulners.com/githubexploit/249A954E-0189-5182-AE95-31C866A057E1  *EXPLOIT*
23079A70-8B37-56D2-9D37-F638EBF7F8B5    8.1     https://vulners.com/githubexploit/23079A70-8B37-56D2-9D37-F638EBF7F8B5  *EXPLOIT*
E606D7F4-5FA2-5907-B30E-367D6FFECD89    7.5     https://vulners.com/githubexploit/E606D7F4-5FA2-5907-B30E-367D6FFECD89  *EXPLOIT*
CVE-2024-40898  7.5     https://vulners.com/cve/CVE-2024-40898
CVE-2024-39573  7.5     https://vulners.com/cve/CVE-2024-39573
CVE-2024-38477  7.5     https://vulners.com/cve/CVE-2024-38477
CVE-2024-38472  7.5     https://vulners.com/cve/CVE-2024-38472
CVE-2024-27316  7.5     https://vulners.com/cve/CVE-2024-27316
CNVD-2024-20839 7.5     https://vulners.com/cnvd/CNVD-2024-20839
CDC791CD-A414-5ABE-A897-7CFA3C2D3D29    7.5     https://vulners.com/githubexploit/CDC791CD-A414-5ABE-A897-7CFA3C2D3D29  *EXPLOIT*
B5E74010-A082-5ECE-AB37-623A5B33FE7D    7.5     https://vulners.com/githubexploit/B5E74010-A082-5ECE-AB37-623A5B33FE7D  *EXPLOIT*
4B14D194-BDE3-5D7F-A262-A701F90DE667    7.5     https://vulners.com/githubexploit/4B14D194-BDE3-5D7F-A262-A701F90DE667  *EXPLOIT*
45D138AD-BEC6-552A-91EA-8816914CA7F4    7.5     https://vulners.com/githubexploit/45D138AD-BEC6-552A-91EA-8816914CA7F4  *EXPLOIT*
CVE-2023-38709  7.3     https://vulners.com/cve/CVE-2023-38709
CNVD-2024-36395 7.3     https://vulners.com/cnvd/CNVD-2024-36395
95499236-C9FE-56A6-9D7D-E943A24B633A    6.9     https://vulners.com/githubexploit/95499236-C9FE-56A6-9D7D-E943A24B633A  *EXPLOIT*
2C119FFA-ECE0-5E14-A4A4-354A2C38071A    6.9     https://vulners.com/githubexploit/2C119FFA-ECE0-5E14-A4A4-354A2C38071A  *EXPLOIT*
CVE-2024-24795  6.3     https://vulners.com/cve/CVE-2024-24795
CVE-2024-39884  6.2     https://vulners.com/cve/CVE-2024-39884
CVE-2024-36387  5.4     https://vulners.com/cve/CVE-2024-36387
```

**Figura 11.** Reporte de vulnerabilidades de la ip analizada

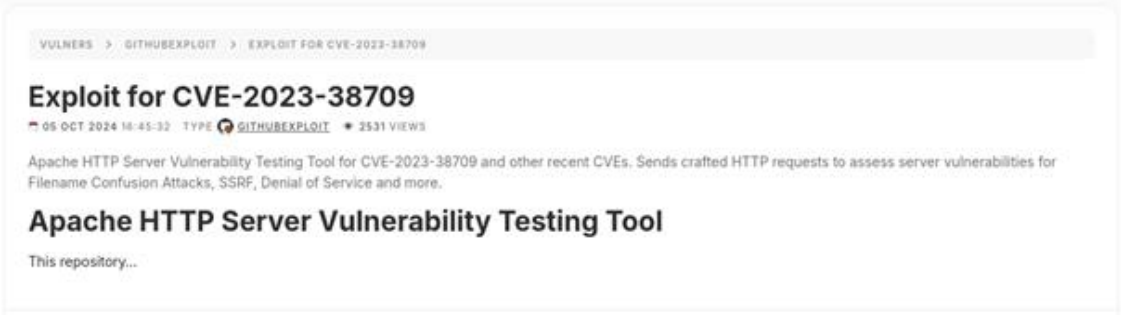



**Figura 12.** Descripción de la vulnerabilidad CVE-2023-38709

- 5 vulnerabilidades en el plugin TablePress v2.2.5 detectado en una instalación de WordPress accesible en /turismolab.

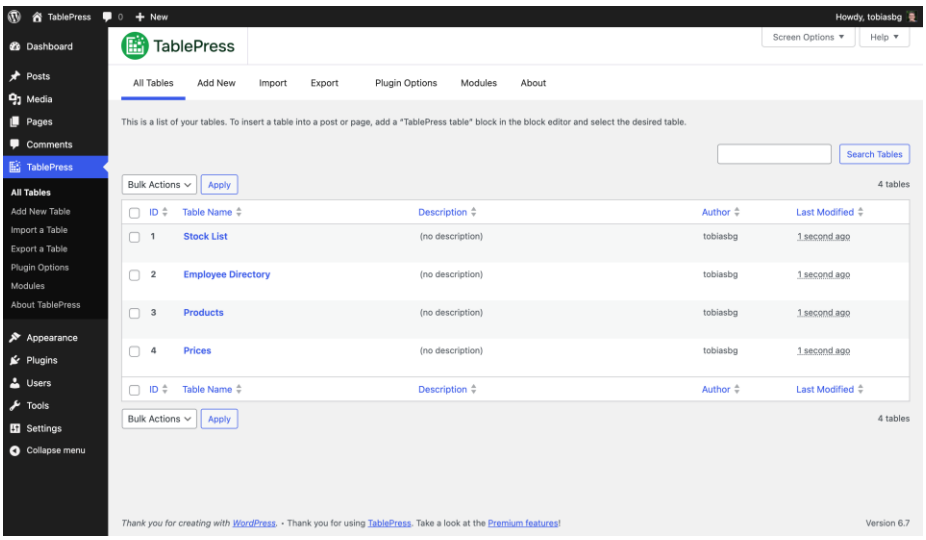



**Figura 13.** Plugin TablePress vulnerable de WordPress

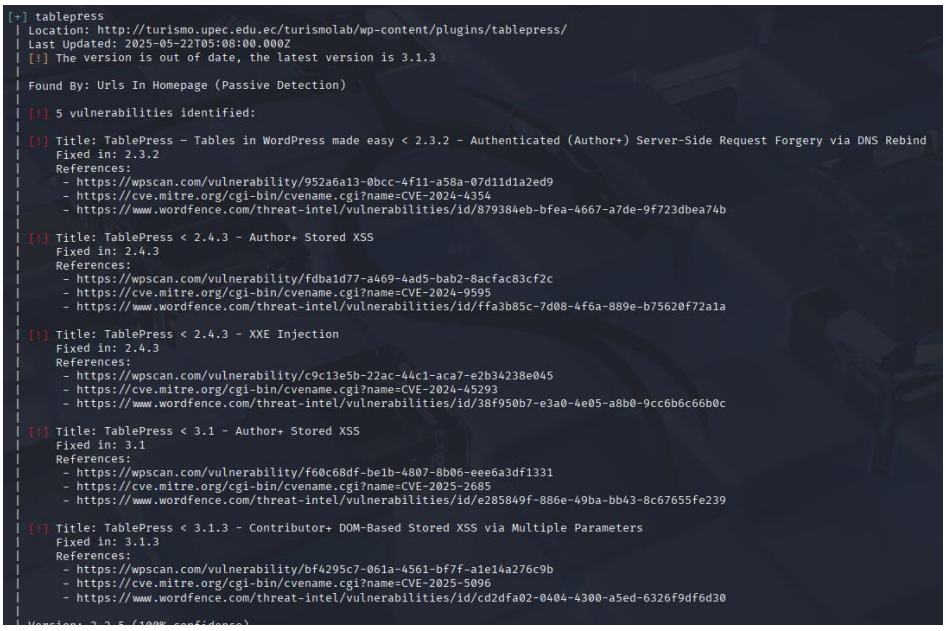



**Figura 14.** Reporte de Vulnerabilidades de Tablepress

Servicios detectados

Mediante escaneo con Nmap y Nikto, se identificaron los siguientes servicios:

Puerto 80 / 443: Apache/2.4.58 (Unix), OpenSSL/1.1.1w, PHP/8.2.12, mod_perl/2.0.12, Perl/v5.34.1

Redirección activa desde IP hacia subdominio

**Tabla 15.** Vulnerabilidades encontradas en servicios

| Vulnerabilidad | CVE | Severidad | Descripción | Referencia |
|---|---|---|---|---|
| Apache httpd < 2.4.59 múltiples fallos | CVE-2024-38474, 38475, 38476 | Crítica (9.8) | RCE y desbordamientos en Apache 2.4.58 | https://vulners.com |
| HTTP TRACE enabled | N/A | Media | Puede facilitar XST (Cross Site Tracing) | https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/TRACE |
| Directorios con listing habilitado | /icons/, /webalizer/ | Media | Riesgo de fuga de información interna del servidor | |
| Plugin expuesto en WordPress | TablePress | Media-Alta | Detectado en ruta pública con versión vulnerable | |

**Tabla 16.** Vulnerabilidades encontradas en WordPress (TablePress)

| CVE | Título | Tipo | Requiere autenticación | Referencia |
|---|---|---|---|---|
| CVE-2024-4354 | SSRF via DNS Rebind | Crítica | Sí (Author+) | https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-4354 |
| CVE-2024-9595 | Stored XSS | Alta | Sí (Author+) | https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-9595 |
| CVE-2024-45293 | XXE Injection | Alta | Sí (Author+) | https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-45293 |
| CVE-2025-2685 | Stored XSS | Alta | Sí (Author+) | |
| CVE-2025-5096 | DOM-Based Stored XSS | Media | Sí (Contributor+) | |

Aunque no se obtuvo acceso directo al sistema (shell o escalamiento de privilegios), se confirmó la existencia de múltiples vectores de ataque reales que permitirían, bajo un escenario combinado (por ejemplo, mediante phishing para capturar credenciales), una explotación crítica del sistema.

## 4.8. FASE 6: POST- EXPLOTACIÓN

El propósito de la post-explotación es identificar los activos internos, relaciones de confianza, credenciales, privilegios, y otros vectores que permitan:

Escalar privilegios

Mantener persistencia

Exfiltrar información valiosa

Durante esta evaluación, no se obtuvo acceso al sistema objetivo, por lo tanto, no se ejecutaron acciones de post-explotación directa. Sin embargo, se realizó una proyección basada en las vulnerabilidades críticas detectadas:

**Tabla 17.** Posibles Vectores de ataque

| Vulnerabilidad | Riesgo post-explotación potencial |
|---|---|
| Apache con RCE | Ejecución de comandos del sistema, posible creación de usuarios maliciosos |
| TablePress con SSRF | Acceso a servicios internos o explotación en la red interna (pivoting) |
| TablePress con XXE | Robo de archivos del servidor (ej. /etc/passwd) si el parser XML es vulnerable |
| Stored XSS | Robo de sesión de administradores si se logra ejecutar JS en su navegador |

A pesar de no haberse obtenido una shell o sesión autenticada, los vectores descubiertos presentan una probabilidad alta de explotación total en un escenario real, especialmente si un atacante logra credenciales válidas o eleva privilegios en la aplicación.

## 4.9. FASE 7: REPORTE

### 4.9.1. Alcance

Evaluación dirigida sobre la IP 190.15.129.70 perteneciente a la infraestructura de la Universidad Politécnica Estatal del Carchi (UPEC), enfocada en descubrir, analizar y reportar vulnerabilidades reales bajo el marco de la metodología PTES.

### 4.9.2. Resultado general

Fase de reconocimiento: Identificación de subdominios, servicios y rutas accesibles.

Fase de escaneo: Enumeración de puertos, servicios y tecnologías.

Fase de explotación: Detección de 19 vulnerabilidades, 14 en servicios y 5 en el CMS WordPress.

Fase de post-explotación: Proyección de impacto basado en escenarios de explotación realista.

Se presenta a continuación un listado detallado de las vulnerabilidades detectadas junto con sus respectivas acciones de remediación:

**Tabla 18.** Soluciones y Mitigaciones Recomendadas

| Vulnerabilidad | CVE(s) | Solución recomendada |
|---|---|---|
| Apache 2.4.58 vulnerable a múltiples RCEs | CVE-2024-38474, 38475, 38476 | Actualizar a Apache 2.4.59 o superior. Revisar módulos activos y aplicar hardening de configuración. |
| HTTP TRACE habilitado | N/A | Deshabilitar TRACE en la configuración de Apache con 'TraceEnable off'. |
| Listado de directorios: /icons/, /webalizer/ | N/A | Deshabilitar listado con 'Options -Indexes' en la configuración de Apache. |
| Estadísticas expuestas: /webalizer/ | N/A | Eliminar el acceso o proteger con autenticación básica o restricción por IP. |
| WordPress potencialmente desactualizado | N/A | Actualizar WordPress a la versión más reciente y mantener actualizaciones automáticas. |
| TablePress v2.2.5 vulnerable | CVE-2024-4354, CVE-2024-9595, CVE-2024-45293, CVE-2025-2685, CVE-2025-5096 | Actualizar TablePress a la versión 3.1.3 o superior. Limitar privilegios de usuario a roles confiables. |
| XSS almacenado (plugin TablePress) | Ver CVEs anteriores | Actualizar plugin y aplicar políticas CSP. Validar y sanitizar entradas en formularios. |
| XXE en plugin TablePress | CVE-2024-45293 | Actualizar plugin. Deshabilitar procesamiento de entidades externas en XML si es aplicable. |
| SSRF vía DNS Rebind (plugin) | CVE-2024-4354 | Actualizar plugin. Filtrar URLs en entrada. Restringir acceso del servidor a redes internas. |

Se realizó una evaluación de seguridad sobre la IP 190.15.129.70 perteneciente a la UPEC. Se identificaron un total de 19 vulnerabilidades, algunas de ellas críticas, como ejecución remota de código (RCE), vulnerabilidades en un plugin de WordPress ampliamente utilizado (TablePress), y mecanismos de ataque como XSS, SSRF y XXE. Se recomienda proceder con actualizaciones inmediatas, segmentación de servicios, revisión de permisos y una auditoría de configuración web. La explotación de estas fallas, bajo un escenario combinado, permitiría un compromiso total del sistema.

### 4.10. DISCUSIÓN

Los hallazgos obtenidos durante el proceso de pentesting aplicado a la infraestructura tecnológica de la Universidad Politécnica Estatal del Carchi (UPEC)

evidencian logros significativos que se alinean con los objetivos específicos planteados en esta investigación.

En relación con el primer objetivo, que consistía en "Analizar y fundamentar bibliográficamente la evaluación de vulnerabilidades de una red", se realizó una revisión exhaustiva de investigaciones previas tanto a nivel nacional como internacional. A diferencia de antecedentes como los de Añazco & Ortiz (2020), quienes emplearon metodologías como OWASP y NIST para evaluar portales educativos, esta tesis aplicó la metodología PTES en su totalidad, abarcando desde la planificación hasta la post-explotación. Además, se integraron marcos como MITRE ATT&CK y STRIDE, ampliando el análisis más allá de los métodos tradicionales encontrados en estudios como el de Briones (2020) o Medina (2021).

En cuanto al segundo objetivo, "Estimar la efectividad de los controles de seguridad mediante pruebas de intrusión controladas", se llevaron a cabo escaneos automatizados y verificaciones manuales, utilizando herramientas como Nmap, Nikto, OpenVAS, Metasploit, Burp Suite y WPScan. A diferencia del trabajo de Alarcón (2022), que se centró en un entorno de salud, este estudio abordó servicios académicos reales y activos públicos de una universidad, identificando 14 vulnerabilidades técnicas asociadas a servicios en puertos 80 y 443, y 5 vulnerabilidades críticas dentro del plugin TablePress versión 2.2.5. Estas vulnerabilidades fueron reportadas conforme a estándares CVE, con referencias a exploits documentados en vulners.com y Wordfence. El servidor objetivo ejecutaba Apache/2.4.58 con PHP 8.2.12 sobre Unix, evidenciando múltiples vectores de ataque que podrían comprometer datos sensibles institucionales.

Respecto al tercer objetivo, "Desarrollar un informe que documente procesos, herramientas y mejores prácticas para la evaluación de seguridad", se construyó un informe técnico completo siguiendo el esquema del PTES. Este incluye fases de modelado de amenazas, pruebas de explotación, y recomendaciones específicas para cada hallazgo. Esta sistematización no fue abordada con tal nivel de detalle en antecedentes como los de Rojas Buenaño (2020) o Chilán (2022), quienes enfocaron sus pruebas en entornos simulados o sin técnicas avanzadas de explotación y clasificación.

En términos metodológicos, se aplicó un enfoque mixto, experimental y empírico, en concordancia con la propuesta de Hernández y Torres (2023), permitiendo no solo

identificar vulnerabilidades, sino simular escenarios reales de explotación sin comprometer la integridad de los sistemas. Esto refuerza la hipótesis planteada, según la cual la infraestructura de la UPEC contiene vulnerabilidades significativas debido a la falta de políticas estructuradas de evaluación de seguridad.

Finalmente, este trabajo de investigación reafirma los hallazgos de investigaciones anteriores respecto a la falta de personal especializado, presupuestos limitados y la escasa institucionalización de procesos de ciberseguridad, pero a su vez amplía el conocimiento al demostrar que el pentesting estructurado es viable técnica y económicamente dentro del contexto universitario ecuatoriano, incluso bajo recursos limitados.

**Tabla 19.** Comparación de Antecedentes

| Aspectos Clave de la Investigación | Añazco & Ortiz (2020) | Briones (2020) | Medina (2021) | Rodríguez (2020) | Alarcón (2022) | Chilán (2022) | Rojas Buenaño (2020) | Mi Investigación (UPEC) |
|---|---|---|---|---|---|---|---|---|
| Foco Principal y Contexto del Estudio | Investigación de vulnerabilidades en portales web de instituciones educativas ecuatorianas | Aplicación de hacking ético para identificar amenazas y vulnerabilidades en una red universitaria (Universidad Estatal del Sur de Manabí). | Exploración del hacking ético como herramienta para la seguridad informática, detallando un proceso sistemático. | Análisis de herramientas fundamentales para el hacking ético en el contexto de redes cubanas. | Pentesting en la red de Salud Valle del Mantaro, utilizando 3 sistemas administrados por el área de informática. | Aplicación de hacking ético para mejorar la seguridad en la red de equipos informáticos en UPOCAM.[1] | Aplicación de hacking ético para evaluar la seguridad informática en la infraestructura de una empresa industrial (Plasticaucho Industrial S.A.). | Evaluación de vulnerabilidades en sistemas informáticos e infraestructura tecnológica de la Universidad Politécnica Estatal del Carchi (UPEC) mediante pentesting estructurado. |
| Metodología y Enfoque | OWASP y NIST. | Hacking ético (general). | Proceso sistemático para pruebas de vulnerabilidad. | Búsqueda bibliográfica, selección de herramientas adaptadas a redes nacionales. | Uso de SO Parrot con herramientas integradas y scripts de GitHub.[1] | Mixta (cuantitativa y cualitativa), encuestas, observación, escaneo de red. | Aplicada (campo y documental-bibliográfica), etapas de hacking ético (Planificación, Recolección, Enumeración, Análisis, Explotación, Documentación). | Mixta (cualitativo y cuantitativo), Experimental (pruebas de intrusión controladas), de Campo (entorno real). Metodología PTES, complementada con STRIDE y MITRE ATT&CK |
| Herramientas Clave (énfasis en Open Source) | No especificadas, pero se infiere uso de herramientas | No especificadas, pero se infiere uso de herramientas | No especificadas, pero se infiere uso de herramientas para pruebas | Nmap, OpenVAS, BetterCap, Metasploit Framework, Armitage, | Parrot OS con herramientas integradas y scripts de GitHub | Herramientas de código abierto proporcionadas por | Google Hacking, TheHarvester, Whois, Nslookup, Fierce, FOCA, | Nmap, OpenVAS, Metasploit, Burp Suite Community Edition, |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | para OWASP/NIST. | de hacking ético. | de vulnerabilidad | OWASP ZAP, DVL/DVWA | | tecnologías de la información. | Nmap, Nessus, OpenVAS, Metasploit. | Wireshark, Nikto, Hydra, John the Ripper, theHarvester, DNSDumpster, WHOIS (todas de código abierto). |
| Resultados Clave y Tipo de Vulnerabilidades | Evaluó y buscó mejorar la seguridad de portales web. | Destacó la importancia de capacitación y herramientas efectivas para seguridad de red | Enfatizó la importancia de planificación y reconocimiento pasivo. | Describió herramientas y su aplicación para encontrar vulnerabilidades en redes | Redujo vulnerabilidades en un 68%, mejorando la seguridad de los sistemas de salud. | Identificó riesgos y vulnerabilidades en la red de UPOCAM. | Identificó debilidades en equipos Linux y vulnerabilidades críticas en la infraestructura industrial.[4] | Identificación de 19 vulnerabilidades: 14 críticas/altas en Apache, PHP, mod_perl (puertos 80/443); 5 en plugin TablePress v2.2.5 de WordPress. Falencias significativas en postura de seguridad. |
| Contribución/Aporte Distintivo | Abordó la preocupación por el cibercrimen en educación. | Subrayó la necesidad de capacitación del personal administrativo. | Sistematizó el proceso de pruebas de vulnerabilidad. | Proporcionó una guía de herramientas esenciales para el contexto cubano. | Demostró una reducción cuantificable de vulnerabilidades. | Propuso la aplicación de hacking ético para soluciones concretas. | Simulación de ataques para evaluar nivel de seguridad en un entorno industrial. | Aplicación integral de PTES, STRIDE y MITRE ATT&CK en un entorno universitario real, identificando vulnerabilidades críticas y proponiendo soluciones concretas y |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | viables con recursos limitados |
| Limitaciones y Desafíos Implícitos/Explícitos | No especificado, pero el foco en portales web sugiere un alcance limitado. | No especificado, pero la importancia de la capacitación sugiere una brecha de conocimiento. | No especificado, pero el enfoque en un "proceso" podría implicar menos énfasis en la ejecución real. | Escaso conocimiento de herramientas entre especialistas cubanos. | No especificado, pero "scripts creados por la comunidad" puede implicar dependencia externa. | No especificado, pero la necesidad de mejorar la seguridad sugiere desafíos existentes. | Falta de credenciales en equipos Linux, lo que facilitaba el acceso. | Falta de personal especializado, presupuesto limitado, ausencia de entornos de pruebas aislados, delegación de tareas a estudiantes. |
| Relevancia para Mi Estudio y Diferenciación | Similar en foco (educación, web), pero mi estudio es más amplio (infraestructura general) y usa una metodología más integral (PTES, STRIDE, MITRE ATT&CK). | Comparte el contexto universitario y la importancia de la seguridad en red, pero mi estudio profundiza en la metodología y resultados técnicos específicos. | Refuerza la importancia de la planificación y el reconocimiento, aspectos clave en mi fase inicial de PTES. | Proporciona un catálogo de herramientas que valida algunas de mis elecciones (Nmap, OpenVAS, Metasploit) y subraya la importancia del "cerebro humano" en el pentesting. | Demuestra la efectividad del pentesting en un entorno real, similar a mi objetivo de cuantificar la reducción de riesgos. | Coincide en el uso de metodologías mixtas y herramientas de código abierto para mejorar la seguridad en entornos organizacionales. | Ofrece un marco metodológico detallado para el hacking ético, que mi estudio expande al aplicar PTES y marcos de modelado de amenazas | |

# V. CONCLUSIONES Y RECOMENDACIONES

## 5.1. CONCLUSIONES

Al analizar y fundamentar bibliográficamente la evaluación de vulnerabilidades, se logró identificar y comparar las metodologías más utilizadas a nivel internacional como PTES, OWASP y NIST. Se integró un marco conceptual sólido sobre técnicas, herramientas y clasificación de vulnerabilidades, demostrando que la metodología PTES ofrece un enfoque estructurado y adaptable a instituciones de educación superior.

Al estimar la efectividad de los controles de seguridad mediante pruebas de intrusión controladas, se ejecutaron con éxito múltiples pruebas sobre la IP 190.15.129.70, identificando 14 vulnerabilidades asociadas al servidor web y al menos 5 vulnerabilidades activas en el plugin TablePress. Se comprobó que los controles implementados son insuficientes para proteger los activos críticos, validando así la efectividad de las técnicas de pentesting aplicadas.

Al desarrollar un informe técnico, se elaboró una documentación detallada siguiendo las fases del PTES. El informe incluye recolección de evidencias, análisis de impacto, clasificación de vulnerabilidades y recomendaciones específicas. Este entregable no solo cumple con el propósito académico, sino que también proporciona a la UPEC una herramienta útil para reforzar sus políticas de ciberseguridad institucional.

## 5.2. RECOMENDACIONES

Institucionalizar una política de evaluación de seguridad periódica en la UPEC, utilizando la metodología PTES aplicada en este trabajo, con el fin de garantizar la continuidad de las evaluaciones técnicas y la implementación de mejoras continuas.

Realizar la actualización inmediata del servidor Apache, los componentes PHP y el plugin TablePress identificados como vulnerables, así como implementar mecanismos de defensa en profundidad tales como un Web Application Firewall (WAF) y segmentación de red.

Incorporar el informe técnico generado en esta investigación como material guía para la capacitación del personal del área TIC, y como documento base para la realización de auditorías de seguridad regulares. Dicho informe constituye un insumo directo para el fortalecimiento de las políticas de ciberseguridad institucional

## VI. REFERENCIAS BIBLIOGRÁFICAS


Aceituno Canal, V. (2020). *Seguridad de la información: Expectativas, riesgos y técnicas de protección*. Limusa.

Acosta, J. J. (2022). *Pentesting en entornos controlados* (Tesis de grado). Universidad de la Laguna, España. http://riull.ull.es/xmlui/handle/915/28744

Alarcón, L. A. (2022). *Pentesting en la red de salud valle del Mantaro para la implementación de políticas de seguridad* (Tesis de grado). Universidad Nacional del centro de Perú, Perú.

Añazco Bedón, J. P., & Ortiz Rodríguez, B. F. (2020). *Análisis de vulnerabilidades en el portal web de una institución de educación superior del Ecuador mediante hacking ético* [Trabajo de titulación, Universidad de Las Américas]. https://dspace.udla.edu.ec/bitstream/33000/10613/1/UDLA-EC-TIS-2019-03.pdf

Areitio Bertolín, J. (2021). *Seguridad de la información: Redes, informática y sistemas de información*. Paraninfo.

Briones, I. E. (2020). *Aplicación De Hacking Ético Para La Determinación De Amenazas, Riesgos Y Vulnerabilidades En La Red De La Universidad Estatal Del Sur De Manabí* (Tesis de grado). Universidad Estatal del Sur de Manabí, Ecuador.

Campi Domínguez, C. A. (2023). *Estrategia para la evaluación de vulnerabilidades del sistema de notas utilizando técnicas de hacking ético en la escuela Mahatma Gandhi* [Tesis de grado, Universidad Técnica de Babahoyo]. Repositorio Institucional UTB. https://dspace.utb.edu.ec/items/d5abbadb-58ce-446e-a1c0-df4acafc6465

Chilán, K.(2022). *Aplicación De Hacking Ético Para Mejorar La Seguridad En La Red De Los Equipos Informáticos En La Upocam.* http://repositorio.unesum.edu.ec/handle/53000/4581

Costas Santos, J. (2022). *Seguridad informática*. RA-MA Editorial

Castillo Vera, O. (2021). *Evaluación de metodologías de hacking ético para el diagnóstico de vulnerabilidades de la seguridad informática en una empresa de servicios logísticos* [Tesis de grado, Universidad Señor de Sipán]. Repositorio Institucional USS. https://repositorio.uss.edu.pe/handle/20.500.12802/9148

Díaz Orueta, G., Alzórriz Armendáriz, I., Sancristóbal Ruiz, E., & Castro Gil, M. A. (2020). *Procesos y herramientas para la seguridad de redes*. UNED.

Galarza García, D. E. (2020). *Estrategia para la evaluación de vulnerabilidades del sistema de notas de instituciones educativas utilizando técnicas de hacking ético. Caso de estudio: Instituto Tecnológico Quito* [Trabajo de titulación de maestría, Escuela Politécnica Nacional]. Repositorio Institucional EPN. https://bibdigital.epn.edu.ec/handle/15000/21377

García Pérez, K. A. (2021). *Aplicación de hacking ético mediante test de intrusión Pentesting para la detección y análisis de vulnerabilidades en la red inalámbrica de una institución educativa de la provincia de Santa Elena* [Tesis de grado, Universidad Estatal Península de Santa Elena]. Repositorio Digital UPSE. https://repositorio.upse.edu.ec/items/b6f72046-b5e9-4f67-8ad6-afd9b2dea875

Gavilán Fontecha, E. (2020). *Hacking ético y ciberseguridad*. 0xWord.

Gómez Vieites, Á. (2020). **Enciclopedia de la seguridad informática**. RA-MA Editorial.

Gómez, R., Sánchez, A., & Martínez, P. (2023). **Manual de pruebas de penetración ética**. Editorial Seguridad Digital.

Hernández, J., & Torres, M. (2023). **Documentación y análisis de vulnerabilidades en sistemas informáticos**. Cybersecurity Journal español, 8(4), 112-128.

López Neira, A., & Ruiz Spohr, J. (2022). *Seguridad informática*. Editorial Editex.

López-Martínez, F. (2022). **Fundamentos de ciberseguridad y hacking ético**. Editorial Tecnológica Española.

Martínez-González, A., Rodríguez, S., & López, R. (2023). **Protocolos de seguridad en pruebas de penetración**. Revista Latinoamericana de Seguridad Informática, 12(3), 78-95.

Medina, E. (2021). *Una herramienta para la seguridad informática*. Universidad Piloto de Colombia, Colombia. http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/2932/Trabajo%20de%20grado.pdf?sequence=1&isAllowed=y

Ramírez-Sánchez, E. (2024). **Metodologías actuales en análisis de vulnerabilidades**. Seguridad Digital Aplicada, 5(1), 15-32.

Roa Buendía, J. F. (2023). *Seguridad informática* (2ª edición). McGraw-Hill.

Rodríguez, A. E. (2020). *Herramientas fundamentales para el hacking ético*. Revista Cubana de Informática Médica, Cuba. http://scielo.sld.cu/pdf/rcim/v12n1/1684-1859-rcim-12-01-116.pdf

Rojas Buenaño, A. (2020). Hacking Etico Para Analizar Vulnerabilidades. Universidad Tecnica de Ambato, Ambato, Ecuador. https://repositorio.uta.edu.ec/bitstream/123456789/28102/1/Tesis_%20t1417si.pdf

## VII. ANEXOS

Anexo 1. Acta de la sustentación de Predefensa del TIC

**UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI**

**FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES**

**CARRERA DE COMPUTACIÓN**

**ACTA**

**DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR CON ENFOQUE EN INVESTIGACIÓN**

| | | | |
|---|---|---|---|
| **ESTUDIANTE:** | Changuan Rodriguez Roberth Guillermo | **CÉDULA DE IDENTIDAD:** | 1002672614 |
| **PERIODO ACADÉMICO:** | PAO 2025II | | |
| **PRESIDENTE TRIBUNAL** | MSC. JEFFERY NARANJO CEDEÑO | **DOCENTE TUTOR:** | MSC. MILTON DEL HIERRO |
| **DOCENTE:** | MSC. LUIS PATIÑO HERNANDEZ | | |
| **TEMA DEL TIC:** | Aplicación de técnicas de pentesting para la evaluación de vulnerabilidades | | |

| No. | CATEGORÍA | Evaluación cuantitativa | OBSERVACIONES Y RECOMENDACIONES |
|---|---|---|---|
| 1 | PROBLEMA - OBJETIVOS | 8,33 | Mejorar la redacción de los objetivos |
| 2 | FUNDAMENTACIÓN TEÓRICA | 9,33 | |
| 3 | METODOLOGÍA | 9,00 | |
| 4 | RESULTADOS | 9,00 | |
| 5 | DISCUSIÓN | 9,00 | |
| 6 | CONCLUSIONES Y RECOMENDACIONES | 9,00 | |
| 7 | DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL | 7,00 | Hacer un video y demostracion de los resultados. Organizar de mejor manera los tiempos de exposicion |
| 8 | FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN | 8,00 | Revisar la redacción y ortografía |

sriendo una nota de: 8,47 Por lo tanto, **APRUEBA** ; debiendo el o los investigadores acatar el siguiente artículo:

Art. 66.- De la aprobación de la pre defensa del informe final de TIC.- El estudiante deberá obtener una nota mínima de 7/10; al finalizar el proceso de pre-defensa se procederá a levantar el acta correspondiente. En el caso de aprobar con observaciones el estudiante deberá adjuntar el informe final de cumplimiento de observaciones y recomendaciones emitido por el Tribunal previo a la defensa final en un término máximo de 10 días.

Para constancia del presente, firman en la ciudad de Tulcán el lunes, 15 de diciembre de 2025

MSC. JEFFERY NARANJO CEDEÑO
**PRESIDENTE TRIBUNAL**

MSC. MILTON DEL HIERRO
**DOCENTE TUTOR**

MSC. LUIS PATIÑO HERNANDEZ
**DOCENTE**

Anexo 2. Certificado del abstract por parte de idiomas

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FOREIGN AND NATIVE LANGUAGES CENTER

**ABSTRACT- EVALUATION SHEET**

**NAME:** Roberth Guillermo Changuan Rodríguez
**DATE**: Lunes, 10 de noviembre de 2025
**Topic:** "Aplicación de técnicas de pentesting para la evaluación de vulnerabilidades"
**MARKS AWARDED** **QUANTITATIVE AND QUALITATIVE**

| | | | | |
|---|---|---|---|---|
| VOCABULARY AND WORD USE | Use new learnt vocabulary and precise words related to the topic | Use a little new vocabulary and some appropriate words related to the topic | Use basic vocabulary and simplistic words related to the topic | Limited vocabulary and inadequate words related to the topic |
| | EXCELLENT: 2 ☐ | GOOD: 1,5 ■ | AVERAGE: 1 ☐ | LIMITED: 0,5 ☐ |
| WRITING COHESION | Clear and logical progression of ideas and supporting paragraphs. | Adequate progression of ideas and supporting paragraphs. | Some progression of ideas and supporting paragraphs. | Inadequate ideas and supporting paragraphs. |
| De | EXCELLENT: 2 ■ | GOOD: 1,5 ☐ | AVERAGE: 1 ☐ | LIMITED: 0,5 ☐ |
| ARGUMENT | The message has been communicated very well and identify the type of text | The message has been communicated appropriately and identify the type of text | Some of the message has been communicated and the type of text is little confusing | The message hasn't been communicated and the type of text is inadequate |
| | EXCELLENT: 2 ■ | GOOD: 1,5 ☐ | AVERAGE: 1 ☐ | LIMITED: 0,5 ☐ |
| CREATIVITY | Outstanding flow of ideas and events | Good flow of ideas and events | Average flow of ideas and events | Poor flow of ideas and events |
| | EXCELLENT: 2 ☐ | GOOD: 1,5 ■ | AVERAGE: 1 ☐ | LIMITED: 0,5 ☐ |
| SCIENTIFIC SUSTAINABILITY | Reasonable, specific and supportable opinion or thesis statement | Minor errors when supporting the thesis statement | Some errors when supporting the thesis statement | Lots of errors when supporting the thesis statement |
| | EXCELLENT: 2 ■ | GOOD: 1,5 ☐ | AVERAGE: 1 ☐ | LIMITED: 0,5 ☐ |

| | | |
|---|---|---|
| TOTAL/AVERAGE | 9 - 10: EXCELLENT<br>7 - 8,9: GOOD<br>5 - 6,9: AVERAGE<br>0 - 4,9: LIMITED | TOTAL 9 |

1

**UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI- FOREIGN AND NATIVE LANGUAGES CENTER**

## Informe sobre el Abstract de Artículo Científico o Investigación.

**Autor:** Roberth Guillermo Changuan Rodríguez
**Fecha de recepción del abstract**: Miércoles, 5 de noviembre de 2025
**Fecha de entrega del informe**: Lunes, 10 de noviembre de 2025

El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

## Observaciones:

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según la rúbrica de evaluación de la traducción en Inglés, ésta alcanza un valor de 9; por lo cual se valida dicho trabajo.

Atentamente

Firmado electrónicamente por: MARTHA ARACELLY VIVEROS ALMEIDA
Validez del documento con firma electrónica

MA. Martha Viveros
Responsable del
CIDEN

2

Anexo 3. Guía de instalación: VirtualBox + Kali Linux (Windows 10)

Esta guía muestra los pasos necesarios para instalar Oracle VirtualBox en un sistema anfitrión con Windows 10 y crear una máquina virtual con Kali Linux. Está pensada para ser usada como entorno de pruebas controladas en el contexto de evaluaciones de seguridad autorizadas (pentesting). Se incluyen sugerencias de configuración de red relevantes para pruebas que requirieron evadir restricciones de firewall institucional (uso de 'Bridged Adapter').

**Notas previas**

• Asegurarse de tener permiso institucional por escrito para ejecutar pruebas.

• Requisitos mínimos en el anfitrión: Windows 10 (64-bit), 8 GB RAM recomendado, 50 GB libre en disco.

• Descarga de Kali Linux: https://www.kali.org/get-kali/ (elegir la ISO 'Kali Linux - 64-bit')

• Descarga de VirtualBox: https://www.virtualbox.org/wiki/Downloads (elegir el instalador para Windows hosts)

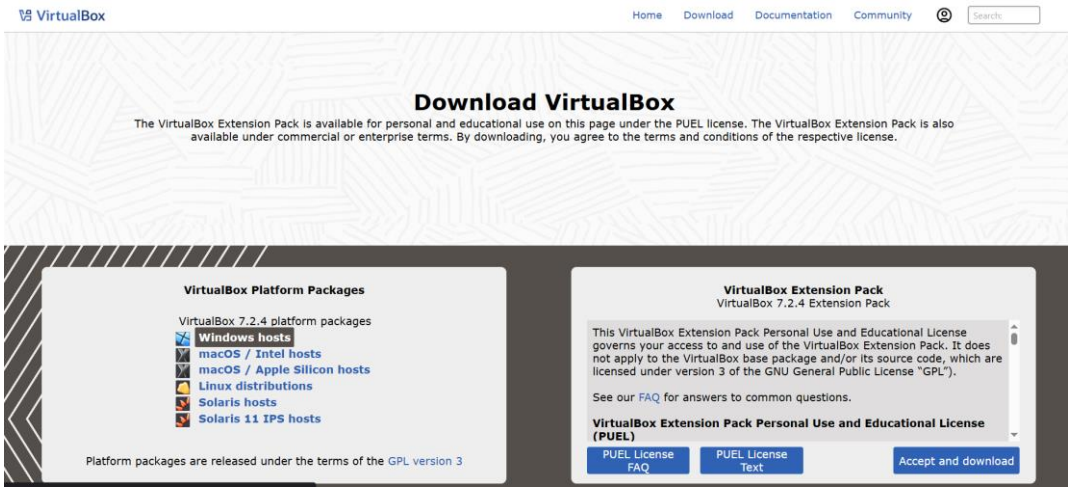
**Descargar e instalar VirtualBox**

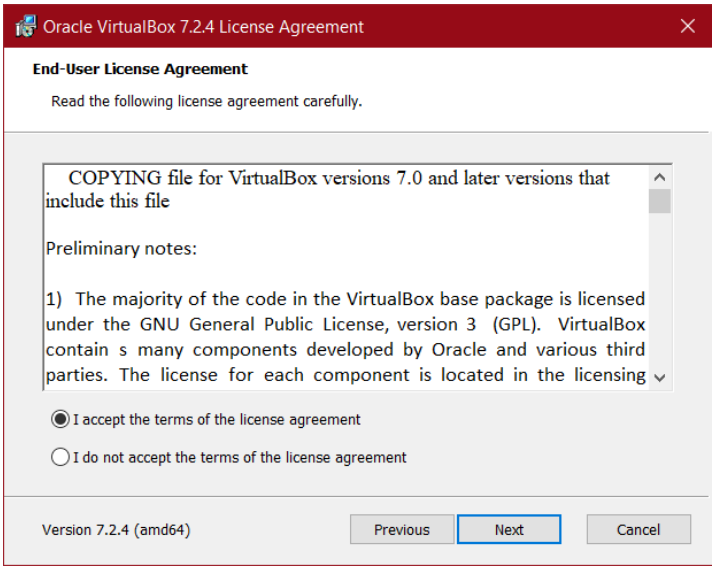
1) Abrir un navegador e ir a la página de descargas de VirtualBox.

2) Descargar 'Windows hosts' (instalador .exe).

3) Ejecutar el instalador con permisos de Administrador. Aceptar el acuerdo de licencia y las opciones por defecto.

4) Si Windows muestra advertencias de red o instalación de drivers, aceptar e instalar.



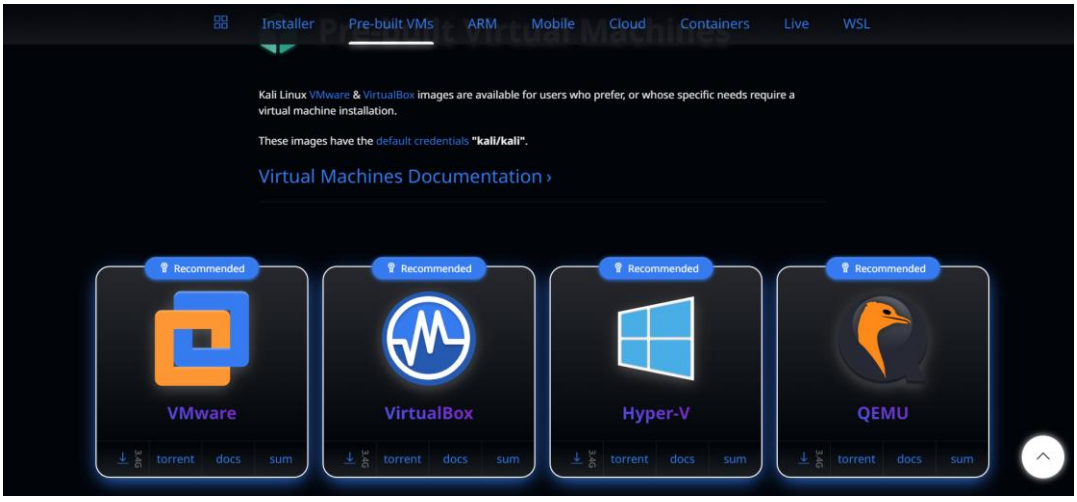


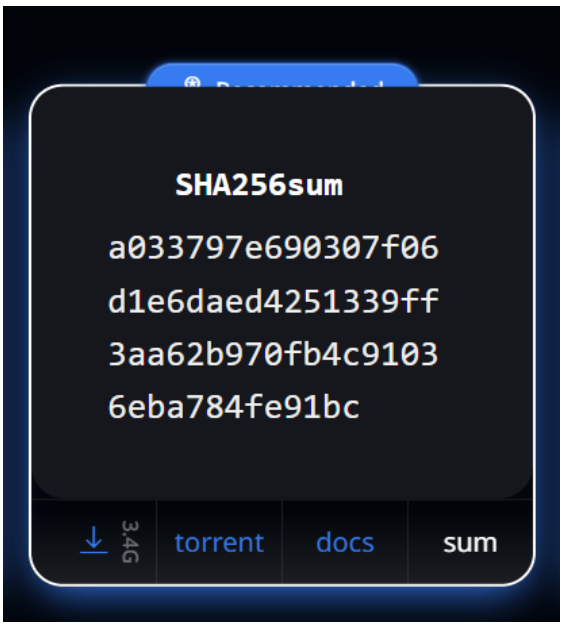

**Descargar Kali Linux (ISO)**

1) Ir a la web oficial de Kali: https://www.kali.org/get-kali/

2) Descargar la ISO de la versión 'Virtual Machines' o 'Live' 64-bit según se prefiera.

3) Verificar la suma SHA256 si es posible para confirmar la integridad de la ISO.





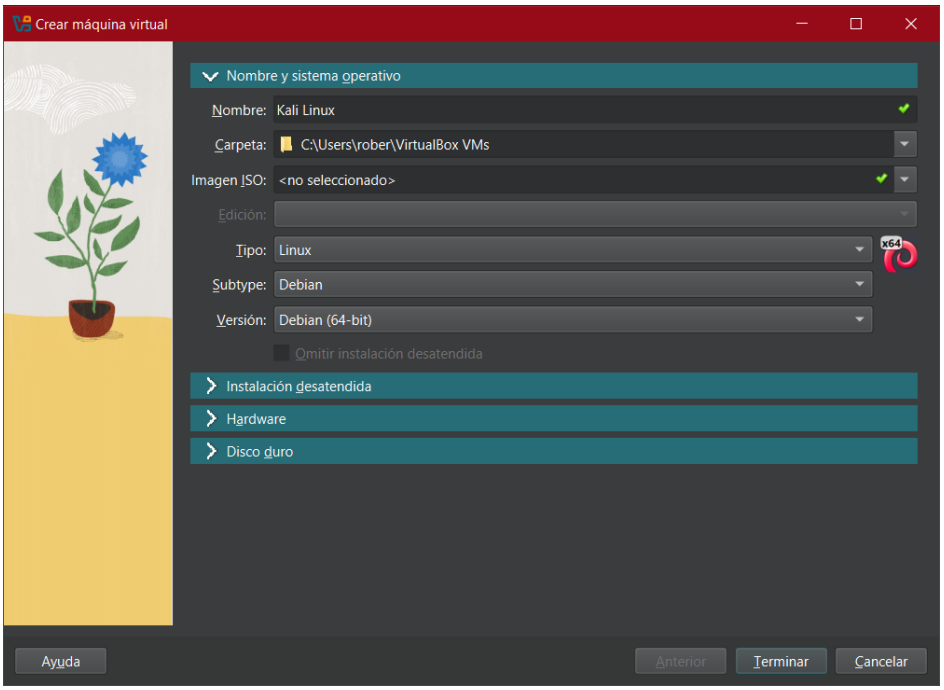

**Crear nueva máquina virtual en VirtualBox**

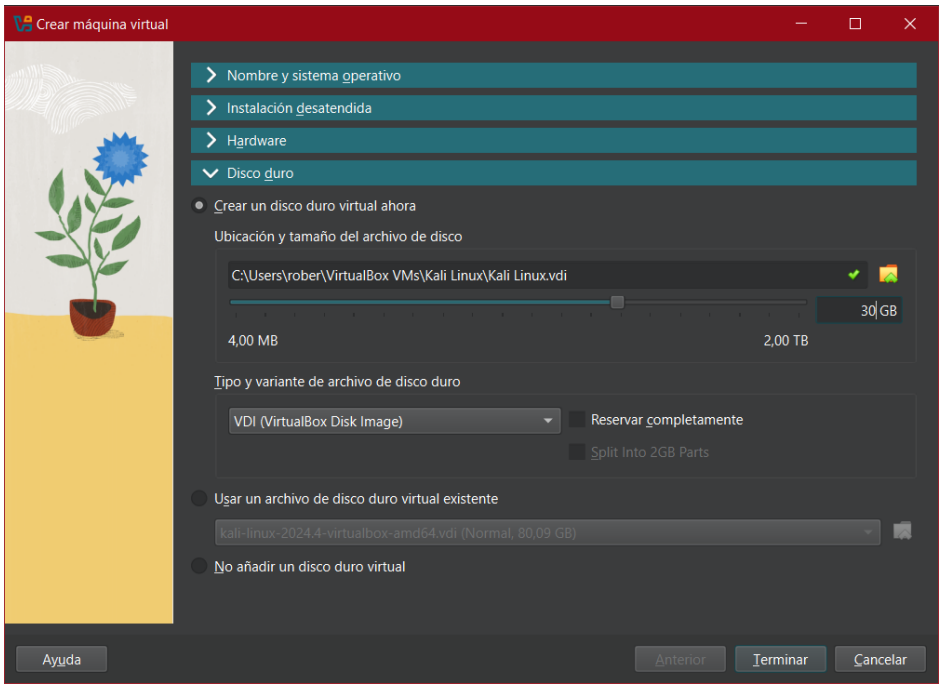
1) Abrir VirtualBox y hacer clic en 'Nuevo'.

2) Nombre: 'Kali-Linux'. Tipo: 'Linux'. Versión: 'Debian (64-bit)' o 'Other Linux (64-bit)' si no aparece.

3) Asignar memoria RAM: mínimo 2048 MB (recomendado 4096 MB si dispone).

4) Crear un disco duro virtual (VDI) dinámico de al menos 30 GB.



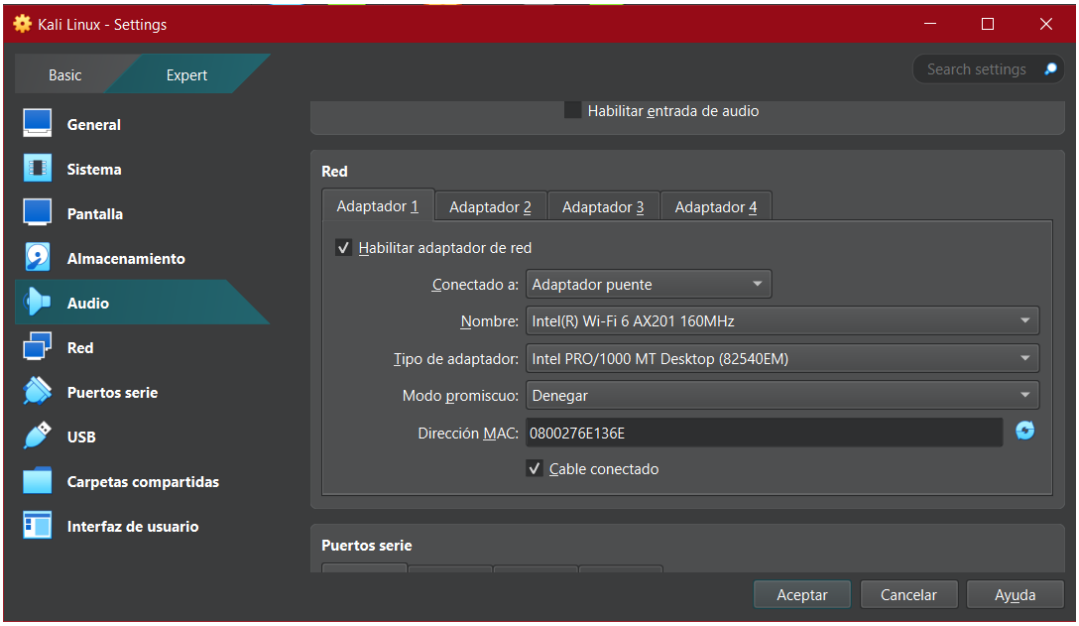

**Configurar red: Bridged Adapter (sugerido para pruebas desde red institucional)**

1) En Configuración → Red → Adapter 1: activar 'Conectado a: Bridged Adapter'.

2) Seleccionar el adaptador físico del host (por ejemplo, 'Intel(R) Ethernet...').

3) Se puede fijar la MAC si su laboratorio lo requiere. Guardar cambios.

Justificación: El modo 'Bridged' conecta la VM a la misma red física que el host, lo que puede ser necesario para que la VM tenga una IP en el mismo segmento y evadir reglas que actúan por subred. Usar sólo si está autorizado.




**Arrancar la VM e instalar Kali Linux**

1) Iniciar la VM y el instalador de Kali arrancará desde la ISO.

2) Seleccionar 'Graphical install' o 'Install' y seguir el asistente: idioma, teclado, particionado (usar disco entero virtual), crear usuario y contraseña, configurar red (DHCP o estática según la red).

3) Instalar paquetes base y GRUB cuando lo solicite.

**Instalación de Guest Additions (opcional)**

1) En la ventana de la VM: Dispositivos → Insert Guest Additions CD image...

2) Dentro de Kali montar el CD y ejecuta: sudo sh /media/cdrom/VBoxLinuxAdditions.run (o ruta equivalente).

3) Reiniciar la VM para aplicar mejoras (resolución de pantalla, carpetas compartidas, portapapeles compartido).



**Configuraciones finales y seguridad**

1) Actualizar Kali: sudo apt update && sudo apt full-upgrade -y

2) Instalar herramientas que se necesite (muchas ya vienen por defecto instaladas): nmap, wireshark, metasploit-framework, openvas, etc.

3) Crear clones antes de ejecutar pruebas destructivas: Máquina → Clonar

4) Documentar las credenciales y accesos utilizados y borrar cualquier dato sensible que no deba quedar en la VM.


**Recomendaciones finales**

• Guardar las ISOs y configuraciones en un repositorio protegido.

• No compartir credenciales por canales inseguros.

• Usar clones antes de pruebas con impacto y eliminar datos sensibles de los discos virtuales cuando no sean necesarios.

Anexo 4. Guía de uso de Nmap

**Descripción**

Nmap es un escáner de red que permite identificar hosts activos, puertos abiertos, versiones de servicios y sistemas operativos. Es fundamental en la fase de reconocimiento y enumeración del pentesting.

**Instalación**

En Kali Linux viene preinstalado.

Para actualizarlo:

*sudo apt update*

*sudo apt install nmap -y*

**Uso básico**

Escaneo rápido de puertos abiertos

*nmap <IP>*

Detección de servicios y versiones

*nmap -sV <IP>*

Escaneo completo de puertos (1–65535)

*nmap -p- <IP>*

Modo agresivo (incluye OS detection y scripts)

*nmap -A <IP>*

Escaneo silencioso evitando ping

*nmap -Pn <IP>*

Anexo 5. Guía de uso de Netdiscover


**Descripción**

Netdiscover permite identificar dispositivos activos dentro de la red interna usando ARP scanning.

**Instalación**

En Kali Linux ya viene instalado.

*sudo apt install netdiscover -y*

**Uso básico**

Escaneo automático

*sudo netdiscover -r 192.168.1.0/24*

Resultado esperado

Lista de IPs activas

MAC address

Fabricante del dispositivo

Anexo 6. Guía de uso de OpenVAS

**Descripción**

OpenVAS es un escáner de vulnerabilidades completo basado en la plataforma Greenbone.

**Instalación**

*sudo apt install openvas -y*

*sudo gvm-setup*

*sudo gvm-check-setup*

**Uso básico**

Iniciar servicios:

*sudo gvm-start*

Acceder desde navegador:

*https://127.0.0.1:9392*

Escaneo recomendado

*Crear un nuevo Target* **->** *colocar IP objetivo.*

*Crear un Task* **->** *seleccionar Full and Fast.*

*Ejecutar el análisis.*

Anexo 7. Guía de uso de Nikto


**Descripción**

Nikto analiza servidores web en busca de configuraciones inseguras, archivos expuestos y vulnerabilidades conocidas.

**Instalación**

Viene instalado en Kali:

*sudo apt install nikto -y*

**Uso básico**

Escaneo HTTP

*nikto -h http://upec.edu.ec*

Escaneo HTTPS

*nikto -h https://upec.edu.ec*

Anexo 8. Guía de uso de Metasploit Framework


**Descripción**

Metasploit es un framework para explotación, post-explotación y automatización de ataques.

**Instalación**

En Kali ya viene incluido.

Actualizar módulos:

*sudo msfupdate*

**Uso básico**

Inicio

*Msfconsole*

Flujo básico de explotación

Buscar módulo:

*search <nombre>*

Seleccionar:

*use exploit/path*

Configurar parámetros:

*set RHOSTS <IP>*

*set RPORT <puerto>*

Añadir payload:

*set payload <payload>*

Ejecutar:

*Exploit*

Anexo 9. Guía de uso de Burp Suite

**Descripción**

Burp Suite es un proxy interceptador usado para pruebas en aplicaciones web.

**Instalación**

Viene en Kali.

**Uso básico**

Configuración inicial

*Abrir Burp Suite*.

*Crear un "Temporary Project"*.

*Configurar navegador → Proxy manual:*

*Host: 127.0.0.1*

*Puerto: 8080*

Herramientas clave

*Proxy: Intercepta peticiones.*

*Repeater: Reenvía solicitudes manualmente.*

*Intruder: Ataques automatizados.*

*Decoder: Decodificación de datos.*

Anexo 10. Guía de uso de Hydra

**Descripción**

Hydra realiza ataques de fuerza bruta contra servicios de autenticación.

**Instalación**

*sudo apt install hydra -y*

**Uso básico**

Ejemplo de fuerza bruta SSH

*hydra -l admin -P rockyou.txt ssh://192.168.1.10*

Ejemplo para HTTP-POST

*hydra -l admin -P rockyou.txt 192.168.1.10 http-post-form "/login.php:user=^USER^&pass=^PASS^:F=Incorrecto"*

Anexo 11. Guía de uso de Jhon the Ripper


**Descripción**

Herramienta para crackeo de hashes.

**Instalación**

Ya incluido en Kali

**Uso básico**

Descubrir algoritmo

john --list=formats

Crackear hash

Jhon hash.txt

Mostrar resultados

Jhon –show hash.txt