UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE TELECOMUNICACIONES

Tema:

SISTEMA DE DETECCIÓN DE INTRUSOS (IDS) EN REDES INFORMÁTICAS APLICANDO PROTOCOLOS DE CIBERSEGURIDAD

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Telecomunicaciones.

ÁREA: Programación y Redes.

LÍNEA DE INVESTIGACIÓN: Tecnología de la información y sistemas de control.

AUTOR: Kevin Jhonatan Vaca Tonato

TUTOR: Ing. Livio Danilo Miniguano Miniguano Mg.

Ambato – Ecuador

julio -2026

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA DE DETECCIÓN DE INTRUSOS (IDS) EN REDES INFORMÁTICAS APLICANDO PROTOCOLOS DE CIBERSEGURIDAD, desarrollado bajo la modalidad Proyecto de Investigación por el señor Kevin Jhonatan Vaca Tonato, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, julio 2026.

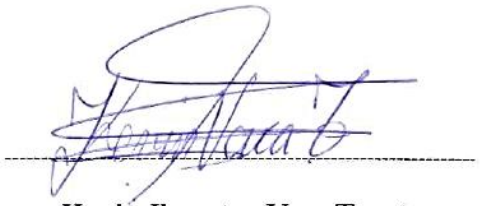
Validar sólo en FirmaEC.
Firmado Electrónicamente por:
1803209921 LIVIO DANILO MINIGUANO MINIGUANO

Ing. Livio Danilo Miniguano Miniguano Mg.
TUTOR

AUTORIA

El presente trabajo de titulación con el tema: SISTEMA DE DETECCIÓN DE INTRUSOS (IDS) EN REDES INFORMÁTICAS APLICANDO PROTOCOLOS DE CIBERSEGURIDAD, es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, julio de 2026



Kevin Jhonatan Vaca Tonato
C.C. 0504282427
AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga de este Trabajo de Integración Curricular o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normativas de la Institución. Además, cedo los Derechos de Autor de mi Proyecto de Investigación con fines de difusión pública, además apruebo la reproducción de este trabajo, dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga ganancia económica y se realice respetando mis derechos de autor de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, julio de 2026

Kevin Jhonatan Vaca Tonato
C.C. 0504282427
AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Kevin Jhonatan Vaca Tonato, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE DETECCIÓN DE INTRUSOS (IDS) EN REDES INFORMÁTICAS APLICADO PROTOCOLOS DE CIBERSEGURIDAD, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, julio de 2026

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
FRANKLIN OSWALDO MAYORGA MAYORGA

Ing. Franklin Oswaldo Mayorga Mayorga Mg.
PRESIDENTE DEL TRIBUNAL

Victor Santiago Manzano Villafuerte
Time Stamping Security Data

Ing. Víctor Santiago Manzano Villafuerte Mg.
PROFESOR CALIFICADOR

Jesus Israel Guaman Molina
Time Stamping Security Data

Ing. Jesús Israel Guamán Molina Mg.
PROFESOR CALIFICADOR

DEDICATORIA

El presente trabajo está dedicado de manera especial a mis padres Lourdes Tonato y Josep Martínez quienes, con su amor incondicional, paciencia y esfuerzo constante han sido el pilar fundamental a lo largo de mi vida y mi carrera universitaria siempre les quedaré muy agradecido por el apoyo moral y económico que me brindan desde la distancia, este logro es de ustedes y de mi persona.

A mi gran hermano Andrés, que desde muy pequeños hemos salido adelante junto a nuestra madre, con su apoyo y palabras de aliento para seguir adelante.

A mi hermano Mateo, mi pequeño que siempre estabas ahí para sacarme una risa con tus ocurrencias este logro es también por ti.

Con mucho amor y cariño, TAVAQUIN.

AGRADECIMIENTO

Primero quiero agradecer a Dios por brindarme la salud y la vida para poder culminar este logro.

A mi madre querida Lourdes, mamita linda gracias por haber hecho todo por mí y mis hermanos te quedo muy agradecido por tus enseñanzas y sacrificios que siempre han demostrado la grandeza de tu amor, Te Amo con todo mi ser.

A mi padrastro papa Josep, gracias por tu ayuda, tu amistad, la grandeza de tu corazón me ha inspirado a ser una mejor persona, el cariño que te tengo es inmenso, Te Amo y mil gracias por todo.

A mi hermano Mateo gracias por todo mi tavaquinsito por ser más que mi hermano mi amigo, Te Amo demasiado mi pequeñín. También a mi hermano Andrés, gracias por tu forma de ser conmigo y el apoyo que hemos tenido. Te quedo infinitamente agradecido, Te amo mi Andi Pandi.

A JuanSaVi, FERICKSR, Chisa, Pancho, Grupo de los Trolls, Toapanta, Ariel, Pilco, Ibarra, Ander, Darex, Don gato, a mi doctorcita Gaby y a mis demás amigos que han estado ahí y han hecho de este viaje el mejor de todos. Les quedo agradecido eternamente.

Con gratitud y mucho aprecio, TAVAQUIN

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN EJECUTIVO

Esta investigación se enfocó en el diseño, implementación y validación de un Sistema de Detección de Intrusos (IDS) basado en host utilizando programas de código abierto Wazuh, con el fin de fortalecer la ciberseguridad en un servidor aplicativo con Moodle configurado sobre Ubuntu. La metodología aplicada sustentó un enfoque cuantitativo y experimental en un entorno virtualizado controlado. Para garantizar la compatibilidad del sistema se instaló una versión previa a la operativa, se modificó el archivo ossec.conf y se configuraron los permisos para asegurar la visibilidad completa sobre los registros de auditoría y módulos de seguridad del kernel, como AppArmor. Las simulaciones de ataque se ejecutaron desde una máquina virtual Kali Linux. En el primer escenario, se evaluó la autenticación SSH mediante la herramienta Hydra; el sistema capturó los paquetes en el archivo auth.log., evidenciando los accesos fallidos y reflejando las alertas personalizadas en el panel principal. En el segundo escenario, se realizaron ataques de denegación de servicio (DoS) por inundación TCP. Las pruebas demostraron que los ataques masivos inhabilitaban al host antes de reportar al IDS. Para mitigar esto, se implementó una programación lógica que supervisaba directamente el estado de los sockets del kernel (netstat) y los errores del servidor web desde otra computadora virtual, logrando registrar y detectar el DoS a tiempo. Finalmente, se ejecutó una Respuesta Activa que aisló a la máquina atacante mediante un script personalizado encargado de apagar la interfaz de red, mitigando con éxito la denegación de servicio.

Palabras clave: redes informáticas, ciberseguridad, sistema de detección, Wazuh, simulación de ataques cibernéticos.

ABSTRACT

This research focused on the design, implementation, and validation of a Host-based Intrusion Detection System (HIDS) using the open-source software Wazuh, aiming to strengthen cybersecurity on an application server running Moodle configured over Ubuntu. The methodology followed a quantitative and experimental approach within a controlled virtualized environment. To ensure system compatibility, a pre-operational version was installed, the ossec.conf file was modified, and system permissions were configured to guarantee full visibility over audit logs and kernel security modules, such as AppArmor. Attack simulations were executed from a Kali Linux virtual machine. In the first scenario, SSH authentication was evaluated using the Hydra tool; the system captured network packets in the auth.log file, demonstrating failed access attempts and displaying customized alerts on the main dashboard. In the second scenario, Denial of Service (DoS) attacks were conducted via TCP flooding. Tests showed that massive attacks disabled the host before it could report to the IDS. To mitigate this, a logical programming script was implemented to directly monitor the state of kernel sockets (netstat) and web server errors from an alternative virtual machine, successfully registering and detecting the DoS on time. Finally, an Active Response was executed, which isolated the attacking machine through a customized script responsible for shutting down the network interface, successfully mitigating the denial of service.

Keywords: computer networks, cybersecurity, detection system, Wazuh, cyberattack simulation.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

"SISTEMA DE DETECCIÓN DE INTRUSOS (IDS) EN REDES INFORMATICAS APLICANDO PROTOCOLOS DE CIBERSEGURIDAD"

1.1.1 Planteamiento del problema

En Latinoamérica los ciber incidentes divulgados crecieron un 25% cada año durante la última década, según un relevamiento realizado por el Banco Mundial, y presentado en el libro Economía de la ciberseguridad para los mercados emergentes (2024). En el mundo, detalla el informe, el aumento de los ciber incidentes fue del 21%, en el lapso de 2014 a 2023, pero con una aceleración más rápida en la región de América Latina y Caribe [1].

Durante los últimos años se evidenció el alto crecimiento de redes de datos con diferentes servicios, el tiempo de comprometimiento promedio del crimen electrónico en el 2023 fue de 62 minutos, y el tiempo de comprometimiento más rápido registrado fue de 2 minutos y 7 segundos. En un ataque típico observado por CrowdStrike, más del 88% del tiempo de ataque se dedicó a obtener acceso inicial, y los adversarios están trabajando para reducirlo. Una vez dentro, un actor de ciber amenazas tardó solo 31 segundos en dejar una herramienta de descubrimiento inicial. Las intrusiones interactivas se están acelerando: la actividad de accesos interactivos para ejecutar comandos aumentó un 60% en el 2023 en comparación con el 2022. En la segunda mitad del 2023, el aumento se elevó al 73% en comparación con el mismo período del año anterior. Tres cuartas partes de los ataques con el objetivo de obtener acceso inicial estuvieron libres de malware, frente al 71% en el 2022, lo que subraya el cambio de los adversarios hacia técnicas más rápidas y efectivas [2].

En cuanto avanza las amenazas cibernéticas en América, México, y el Caribe, las organizaciones regionales tiene retos que evolucionan rápidamente. Entre el 2023 y el 2024, hubo un aumento del 15% en la cantidad de víctimas ubicadas en América Latina

nombradas en sitios web de filtración de datos para extorsión y ransomware, y casi un 38% de aumento en los anuncios de bróker de acceso [3].

El Panorama de Amenazas 2024 de Kaspersky revela que la compañía bloqueó 1,185,242 ataques de ransomware en América Latina entre junio de 2023 y julio de 2024, lo que representa un aumento del 2.8% en el último año. En América Latina hubo 1.185.242 ataques, es decir, 3.247 al día. Brasil, México, Ecuador y Colombia lideran la lista de los países más atacados de la región respectivamente. Una de las familias de ransomware más encontrada en la región fue "TrojanRansom.Win32.Blocker", un tipo de troyano que modifica la computadora de la víctima para que no pueda utilizar los datos o impide que el equipo funcione correctamente. Una vez bloqueados o cifrados los datos, el usuario recibe una petición de rescate, en la que se le solicita que envíe dinero para recuperarlos [4].

Otro hecho destacado en el panorama fue la interrupción del funcionamiento del grupo LockBit en Latinoamérica, que se vio afectado por agencias de seguridad internacionales: la Agencia Nacional contra el Crimen del Reino Unido, la Oficina Federal de Investigación (FBI) de EE.UU. y Europol, entre otras. El grupo era famoso por robar dinero a las empresas al acceder a sus datos confidenciales y amenazando con filtrarlos si las víctimas no pagaban el rescate [4].

Ecuador registró más de 12 millones de ciberataques en 2023. Aunque el número total de intentos de ataque disminuyó un 27% el año anterior, los ciberdelincuentes ahora adoptaron un enfoque que pretende maximizar las ganancias mediante extorsiones. Esta especialización de la ciberdelincuencia lleva a los expertos a estimar que para 2026, el 10% de las grandes empresas contará con un programa de confianza integral, maduro y medible de ciberseguridad, frente a menos del 1% que existe en la actualidad. La firma Gartner menciona que, hasta 2027, el 50% de los responsables de ciberseguridad en las empresas adoptará formalmente prácticas de diseño [5].

El crecimiento en el número y la dificultad de las violaciones de la red plantea un riesgo continuo y significativo para la protección de datos en empresas de cada escala. Estas infracciones pueden aparecer como acciones de red extrañas o dañinas, junto con tareas internas peligrosas y el mal uso de los sistemas internos, que afectan la seguridad, la privacidad y la accesibilidad de los recursos en línea. Hasta con sistemas de detección de intrusos estas amenazas no pueden ser reducidas en su totalidad.

Además, ni con la implementación de un IDS puede ser factible reconocer los falsos positivos que pueden emitir. La sensibilidad antes estas alertas debe ser preciso, para la optimización de actividades que se realicen en los diferentes equipos atacados. Es un desafío muy grande el verificar alertas legítimas las cuales se mezclan con el ruido del tráfico de red siendo los principales problemas para realizar respuestas rápidas y efectivas. Este escenario enfatiza la importancia de investigar y crear métodos que mejoren la precisión de la detección, minimicen los rechazos incorrectos y permitan un mejor reconocimiento de las irregularidades de seguridad en entornos de red intrincados, con el objetivo de reforzar la ciberseguridad de un sistema ya sea en organizaciones o en redes personales.

1.2 Antecedentes investigativos

La vasta investigación en ciberseguridad ha explorado múltiples enfoques para la detección de intrusiones en redes. A pesar de los avances, persisten desafíos como la alta tasa de falsos positivos/negativos y la adaptación a nuevas amenazas. Este estudio se basa en dichos antecedentes para identificar brechas y proponer una contribución relevante al estado del arte.

En el año 2023, Rudibel Perdigón Llanes en [6] el objetivo de esta investigación consiste en analizar la pertinencia y aplicabilidad de Suricata como sistema de detección de intrusiones para fortalecer la seguridad en las redes digitales de las pequeñas y medianas empresas. Se desarrolló una investigación de tipo descriptiva donde se emplearon como métodos científicos el analítico sintético y el experimental. Se evaluó el consumo de recursos de hardware de Suricata y su capacidad para la detección de intrusiones mediante el análisis basado en firmas en la red de una mediana empresa agroindustrial cubana. Se generó tráfico de red malicioso mediante la distribución Kali Linux para simular ataques de sondeo de redes, denegación de servicios y de fuerza bruta. Los resultados obtenidos evidenciaron que Suricata posee una buena efectividad para la detección de intrusiones mediante el análisis basado en firmas con un consumo eficiente de recursos de hardware. El uso de Suricata en las pequeñas y medianas empresas contribuirá a asegurar la confidencialidad, disponibilidad e integridad de sus recursos digitales.

En 2025, en [7] la sofisticación de las ciberamenazas exige herramientas más eficientes e inteligentes para apoyar a los Centros de Operaciones de Seguridad (SOC) en la gestión y mitigación de incidentes, razón por la cual se desarrolló el Security Event Response Copilot (SERC), un sistema diseñado para asistir a los analistas en la respuesta y mitigación de brechas de seguridad de manera más efectiva. Este sistema integra dos componentes fundamentales: la extracción de datos de eventos de seguridad mediante métodos de Generación Aumentada por Recuperación (RAG) y la orientación para la respuesta ante incidentes basada en modelos de lenguaje de gran tamaño (LLM). El estudio utiliza específicamente Wazuh, una plataforma de código abierto de Gestión de Eventos e Información de Seguridad (SIEM), como base para capturar, analizar y correlacionar eventos de seguridad desde los puntos finales (endpoints), aprovechando sus capacidades de recopilación de datos en tiempo real; además, aplica un enfoque RAG para recuperar perspectivas contextuales específicas a partir de tres colecciones de datos vectorizados que incluyen conocimientos de respuesta ante incidentes, el marco MITRE ATT&CK y el Marco de Ciberseguridad de NIST (CSF) 2.0. Esta integración logra cerrar la brecha entre la gestión estratégica de riesgos y la inteligencia táctica, permitiendo identificar con precisión las tácticas y técnicas de los adversarios bajo las mejores prácticas de ciberseguridad, demostrando así el potencial de combinar marcos estructurados de inteligencia de amenazas con modelos impulsados por IA y las robustas capacidades de Wazuh para enfrentar los desafíos dinámicos del complejo entorno digital actual.

Las diferentes técnicas de investigación en [8] acerca de ciberseguridad denotan la importancia de esta, siendo la idea central de este estudio se centró en la revisión de la literatura existente concerniente a los procesos de prevención de ciberataques, con una orientación específica hacia las infraestructuras tecnológicas. Metodológicamente, se empleó un procedimiento de revisión bibliográfica que adhirió a una guía de ocho pasos, inspirada en la declaración PRISMA (Preferred Reporting Items for Systematic reviews and Meta-Anayses).

Para la recolección de datos se empelo varios sitios web de publicación i divulagacion científica como Scopus, Scielo entre otros. Para la organización y selección rigurosa de los artículos, se utilizó la herramienta tecnológica Harzing's Publish or Perish. Los resultados del estudio ayudaron a realizar buenas prácticas de tecnología que fueron

dirigidas a detener amenazas más comunes, así también como acciones que ayuden a la prevención de estas. Dentro de estos hallazgos, se incluyen estrategias para la prevención de ransomware mediante respaldos frecuentes de información y el uso de aplicaciones anti-ransomware , y para la protección de dispositivos IoT, se sugieren medidas como autenticación robusta, control de acceso físico, descargas seguras, detección de malware con antivirus como Kaspersky o Norton, y la aplicación de actualizaciones de software [8].

En 2025, según. Josué Guillermo en [9] se investigó la implementación de un Sistema de Detección de Intrusos (IDS) con el propósito de fortalecer la seguridad informática en la empresa Ambacar. La metodología aplicada se desarrolló en tres etapas fundamentales. La primera etapa abarcó acuerdos legales, una fase de reconocimiento tipo grey box y un análisis de amenazas. La segunda etapa se centró en la implementación y configuración del IDS con Suricata. Finalmente, la tercera etapa consistió en la validación del IDS a través de simulaciones de ataques, incluyendo Slowloris, EternalBlue, y ataques de denegación de servicio (DoS). Además, se optimizó la eficiencia en la detección de amenazas reales al implementar umbrales (thresholds) para filtrar eventos y priorizar alertas críticas, tras identificar una alta generación de eventos por tráfico legítimo.

1.3 Fundamentación Teórica

1.3.1 Detección de intrusos (IDS)

El sistema de detección de intrusos es una aplicación o programa que identifica el tráfico malicioso de red y busca amenazas conocidas. El IDS puede enviar alertas a diferentes equipos según su configuración. La mayoría de los sistemas de detección de intrusos solo monitorean el tráfico y lo clasifican, dependiendo de ello se verifica si es malicioso o hay sospechas de tráfico inusual. Sin embargo, otros sistemas pueden tomar medidas configuradas por el usuario cuando se detecta actividades maliciosas o sospechosas, como bloquear las interfaces, detener el tráfico entre otras [10].

1.3.2 Tipos de IDS (Detección de intrusos)

- **Sistema de detección de intrusiones en la red (NIDS):** Los sistemas de detección de intrusiones en la red se ubican de forma critica en los nodos de cualquier topología e infraestructura de red con el objetivo de evaluar todo el flujo de datos ya sea de ida como de vuelta.[11].
- **Sistema host de detección de intrusiones (HIDS):** Los sistemas de detección de intrusiones basados en host se implementan de forma local en cada endpoints que este conectada la infraestructura de la empresa. Esta tiene la capacidad de interceptar el tráfico interno y las amenazas específicas del sistema operativo que un NIDS no lo podría detectar. El cual facilita la identificación de software malicioso que pretenda penetrar en toda la red de forma lateral [11].
- **Sistema de detección de intrusiones basado en firmas (SIDS):** Monitorea todos los paquetes y genera alertas automáticas únicamente cuando detecta una coincidencia exacta con otras firmas digitales registradas de amenazas [11].
- **Sistema de detección de intrusiones (AIDS) basado en anomalías:** Este sistema examina el flujo de datos y lo compara con un umbral ya preestablecido como legitimo o estándar. El objetivo de este sistema es identificar variaciones que estén fuera de lo establecido en toda la red o infraestructura el cual supervisa parámetros que sean críticos como el uso indebido del ancho de banda típico, el estado de todos los puertos, la activación de nodos y los protocolos de comunicaciones de la infraestructura [11].
- **Sistema de detección de intrusiones perimetrales (PIDS):** Este sistema se coloca dentro de la red para la detección de intrusión que sea en la red o la infraestructura critica de las diferentes organizaciones [11].
- **Sistema virtual de detección de intrusiones basado en máquinas (VMIDS):** Este sistema virtual de detección identifica amenazas de forma que las máquinas sean revisadas de forma continua por medio de programas virtuales. [11].
- **Sistema de detección de intrusiones basado en pila (SBIDS):** El sistema de detección de intrusos basado en pila se acoplan directamente a los protocolos TCP/IP de la infraestructura de una red corporativa. Inspecciona los flujos de

los datos además que logra la intercepción y filtración de los diferentes paquetes o tramas antes de que el sistema operativo final o anfitrión inicien su procesamiento lógico de red [11].

1.3.3 Intrusión en Ciberseguridad

La intrusión en ciberseguridad es cuando un atacante malintencionado accede a una red, sistema, dispositivo o aplicación sin autorización para obtener información confidencial o realizar actividades maliciosas. Las intrusiones pueden ocurrir de varias formas, como a través de programa maligno, phishing, hacking o ingeniería social adicionalmente pueden tener graves consecuencias para las empresas y organizaciones, como la pérdida de datos valiosos, daño a la reputación, pérdida económica y sanciones legales. Por lo que, la importancia de que las empresas deban implementar formas de seguridad para la prevención, detección y así responder a las diferentes formas de intrusión [12].

1.3.4 Tipos de medidas de seguridad para la prevención de ciberataques

- **Crear contraseñas robustas y únicas:** Dentro de la cotidianidad del mundo digital se ve inmerso en contraseñas ya sea sencillas para ciertas actividades y otras tan robustas que se crean con algoritmos aleatorios además de llaves físicas, es por ello por lo que se sugiere siempre la creación y respaldo de todas las contraseñas que se vayan a crear para diferentes plataformas además de que cada contraseña no se repita en varias instancias para que no sean vulneradas en otros sitios [13].
- **Actualización de software:** Para mayor seguridad hay que mantener actualizado el sistema operativo en todos los dispositivos electrónicos, ya que siempre se realiza diferentes actualizaciones y parches de seguridad. Todos los desarrolladores de sistemas operativos están al pendiente para la realización de actualizaciones mayormente centradas en la optimización y seguridad del entorno operativo [13].
- **Evitar correos maliciosos o sospechosos:** Los ataques de phishing empiezan por correos sospechosos que parecen legítimos ante el usuario final, es por eso

que se recomienda siempre verificar el URL que siempre anteponga la seguridad del sitio, además que ninguna empresa o entidad seria solicita información sensible por correo electrónico [13].

1.3.5 Factores que Influyen en el Tiempo de Respuesta ante ciberataques

- **Preparación:** Creación de políticas de seguridad, definir quien responde ante un ataque además de configurar e instalar las herramientas necesarias para su control.
- **Detección y análisis:** Las diferentes amenazas hay que categorizarlas según el nivel de amenaza analizando indicadores de compromiso y determinando el alcance del daño potencial [14].
- **Técnicas de mitigación rápida:** Un aspecto crítico del manejo de incidentes de ciberseguridad es la rápida aplicación de técnicas de mitigación, que pueden minimizar significativamente el impacto y una mayor propagación de la amenaza. Las técnicas de mitigación rápida son esenciales para que los equipos de respuesta a incidentes aborden y contengan las amenazas cibernéticas de manera efectiva [14].
- **Actividad luego del incidente de ciberseguridad:** Una vez resuelto el problema de ciberataque realizar un análisis de que fallo y como mejorarlo. Se documenta todo el proceso desde el ataque hasta la resolución de este como se muestra en la Figura 1.

Figura 1. Planificación de Respuesta a Incidentes

1.3.6 Redes Informáticas

Como se define en [15], una red informática es un sistema de dispositivos informáticos interconectados tomando en cuenta que van desde entornos tradicionales hasta entornos basados en la nube) que se comunican y comparten recursos entre sí.

La conexión en red, o interconexión informática, implica la conexión de dos o más dispositivos informáticos (por ejemplo, ordenadores de sobremesa, portátiles, dispositivos móviles, enrutadores, aplicaciones) para permitir la transmisión y el intercambio de información y recursos [15].

Los dispositivos en red se basan en protocolos de comunicación (reglas que describen cómo transmitir o intercambiar datos a través de una red), lo que les permite compartir información a través de conexiones físicas o inalámbricas [15].

Las redes informáticas forman la columna vertebral de casi todas las experiencias digitales, desde la comunicación personal y el entretenimiento hasta las operaciones empresariales nativas de la nube y la infraestructura global. Diseñadas para la escalabilidad, la velocidad y la seguridad de TI, las redes actuales admiten flujos de datos dinámicos tanto en sistemas en las instalaciones como en entornos de nube virtualizados como se muestra en la Figura 2 [15].

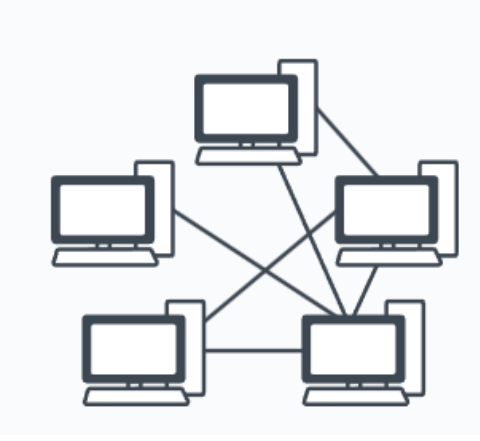


Figura 2. Diagrama de una red informática [16]

1.3.7 Tipos de redes informáticas

Hay varias arquitecturas que se puede aplicar a las redes informáticas, por lo que existen varios tipos de clasificación que dependen del alcance y de la forma en las que están conectadas.

1.3.8 Clasificación de las redes según su Alcance Geográfico

De acuerdo con la clasificación establecida en [17] las redes informáticas se dividen en diferentes alcances geográficos detallados a continuación:

- PAN (Red de Área Personal): Comprende el entorno más cercano del usuario. Se utiliza para la interconexión de periféricos personales, siendo el estándar Bluetooth el protocolo predominante para enlazar dispositivos como terminales móviles y accesorios de audio como se muestra en la Figura 3.

Figura 3. Red de Área Personal [18]

- LAN (Red de Área Local): Las redes de área local forma la columna principal de la conectividad de los hogares y oficinas. Este tipo de diseño es caracterizado por ser implementado dentro de un espacio restringido o cerrado, como una oficina de algún servicio o un edificio residencial, el cual ayuda a la velocidad y además de la seguridad al momento de implementar servicios, documentos, archivos de forma interna como se muestra en la Figura 4.

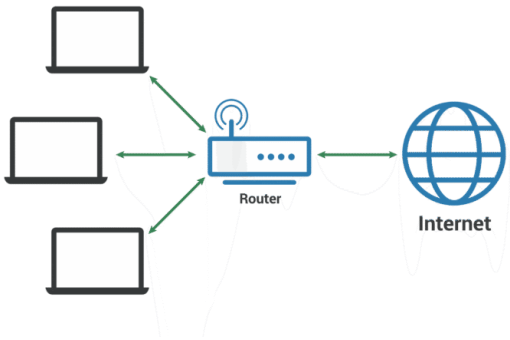


Figura 4. Red de Area Local [19]

- WLAN (Red de Área Local Inalámbrica): Constituye la variante sin cables de la LAN (Wi-Fi). Su principal ventaja es que permite la movilidad de los nodos dentro del área local sin depender de una infraestructura de cableado estructurado como se muestra en la Figura 5.

Figura 5. Red WLAN[20]

- CAN (Red de Área de Campus): Se despliega para unificar diversas LANs dentro de una propiedad extensa. Es la configuración típica en centros de salud, plantas industriales o campus universitarios donde se requiere una administración centralizada como se muestra en la Figura 6.

Figura 6. Red de Área de Campus [21]

- MAN (Red de Área Metropolitana): Esta hecha para la conexión de áreas urbanas o municipales de alta velocidad. Es la unión de varias LAN que estas en diferentes sitios dentro de una misma ciudad como se muestra en la Figura 7.

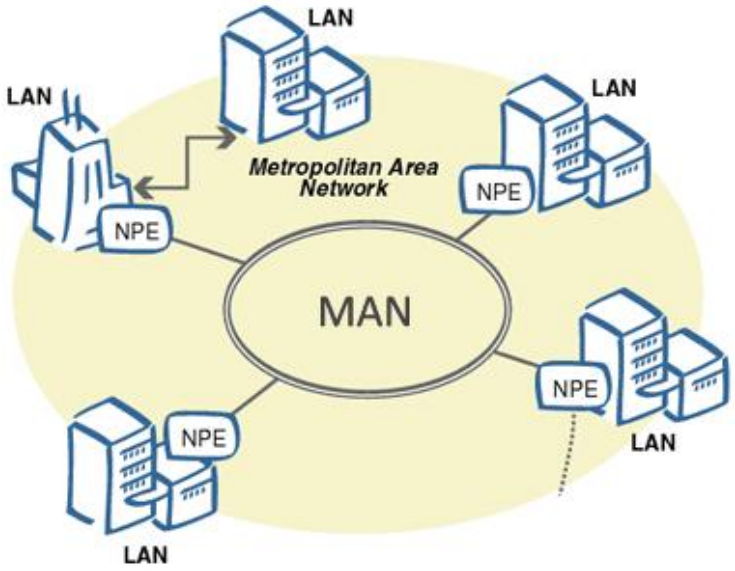


Figura 7. Red de Area Metropolitana[22]

- WAN (Red de Área Amplia): Rompe las barreras geográficas mediante el uso de tecnologías de largo alcance como satélites o tendidos de fibra óptica transcontinentales. La red global Internet es el máximo exponente de esta categoría como se muestra en la Figura 8.

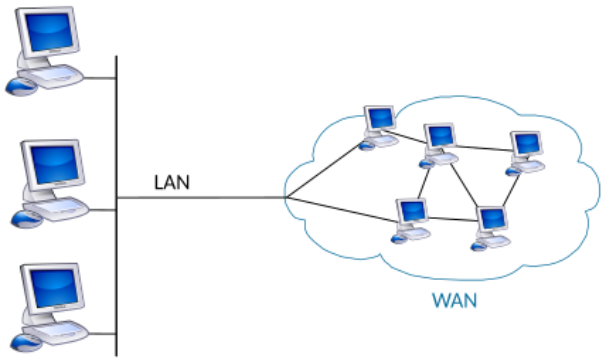


Figura 8. Red de Area Amplia [23]

- GAN (Red de Área Global): Es la capa superior de conectividad que permite la interoperabilidad de las comunicaciones móviles a nivel mundial, garantizando que un dispositivo mantenga el servicio sin importar su ubicación en el planeta como se muestra en la Figura 9.

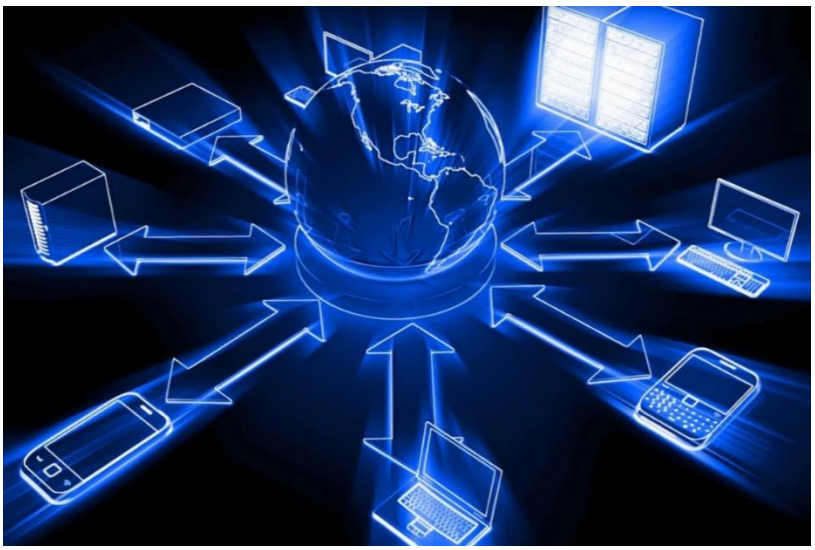


Figura 9. Red GAN[24]

1.3.9 Profundización en Ciberseguridad y Amenazas

La idea fundamental detrás de una táctica de defensa multilínea es que un único producto de seguridad no es suficiente para salvaguardar completamente una red de todos los tipos de ataques posibles. No obstante, la adopción de diversas herramientas y procedimientos de seguridad puede ser útil para identificar e impedir los ataques a

medida que ocurren, permitiendo a las empresas gestionar eficazmente una variedad de riesgos. Este método se vuelve más crucial a medida que las entidades amplían sus redes, sistemas y número de usuarios. [25].

1.3.10 Tríada de la Seguridad de la Información

Las tres letras que componen la triada se refieren a la confidencialidad, la integridad y la disponibilidad. La tríada CID es un modelo frecuentemente utilizado que sirve como fundamento para crear sistemas de seguridad. Se emplean para detectar debilidades y enfoques para desarrollar soluciones. El CID de los datos son esenciales para el funcionamiento de una empresa, y la tríada CIA descompone estas tres nociones en aspectos individuales. Esta separación es beneficiosa porque orienta a los equipos de seguridad en el reconocimiento de diversas maneras de abordar cada preocupación. En condiciones ideales, cuando se satisfacen los tres criterios, el perfil de seguridad de la entidad se fortalece y está más preparado para afrontar incidentes de amenazas [26].

Figura 10. CID de la Seguridad de la Información [27]

- **Triada CID: Confidencialidad**

  La confidencialidad constituye el pilar fundamental de la tríada CIA (Confidencialidad, Integridad y Disponibilidad) y resulta imperativa para la salvaguarda de activos de información críticos. En este ámbito, se define como

el conjunto de mecanismos y protocolos diseñados para garantizar que el acceso a la información esté restringido exclusivamente a entidades o usuarios debidamente autorizados[28]. Este concepto, intrínsecamente ligado a la privacidad y el secreto institucional, es vital en sectores que gestionan datos sensibles, tales como registros clínicos, estados financieros e información de identificación personal (PII), donde una brecha de seguridad podría derivar en repercusiones legales y operativas de gravedad [28].

Figura 11. Confidencialidad de la información.[29]

- **Triada CID: Integridad**

La integridad de la información representa uno de los ejes rectores de la seguridad informática, cuyo objetivo primordial es salvaguardar los datos frente a cualquier proceso de alteración o destrucción no autorizada. Este principio garantiza que la información permanezca precisa, coherente y fidedigna a lo largo de todas las fases de su ciclo de vida, desde su generación y procesamiento hasta su eliminación final. Por consiguiente, se asegura que el estado de los datos sea, en todo momento, el estado original previsto o una transición validada como legítima[30].

Más allá de la protección contra vectores de ataque externos, como el software malicioso o las intrusiones persistentes, la integridad abarca la mitigación de riesgos derivados de errores humanos y fallos sistémicos que puedan comprometer la veracidad de la información. Una degradación en la integridad

de los sistemas puede conducir a una toma de decisiones errónea, la pérdida de credibilidad institucional y posibles responsabilidades legales de carácter crítico. En el ámbito corporativo, la preservación de este pilar es indispensable para fomentar la confianza entre los diversos actores interesados (stakeholders), adquiriendo una relevancia máxima en sectores como el financiero y el sanitario, donde la exactitud de los datos tiene una incidencia directa en la operatividad y la integridad humana [30].

- **Disaster Recovery Plan (DRP) en Integridad de la información.**

  En [30] el DRP es un plan detallado con una estructura diseñada para guiar a las diferentes instituciones ante colapsos en su infraestructura critica como fallos de hardware o ciberataques dirigidos a zonas críticas.

  Un DRP integral debe tener los siguientes componentes esenciales:

  - Fase de evaluación de riesgos: Determinar todos los tipos de ocurrencias y el origen de todas las amenazas que puede interrumpir cualquier servicio de naturaleza tecnológica enfocada al área de tecnología de la información.
  - Detallar el Impacto Operativo y Financiero: Analizar cada aspecto en el cual afectaría ya sea de forma operativa o económica la caída de algún servicio prestado por la institución.
  - Fase de Restauración: Precisar como se puede restaurar los servicios a su estado operativo normal luego del incidente.
  - Protocolo De Comunicación Y Emergencias: Poseer una lista de contactos a los cuales se debe llamar o notificar de forma inmediata durante y luego de los incidentes.
  - Pruebas y Actualizaciones Regulares: Ejecutar pruebas de forma recurrente o periódicas sobre el plan para la corrección de vulnerabilidades presentes.

Figura 12. Plan de recuperación ante desastres[31]

- **Relación entre DRP e integridad de la información**

  La integridad de la información mantiene una correlación directa con la Planificación de Recuperación ante Desastres (DRP, por sus siglas en inglés), dado que la alteración o corrupción de los datos críticos puede ser catalogada como un evento catastrófico para la organización. Un DRP ejecutado de manera eficiente debe garantizar la preservación de la veracidad y consistencia de los activos digitales, tanto durante el incidente como en las fases posteriores de restauración. [30].

- **Triada CID: Disponibilidad**

  Dentro del dominio de la seguridad de la información, la disponibilidad se define como la capacidad de garantizar la fiabilidad y el acceso oportuno a los activos de datos y a la infraestructura tecnológica subyacente por parte de los usuarios autorizados. La disponibilidad debe asegurar que todos los servicios de índole informático dentro de una entidad se deberá mantener siempre operativa además de ser eficiente [32].

  Para que esto sea factible las personas idean siempre planes de mejora, siendo ejecutados para así asegurar la continuidad del negocio o servicio siendo de alta disponibilidad con ello mitigar los diferentes puntos de fallo que puedan existir. Entre las medidas técnicas fundamentales para el aseguramiento de este componente destacan la implementación de esquemas de redundancia, tales como sistemas RAID, configuraciones en clúster y balanceo de carga entre nodos. De igual forma, se requiere la gestión de copias de seguridad de datos

y configuraciones, la redundancia en el suministro eléctrico y enlaces de datos, así como la utilización de centros de datos alternos o co-ubicados. Estas metodologías garantizan que tanto la información como los sistemas que la procesan permanezcan accesibles ante contingencias técnicas o desastres operativos [32].

1.3.11 El Ecosistema del Software Libre en Ciberseguridad

En [33] define el software de código abierto como la distribución bajo una licencia particular que legalmente permite a los usuarios acceder a un código fuente. Existen diversas licencias de este tipo, pero comúnmente se define como open source si cumple estas características:

- Se ofrece como código fuente sin ningún coste adicional, lo que implica que los usuarios tienen la posibilidad de ver el código del software y realizar modificaciones a su gusto [33].
- El código fuente puede ser reaprovechado en nuevas aplicaciones, por lo que cualquiera tiene la libertad de utilizar el código para crear su propio software y compartirlo [33].

La adopción de soluciones basadas en software de código abierto (OSS) constituye una ventaja estratégica para la infraestructura tecnológica de las organizaciones, especialmente para aquellas con equipos de desarrollo limitados. Desde la perspectiva de la gestión de la ciberseguridad, la naturaleza abierta del código facilita un ciclo de actualización y parcheo más dinámico frente a exploit emergentes, superando la rigidez operativa característica del software propietario o de código cerrado [34].

En la Tabla 1 se muestra las comparativas de un software open source ante un software cerrado o de propietario:

Tabla 1. Contraste operativo y de seguridad: código abierto vs. Propietario [34]

Característica	Software de Código Abierto (Open Source)	Software Propietario (Código Cerrado)
Visibilidad del código	Pública: Cualquier usuario puede revisar el código fuente en busca de errores.	Privada: Solo los desarrolladores de la empresa dueña tienen acceso al código.
Corrección de errores	Colaborativa: La comunidad o el equipo interno pueden crear "parches" rápidamente.	Centralizada: Hay que esperar a que la empresa lance una actualización oficial.
Costo de adquisición	Gratuito, eliminando el pago de licencias por uso.	Pago de licencias mensuales o anuales hechas por el usuario.
Seguridad	Transparencia de código, mejoras de software continuas y configurables .	Seguridad muy estricta.
Mantenimiento	Sencilla, se puede realizar con la ayuda de la comunidad.	La dependencia del propietario hace que solo sea disponible por personal autorizado.
Independencia	Evita la dependencia de un solo proveedor (Vendor Lock-in).	Existe una dependencia total del fabricante para soporte y mejoras.

La selección de un modelo que se basa en software de código abierto para la implementación del Sistema de Detección de Intrusos (IDS)En contraste con las soluciones propietarias, caracterizadas por su arquitectura cerrada que limita el acceso a la visualización del flujo de datos, así como por una dependencia de los ciclos de actualización impuesta por el fabricante, el software de código abierto permite el acceso total al código fuente, habilitando una inspección a fondo de este. Esta posibilidad se convierte en una necesidad real dentro del despliegue de los protocolos de ciberseguridad, al facilitar la búsqueda de vulnerabilidades a través de una revisión por pares de gran amplitud y la implementación de parches que, con independencia de las restricciones de licenciamiento, logra introducir una versión personal y optimizada de la de la solución de gestión de a seguridad. La eficiencia de este encuadre se concentra en la Tabla I, a partir de la cual las particularidades operativas de ambos modelos de desarrollo son capturadas. Como puede ser apreciado, el uso de herramientas abiertas no sólo mejora la estructura de costes de un proyecto, sino que elimina el vendor lock-in con el que el IDS cambia para cubrir rápidamente cualquier vulnerabilidad que cada exploit reportado en la comunidad global de seguridad le

permita facturar. En este contexto, usando plataformas como Wazuh se logra una posición de defensa fuerte, y se vale de las condiciones ofrecidas por la transparencia del código para convertir a la organización en una de defensa de los activos informáticos tanto de integridad como de privacidad.

1.3.12 Análisis Comparativo de Soluciones IDS Open Source

1.3.13 SNORT

Snort se define como una herramienta de código abierto de alto rendimiento, diseñada para operar tanto como un Sistema de Detección de Intrusos (IDS) como un Sistema de Prevención de Intrusos (IPS). Su funcionalidad principal radica en la capacidad de ejecutar un análisis de tráfico en tiempo real y el registro pormenorizado de paquetes en redes basadas en IP. Para su operación, Snort emplea un motor de inspección fundamentado en un lenguaje de reglas flexible, el cual integra métodos de detección por firmas, análisis de protocolos y detección de anomalías para identificar vectores de ataque y actividades maliciosas [35].

Figura 13. SNORT[36]

a. Mecanismos de Detección y Gestión de Amenazas

La arquitectura de Snort permite a los administradores de red identificar una amplia gama de incidentes de seguridad, incluyendo ataques de Denegación de Servicio (DoS), Denegación de Servicio Distribuido (DDoS), ataques dirigidos a interfaces CGI, desbordamientos de búfer (buffer overflows) y escaneos de puertos sigilosos

(stealth port scanning). El sistema opera mediante la definición de conjuntos de reglas que caracterizan el tráfico malintencionado; al producirse una coincidencia entre el tráfico capturado y las firmas predefinidas, el motor genera alertas automáticas para la gestión del incidente[35].

b. Implementación y Versatilidad Operativa

Al ser una solución de código abierto y libre distribución, Snort se posiciona como una alternativa accesible y robusta tanto para entornos individuales como corporativos. Su versatilidad permite su implementación en diversas capas de la arquitectura de red: puede actuar como un sensor de detección pasivo para el monitoreo de paquetes o como una solución IPS integral con capacidad de respuesta activa, bloqueando de manera autónoma los paquetes identificados como amenazas. En este contexto, el lenguaje de reglas de Snort se constituye como el núcleo del sistema, determinando con precisión los criterios de recolección de datos y los protocolos de respuesta ante actividades sospechosas[35].

En la Tabla 2 se muestra las ventajas y desventajas técnicas de este software open source

Tabla 2. Análisis técnico de Snort respecto al objetivo del sistema IDS [35]

Categoría	Ventajas Técnicas	Desventajas / Limitaciones
Arquitectura de Procesamiento	Madurez del motor: Su arquitectura monohilo es sumamente estable y consume pocos recursos en redes de baja carga.	Limitación de rendimiento: Al no ser multihilo de forma nativa (en versiones base), puede presentar pérdida de paquetes (packet drop) en redes de alta velocidad.
Gestión de Reglas	Sextandarización: El lenguaje de reglas de Snort es el estándar de la industria, lo que facilita la creación de firmas personalizadas para un IDS.	Complejidad en la sintaxis: La estructura de código es compleja y muy estricta haciendo que la curva de aprendizaje sea muy elevada.
Detección de Amenazas	Flexibilidad de análisis: Excelente Procesamiento de protocolos eficientes, lo cual facilita el reconocimientos de amenazas	Limitación por cifrado: Trafico encriptado por protocolos TLS y SSL el cual se necesita más herramientas para su análisis.
Integración con el Sistema	Compatible con Linux: Al ser compatible con Linux también es	Interfaz Nativa: Carece de un entorno gráfico (GUI) robusto, dependiendo

	compatible con Kali Linux, siendo las pruebas de penetración mas sencillas y rápidas.	de herramientas externas (como Barnyard2 o SIEMs) para la visualización de alertas.
Ciberseguridad Proactiva	Modo IPS eficiente: Permite la transición rápida de detección (IDS) a prevención (IPS) mediante la integración con ip-tables.	Consumo de CPU: En análisis de inspección profunda de paquetes (DPI), el uso de un solo núcleo puede saturar el procesador del servidor IDS.

Como se evidencia en la Tabla 2, la selección de Snort como componente del sistema IDS se justifica por su estabilidad y su lenguaje de reglas estandarizado, lo cual facilita la aplicación de protocolos de ciberseguridad específicos. No obstante, es imperativo considerar su naturaleza monohilo al dimensionar el hardware de red, asegurando que el procesamiento de paquetes no se convierta en un cuello de botella para la disponibilidad de la infraestructura.

1.3.14 SURICATA

Desarrollado por la Open Information Security Foundation (OISF) en 2009, Suricata se presenta como un motor de código abierto especializado en la detección de amenazas de red. Su arquitectura ha sido diseñada para operar de manera robusta como un sistema IDS/IPS y de Monitoreo de Seguridad de Red (NSM). A diferencia de otras soluciones, Suricata capitaliza las capacidades del hardware moderno mediante el uso de procesamiento multinúcleo y aceleración por GPU, posicionándose como una solución escalable para la mitigación de ataques complejos[37] .

Figura *14*. Suricata IDS[38]

a. Capacidades Arquitectónicas y Funcionalidades Clave

La superioridad técnica de Suricata se fundamenta en los siguientes pilares operativos:

- Multihilo (Multi-threading): A diferencia de arquitecturas monohilo, Suricata distribuye la carga de trabajo entre múltiples núcleos de procesamiento. Esto permite la gestión de volúmenes de tráfico masivos y la ejecución de análisis heurísticos complejos con una latencia mínima[37].
- Identificación y Análisis de Protocolos: El motor posee una capacidad avanzada para la decodificación automática de protocolos críticos (HTTP, TLS, FTP, SMB). Esto facilita una inspección profunda de paquetes (Deep Packet Inspection) y una detección precisa de ataques específicos a nivel de aplicación[37].
- Extracción de Archivos: Suricata permite la captura y almacenamiento de archivos en tránsito para su posterior análisis forense. Siendo esta funcionalidad una de las más esenciales para el análisis de archivos corruptos o maliciosos con el fin de extraer información sensible y delicada [37].
- Interoperabilidad y Ecosistema SIEM: Este sistema es versátil al momento de integrarlo con otras herramientas de gestión de seguridad para la evasión de intrusos [37].
- Compatibilidad y Granularidad de Reglas: El código es compatible con las diferentes instancias de Snort lo que facilita el cambio y la escalabilidad tecnológica. No obstante, extiende estas capacidades mediante palabras clave avanzadas que permiten una detección más granular y específica [37].

b. Modos de Despliegue y Aplicaciones Prácticas

En [37] se detalla que Suricata ofrece una versatilidad de configuración adaptable a diversos escenarios de red:

- Modo NIDS: Monitoreo pasivo del tráfico y generación de alertas basadas en firmas.
- Modo IPS: Intervención activa sobre las amenazas mediante el descarte o rechazo de paquetes maliciosos.
- Modo NSM: Registro pormenorizado de metadatos de red para auditoría.

Gracias a esta flexibilidad, Suricata es implementado en entornos de alta demanda para la detección de amenazas en tiempo real, la caza de amenazas (threat hunting) y el análisis forense post-incidente, proporcionando una visibilidad integral de la infraestructura de red [37].

En la Tabla 3 se realiza un análisis dentro del enfoque técnico y académico con respecto a comparativas para la elección de un software open source que se adecue a un IDS:

Tabla 3.Análisis técnico de Suricata: Características, Ventajas y Limitaciones[37]

Categoría	Ventajas Técnicas	Desventajas / Limitaciones
Arquitectura de Procesamiento	Nativo Multihilo (Multi-threading): Permite procesar grandes volúmenes de tráfico distribuyendo la carga en todos los núcleos del CPU.	Mayor consumo de memoria: Al gestionar múltiples hilos y estados de flujo, requiere más memoria RAM que un motor monohilo.
Rendimiento de Hardware	Aceleración por GPU: Capacidad de delegar el escaneo de firmas a la tarjeta gráfica mediante CUDA o OpenCL para maximizar el throughput.	Optimización Precisa: La carga de tareas entre el hilo y el hardware debe ser preciso y demanda una administración avanzada.
Inspección de Tráfico	Detección de tráfico: Identifica los protocolos de comunicación y neutraliza ataques silenciosos.	Reglas sensibles: Una configuración errónea puede afectar la sensibilidad de falsos positivos.
Análisis Forense	Archivos de forma extraíble: Facilita la extracción y almacenamiento para su respectivo análisis forense.	Almacenamiento: El alto consumo de recursos al monitoreo y extracción masivo de archivos deja sin almacenamiento.
Ecosistema de Datos	Operabilidad con otros sistemas: Por su formato tipo JSON se puede operar con otros sistemas como Wazuh o Elastic Stack.	Depende de un SIEM en el que pueda centralizar todos los logs para su funcionamiento al máximo ya que carece de un entorno gráfico.

La inclusión de Suricata en la estructura del sistema IDS se basa en su diseño de múltiples hilos, lo que asegura que el sistema pueda escalar al aumentar el flujo de tráfico de red. Como se muestra en la Tabla 3, su habilidad para realizar una revisión exhaustiva de los protocolos y la recuperación de archivos ofrece una importante capa de observación para identificar amenazas sofisticadas y llevar a cabo análisis forenses,

coincidiendo con los protocolos de ciberseguridad preventiva sugeridos en este estudio.

1.3.15 WAZUH

Wazuh se posiciona como la arquitectura de código abierto con mayor nivel de implementación en el ámbito de la ciberseguridad, logrando la convergencia de las capacidades de Detección y Respuesta Extendidas (XDR) y la Gestión de Eventos e Información de Seguridad (SIEM) en una infraestructura unificada.

A través del análisis sistemático de telemetría proveniente de terminales, entornos de computación en la nube y activos de red, la plataforma facilita la identificación proactiva de amenazas, la mitigación de incidentes y la observancia de marcos normativos. De este modo, el sistema permite a las organizaciones robustecer su resiliencia operativa mediante el monitoreo constante y la implementación de procesos automatizados[39].

a. Dimensiones de Aplicación y Casos de Uso

En [39] describe la versatilidad de la plataforma la cual permite su despliegue en diversos escenarios operativos, fortaleciendo la resiliencia organizacional mediante:

- **Evaluación de la Configuración y Vulnerabilidades:** Identificación sistemática de debilidades de seguridad y desajustes en las políticas de endurecimiento (hardening).
- **Monitoreo de Integridad de Archivos (FIM):** Supervisión en tiempo real de cambios no autorizados en archivos críticos y registros del sistema.
- **Detección de Malware y Respuesta ante Incidentes:** Herramientas para la defensa y evasión de agentes sospechosos o maliciosos al sistema.
- **Análisis de Registros y Búsqueda de Amenazas:** Examen exhaustivo para la detección de indicadores de compromiso que no llegan a ser detectadas en tiempo real.

- **Seguridad en Contenedores y Cargas de Trabajo:** Protección especializada para entornos virtualizados, garantizando la visibilidad en arquitecturas de microservicios.

b. Arquitectura de Componentes Centrales

El funcionamiento de Wazuh se sustenta en una arquitectura distribuida compuesta por cuatro pilares fundamentales:

Tabla 4. Arquitectura de Wazuh[39]

Componente	Función Académica / Técnica
Wazuh Indexer	Motor de búsqueda y análisis de texto completo, caracterizado por su alta escalabilidad. Se encarga del indexado y almacenamiento de alertas bajo estructuras de clúster para optimizar la recuperación de datos.
Wazuh Server	Se encarga del procesamiento de los diferentes agentes que posee el cual implementa varios codificadores para procesar la información de forma correcta de las amenazas.
Wazuh Dashboard	Interfaz gráfica editable para la visualización de anomalías y detección de varias reglas predefinidas acerca de tipos de escaneos o ataques.
Wazuh Agent	Componente administrable en los dispositivos finales para el monitoreo de datos hacia el servidor con el fin de actuar como el sensor primario, proporcionando capacidades locales de prevención, detección y respuesta inmediata.

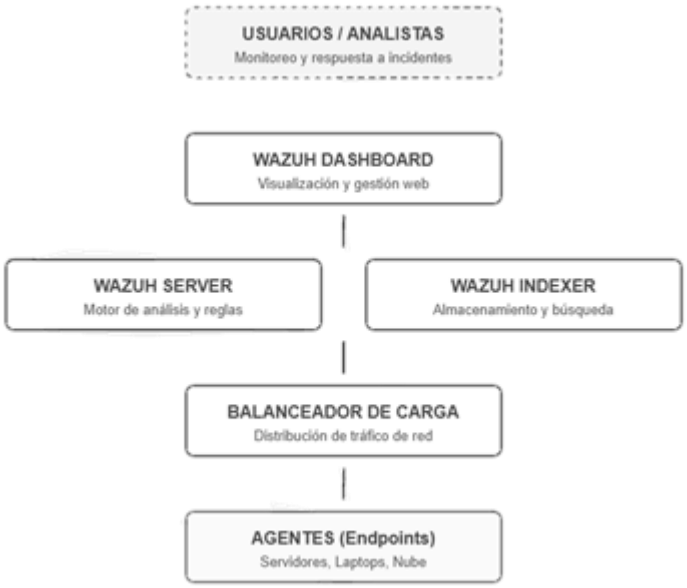


Figura 15. Jerarquía de la estructura general de Wazuh[39]

c. Fortalecimiento de la Postura de Seguridad (IT Hygiene)

A través de la automatización y el monitoreo continuo, la plataforma no solo detecta eventos adversos, sino que promueve una higiene de TI proactiva. Esto se traduce en una mejora constante de la postura de seguridad, asegurando que los activos cumplan con estándares de cumplimiento regulatorios y mejores prácticas de la industria, mitigando así el riesgo residual en entornos corporativos complejos [39].

d. Motores de Detección y Análisis de Reglas

- ***Decodificadores: Traduciendo el Caos***

  Un decodificador es el primer filtro que atraviesa un log. Dado que cada aplicación (Apache, SSH, Windows Event Log) escribe sus registros de forma distinta, el IDS necesita extraer datos clave para poder compararlos con las reglas [40].

- **Extracción de campos:** El decodificador identifica elementos como la IP de origen, el usuario, el ID del evento o el puerto.
- **Pre-decodificación:** Primero se extraen metadatos básicos (como el nombre del host y la fecha).
- **Decodificación específica:** Luego, mediante expresiones regulares, el motor busca patrones que coincidan con el formato de una aplicación conocida.

- ***Jerarquía de Reglas: Niveles de Alerta***

Wazuh organiza sus reglas de forma jerárquica y utiliza niveles de severidad numéricos (del 0 al 15). Esto permite filtrar el ruido y enfocarse en lo que realmente importa[41].

Tabla 5. Clasificación de alertas [41]

Niveles	Clasificación	Descripción
0 - 2	Informativo	No generan alertas. Se usan para estadísticas o ignorar eventos conocidos.
3 - 7	Bajo / Medio	Errores de autenticación, avisos de sistema o cambios menores en archivos.
8 - 11	Alto	Múltiples errores de login (posible fuerza bruta) o ataques web conocidos.

12 - 15	Crítico	Ataques confirmados, rootkits detectados o cambios en archivos críticos del sistema.

- Dependencia: Las reglas suelen tener un elemento <if_sid>, lo que significa que una regla de nivel alto solo se dispara si primero se cumplió una regla base (por ejemplo: Fallo de login nivel 5 $\rightarrow$ -10 fallos en 1- minuto nivel 10).

e. Análisis de Firmas vs. Comportamiento

Wazuh combina ambas metodologías para ofrecer una defensa integral.

A. Análisis de Firmas (Basado en Reglas)

Es el método tradicional. El IDS busca una "huella" o patrón específico que ya se conoce[42].

- Ventaja: Muy preciso y genera pocos falsos positivos para amenazas conocidas.
- Ejemplo: Una regla que detecta la cadena /etc/passwd en una solicitud HTTP (intento de Directory Traversal)[42].

B. Análisis de Comportamiento (Heurístico y Estadístico)

En lugar de buscar un patrón fijo, observa desviaciones en el funcionamiento normal del sistema.

- Detección de Anomalías: Si un usuario que suele conectarse desde Ecuador de repente inicia sesión desde un país distinto a una hora inusual.
- FIM (File Integrity Monitoring): Wazuh monitorea si archivos críticos cambian. No sabe "quién" es el virus, pero sabe que el archivo hosts no debería haber sido modificado [42].
- Rootcheck: Escaneo periódico del sistema buscando procesos ocultos o puertos abiertos sospechosos que no coinciden con las firmas estándar [42].

1.3.16 Hipervisor

El hipervisor, también conocido como monitor de máquina virtual (Virtual Machine Monitor, VMM), se define como un software de virtualización cuya función principal consiste en facilitar la creación y administración de máquinas virtuales (Virtual

Machines, VMs). Su operación se fundamenta en la traducción de solicitudes entre los recursos físicos y virtuales, lo que permite desacoplar el software de una computadora respecto de su hardware subyacente[43].

a. Hipervisores en entornos de nube

Con la creciente adopción de la computación en nube, el hipervisor se ha consolidado como una herramienta indispensable para la ejecución de máquinas virtuales y el fomento de la innovación en dichos entornos. Al ser en un software se puede desplegar varias máquinas que brindan diferentes servicios dentro de un contenedor físico brindando diferentes servicios en la nube. Los hipervisores diseñados específicamente para plataformas de nube permiten a los proveedores ofrecer instancias aisladas y seguras, así como escalar o reducir dinámicamente dichos entornos [43].

b. Justificación del uso de hipervisores

El empleo de hipervisores posibilita una mayor utilización de los recursos disponibles en un sistema y confiere una movilidad superior a las cargas de trabajo de TI, dado que las máquinas virtuales invitadas son independientes del hardware del anfitrión, lo que facilita su migración entre diferentes servidores. Al ejecutar múltiples máquinas virtuales sobre un único servidor físico mediante un hipervisor, se logra reducir los requerimientos de espacio físico, consumo energético y necesidades de mantenimiento [43].

c. Principio de funcionamiento

Los hipervisores respaldan la creación y gestión de máquinas virtuales mediante la abstracción del software respecto del hardware. Esta capacidad de virtualización se alcanza traduciendo las solicitudes entre los recursos físicos y los virtuales. En el caso de los hipervisores de tipo 1 o de "metal desnudo" (bare-metal), estos pueden integrarse en el firmware al mismo nivel que el sistema básico de entrada/salida (BIOS) de la placa base, permitiendo así que el sistema operativo acceda y utilice directamente el software de virtualización[43].

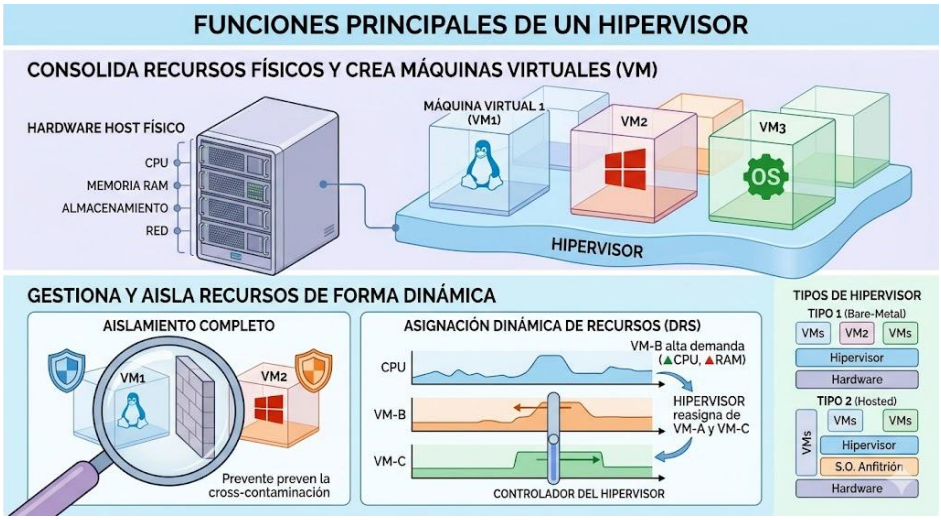


Figura 16. Funciones de un Hipervisor

1.3.17 Clasificación de los hipervisores

Los hipervisores se clasifican en dos tipologías principales según el enfoque de virtualización:

a. Tipo 1 (metal desnudo o básico)

Actúa como un sistema operativo liviano que se ejecuta directamente sobre el hardware del anfitrión. En esta modalidad, no existe ningún sistema operativo intermedio entre el software de virtualización y el hardware, ya que dicho software reside directamente sobre el "metal desnudo". Debido a su aislamiento respecto del sistema operativo —el cual es más propenso a ataques—, los hipervisores tipo 1 resultan extremadamente seguros y, por lo general, ofrecen un rendimiento superior y una mayor eficiencia que los hipervisores tipo 2. Por estas razones, numerosas organizaciones optan por hipervisores básicos para sus centros de datos. Ejemplos representativos incluyen ESX, Hyper-V, Citrix Hypervisor, Xen y Linux KVM [43].

b. Tipo 2 (alojado o hospedado)

Se ejecuta como una capa de software sobre el sistema operativo de la máquina anfitriona, de manera similar a cualquier otro programa informático. Aunque residen dentro del sistema operativo, permiten instalar sistemas operativos adicionales y heterogéneos sobre ellos. Su principal desventaja radica en una latencia superior a la

de los hipervisores tipo 1, debido a que la comunicación entre el hardware y el hipervisor debe atravesar la capa adicional del sistema operativo. Se denominan también hipervisores de cliente, ya que su uso es frecuente en entornos de usuario final y en pruebas de software, donde una mayor latencia resulta menos crítica. Ejemplos notables son VMware Fusion, Oracle VM VirtualBox y VMware Workstation [43].

Tabla 6. Cuadro comparativo de los tipos de hipervisores[43]

Característica	Hipervisor Tipo 1 (Bare-metal / Metal)	Hipervisor Tipo 2 (Hosted / Alojado)
Instalación	Directamente sobre el hardware físico (sin SO previo).	Sobre un sistema operativo ya instalado (Windows, Linux, macOS).
Arquitectura	Actúa como el propio sistema operativo básico.	Funciona como una aplicación más dentro del sistema operativo.
Rendimiento	Máximo. Acceso directo a los recursos de CPU y RAM.	Menor. El SO anfitrión consume recursos y añade latencia.
Seguridad	Mayor. Menos capas expuestas a vulnerabilidades.	Menor. Si el SO anfitrión falla, todas las VMs caen.
Uso Principal	Centros de datos, servidores de producción y nubes.	Desarrollo, pruebas rápidas y uso educativo.
Ejemplos	VMware ESXi, Microsoft Hyper-V, KVM, Xen.	VMware Workstation, VirtualBox, VMware Fusion.

1.3.18 Aceleración de hardware

La tecnología de aceleración de hardware permite crear y administrar recursos virtuales con mayor celeridad al incrementar la velocidad de procesamiento tanto para hipervisores tipo 1 como tipo 2. Un ejemplo es el acelerador virtual de gráficos dedicados (virtual Dedicated Graphics Accelerator, vDGA), el cual se encarga del procesamiento y actualización de gráficos 3D de alta gama, liberando al sistema principal para otras tareas y mejorando sustancialmente la velocidad de visualización de imágenes. Esta tecnología resulta de gran utilidad en sectores como la exploración de petróleo y gas, donde se requiere visualizar rápidamente datos complejos[43].

a. Consideraciones sobre entornos multi inquilino

Ambos tipos de hipervisores pueden ejecutar múltiples servidores virtuales para varios inquilinos sobre una misma máquina física. En la nube pública, los proveedores de servicios asignan espacio de servidor en diferentes servidores virtuales a distintas empresas. Un mismo servidor físico puede albergar múltiples servidores virtuales que ejecutan cargas de trabajo para diversos clientes. No obstante, este modelo de compartición de recursos puede generar el denominado efecto "vecino ruidoso" (noisy neighbor), en el cual uno de los inquilinos ejecuta una carga de trabajo intensiva que interfiere en el rendimiento del servidor para los demás inquilinos. Asimismo, este enfoque conlleva un riesgo de seguridad mayor en comparación con el uso de un servidor básico dedicado. Un servidor físico sobre el cual una única empresa posee control total proporciona invariablemente un rendimiento superior al de un servidor virtual que comparte ancho de banda, memoria y capacidad de procesamiento con otros servidores virtuales. El hardware de los servidores físicos también puede optimizarse para incrementar el rendimiento, algo que no ocurre en los servidores públicos compartidos. Las empresas que deben cumplir con normativas que exigen la separación física de los recursos requieren emplear sus propios servidores físicos que no compartan recursos con otros inquilinos [43].

1.3.19 Comparación entre hipervisores y contenedores

Los hipervisores y los contenedores contribuyen a mejorar la rapidez y eficiencia de las aplicaciones, aunque mediante enfoques distintos. Los hipervisores permiten que un sistema operativo se ejecute de forma independiente del hardware subyacente mediante el uso de máquinas virtuales, comparten recursos virtuales de cómputo, almacenamiento y memoria, y pueden ejecutar múltiples sistemas operativos sobre un servidor (en el caso de hipervisores tipo 1) o instalarse sobre un sistema operativo estándar y aislarse de él (en el caso de hipervisores tipo 2). Por su parte, los contenedores permiten que las aplicaciones se ejecuten de manera independiente de un sistema operativo, funcionan sobre cualquier sistema operativo siempre que dispongan de un motor de contenedores y son extremadamente portátiles, ya que cada contenedor incluye todo lo necesario para su ejecución. Mientras los hipervisores se

emplean para crear y ejecutar máquinas virtuales —cada una con su propio sistema operativo completo y aislada de las demás—, los contenedores empaquetan únicamente una aplicación y sus servicios asociados, resultando más livianos y portátiles, por lo que se utilizan frecuentemente en entornos de desarrollo y en despliegues ágiles de aplicaciones[43].

Tabla 7. Comparativa de Hipervisores vs Contenedores[43]

Característica	Hipervisores (Máquinas Virtuales)	Contenedores
Arquitectura	Cada instancia incluye un Sistema Operativo (SO) completo, aplicaciones y binarios.	Comparten el núcleo (kernel) del SO anfitrión y solo contienen la aplicación y sus dependencias.
Aislamiento	Alto: El aislamiento es a nivel de hardware, lo que ofrece mayor seguridad.	Moderado: El aislamiento es a nivel de proceso, compartiendo recursos del SO.
Peso / Tamaño	Pesado: Gigabytes por instancia debido al SO invitado.	Ligero: Megabytes, ya que no requieren un SO completo por instancia.
Tiempo de Inicio	Minutos (debe arrancar todo el sistema operativo).	Segundos (arranca como un proceso más del sistema).
Uso de Recursos	Mayor consumo de CPU, RAM y almacenamiento.	Uso eficiente y dinámico de los recursos del host.

1.3.20 Beneficios del uso de hipervisores

El empleo de hipervisores que alojan múltiples máquinas virtuales reporta diversos beneficios:

- Velocidad: Los hipervisores permiten crear máquinas virtuales de forma casi instantánea, a diferencia de los servidores físicos, lo que facilita el aprovisionamiento dinámico de recursos para cargas de trabajo variables[43].
- Eficiencia: Al ejecutar varias máquinas virtuales sobre los recursos de una única máquina física, los hipervisores posibilitan una utilización más eficiente del servidor físico, resultando más rentable y energéticamente eficiente que operar múltiples servidores físicos infrautilizados para una misma función[43].
- Flexibilidad: Los hipervisores de tipo 1 (metal desnudo) permiten que los sistemas operativos y sus aplicaciones asociadas se ejecuten sobre diversos tipos de hardware, dado que el hipervisor separa el sistema operativo del

hardware subyacente, eliminando la dependencia de dispositivos o controladores específicos[43].

- Portabilidad: Dado que las máquinas virtuales ejecutadas por el hipervisor son independientes de la máquina física, resultan fácilmente portables. Los equipos de TI pueden reasignar cargas de trabajo y distribuir recursos de red, memoria, almacenamiento y procesamiento entre múltiples servidores según los requerimientos, trasladándolos de una máquina a otra o entre distintas plataformas. Cuando una aplicación demanda mayor capacidad de procesamiento, el software de virtualización permite acceder sin fisuras a recursos computacionales adicionales[43].

1.3.21 VMWare Workstation

VMware Workstation se define como una línea de soluciones de hipervisor de tipo alojado (hosted), diseñada para operar sobre arquitecturas de computación x64. Esta tecnología faculta la ejecución simultánea de máquinas virtuales, contenedores y clústeres de Kubernetes dentro de una única unidad física, permitiendo la coexistencia operativa con el sistema operativo anfitrión. Cada instancia virtual posee la capacidad de gestionar de manera independiente su propio sistema operativo, abarcando un espectro que incluye distribuciones de Windows, Linux, BSD y MS-DOS. Desarrollado por VMware, Inc. (entidad perteneciente a Dell Technologies), este software se orienta a la optimización de recursos de hardware[44].

Desde una perspectiva funcional, la herramienta facilita la implementación concurrente de diversos sistemas operativos, tanto de estaciones de trabajo como de servidores. Su utilidad es fundamental en el ámbito de la administración de sistemas y redes, ya que proporciona un entorno controlado para la revisión, experimentación y validación de arquitecturas cliente/servidor. Asimismo, el software garantiza a los usuarios autorizados la transición fluida y el acceso simultáneo a múltiples entornos virtuales, promoviendo la eficiencia en tareas de desarrollo y pruebas técnicas[44].

Tabla 8. Ventajas y desventajas de VMware Workstation

Aspecto Técnico	Ventajas Técnicas (Fortalezas)	Desventajas Técnicas (Limitaciones)
Arquitectura y Rendimiento	Rendimiento de CPU/RAM nativo cercano: Gracias a la aceleración por hardware (Intel VT-x, AMD-V), el rendimiento para cálculo puro es excelente. Aislamiento de hardware: El sistema operativo del invitado no afecta al de las máquinas virtuales y al anfitrión.	Hipervisor de Tipo 2: Necesita un sistema operativo completo y superior al de las máquinas virtuales. Recursos Altos: Los procesos de las máquinas virtuales consumen RAM del anfitrión por lo que es necesario una RAM elevada para el rendimiento de las máquinas virtuales.
Gestión de Recursos Dinámicos	Configuración de recursos: La configuración de ajustes como RAM, procesadores , espacio son de forma precisa y reconfigurable. Tecnologías de ahorro de memoria: Utiliza técnicas como memory ballooning y compartición de páginas de memoria.	Dependencia del Anfitrión: El S.O del anfitrión falla o se degrada todas las VMs también se degradan. Retraso en los flujos de entrada y salida: La lectura o escritura de las VMs deben pasar por el sistema operativo anfitrión.
Capacidades de Red	Editor de red virtual avanzado: Permite crear topologías complejas, simular latencia de red, pérdida de paquetes y ancho de banda limitado. Modos versátiles: Soporta Puente (Bridged), NAT, Solo anfitrión y segmentos de área local.	Curva de aprendizaje: Configurar redes complejas requiere conocimientos sólidos de networking. Conflictos de controladores: En ocasiones, las actualizaciones del S.O. anfitrión pueden corromper la configuración de los adaptadores de red virtuales de VMware.
Gráficos y 3D	Tarjeta de Video: Se asigna una memoria de video dedicada para el correcto funcionamiento de los controladores de video	Dependencia de anfitrión: La GPU de las VMs depende de la memoria grafica del anfitrión
Herramientas y Gestión	Clonación e Instantáneas: Permite la clonación de VMs además de realizar puntos de guardado de máquinas virtuales en diferentes instancias.	Manejo de instantáneas: Tener muchos puntos de guardado afecta el rendimiento de la máquina virtual. Actualización Manual de VM Tools: Cada actualización del software

	VMware Tools: Es un controlador para los periféricos de entrada como de salida.	requiere también actualizar o reinstalar VM Tools.
Compatibilidad	Soporte de Sistemas Operativos: Soporta cualquier sistema operativo que sea instalado dentro de VMWare ya sea Windows, Linux en todas sus distribuciones y versiones de Android.	Hardware Fisico Limitado: No se puede instalar sistemas que demanden de un dispositivo físico especializado. Sistema Operativo Acondicionado: Solo se puede realizar la instalación en Linux o Windows siendo exclusivos.

1.3.22 Virtualización de Redes

El Networking Definido por Software es un cambio a la administración de redes anteriores. Este sistema deja de depender de equipos físicos y empieza a utilizar APIs para hablar con el hardware y redireccionar el tráfico según sea necesario para su estructura informática [45].

a. Arquitectura y Funcionamiento

El principio fundamental del SDN es la desacoplamiento del software y el hardware. Esta arquitectura traslada el plano de control (encargado de decidir hacia dónde enviar el tráfico) al software, manteniendo el plano de datos (responsable del reenvío físico de paquetes) en los dispositivos de hardware[45].

De acuerdo con el modelo estándar, la arquitectura SDN se divide en tres niveles:

- Aplicaciones: Transmiten solicitudes de recursos o información sobre el estado global de la red [45].
- Controladores: Procesan la información de las aplicaciones para determinar las rutas óptimas de los paquetes de datos [45].
- Dispositivos de red: Realizan todos los comandos que envía el controlador para enviar datos a través de la infraestructura ya sea física o virtual [45].

b. Importancia y Beneficios Estratégicos

La implementación de SDN resulta crítica para los servicios modernos, especialmente en entornos de nube, debido a las siguientes ventajas funcionales:

- Control y Flexibilidad: No se necesita aprender un modo distintivo de programar por cada fabricante. El uso de estándares abiertos para la gestión de tráfico hace que se puedan interconectar varias quipos de diferentes marcas [45].
- Infraestructura Personalizable: Se puede ajustar los servicios y asignar recursos virtuales desde un solo punto, sin la necesidad de ir equipo por equipo [45].
- Seguridad más Alta: Se tiene una vista general de toda la red, permite crear zonas con distintos niveles de seguridad, se puede aislar al instante [45].
- Escalabilidad y Agilidad: Mover cargas de trabajo o ampliar el sistema de la red es más fácil y ágil [45].

c. Modelos de Implementación

Existen diversos enfoques para la adopción de redes definidas por software:

Tabla 9. Modelos de Implementación de SDN [45]

Modelo	Descripción Técnica
Open SDN	Utiliza protocolos abiertos como OpenFlow para controlar el comportamiento de los interruptores en el plano de datos.
SDN por APIs	Emplea interfaces de programación para controlar el movimiento de datos en cada dispositivo individual.
Modelo Overlay	Crea una red virtual sobre la infraestructura ya existente se puede ampliar sin necesidad de la configuración individual de cada equipo.
SDN Híbrido	El sistema integra protocolos tradicionales y SDN sin ningún problema es una red combinada hibrida.

1.3.23 Comparativa: SDN vs. Networking Tradicional

La distinción fundamental radica en la infraestructura: mientras el modelo tradicional es basado en hardware, el SDN es basado en software. Esto otorga al SDN una

flexibilidad superior para ajustar configuraciones y aprovisionar recursos desde una interfaz de usuario centralizada [45].

No obstante, se debe considerar que la centralización del control introduce un punto único de falla. Por lo tanto, garantizar la integridad y seguridad del controlador central resulta indispensable para la estabilidad de toda la infraestructura [45].

1.3.24 VMware vSwitch

Las máquinas virtuales se conectan de una manera similar a una red física común. La diferencia está en que las máquinas virtuales pueden utilizar adaptadores y conmutadores virtuales para enlazarse de forma diferente según la necesidad del usuario[46]. Cada uno de ellos utiliza un conmutador virtual diferente:

- VMnet0 (Red puenteada):La máquina virtual se conecta a la misma red que el quipo físico siendo un equipo más dentro de esta red [46].
- VMnet1 (Solo Host): La máquina virtual queda en una subred diferente y solo se puede comunicar con el quipo anfitrión, no con toda la red externa [46].
- VMnet8 (red NAT): La máquina virtual se encuentra detrás de una NAT en una subred separada. Por lo que se puede comunicar con el host utilizando esa dirección IP pero no queda expuesta directamente a la red externa [46].

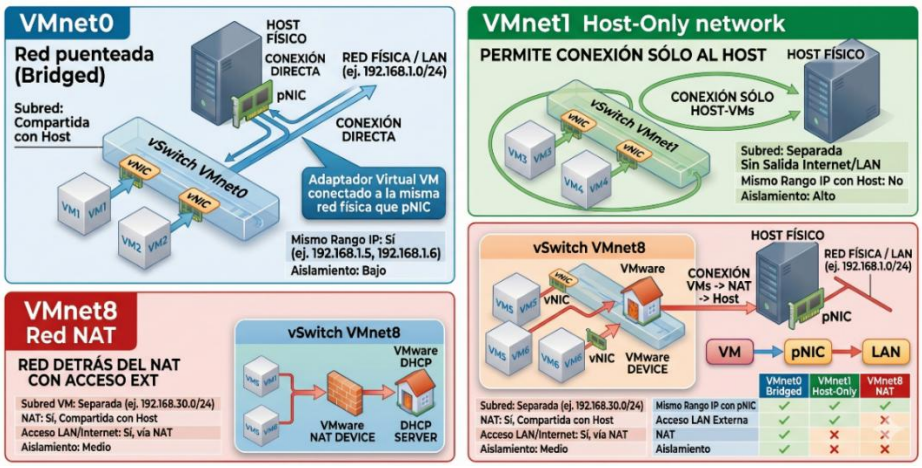


Figura 17. Resumen de redes VMware WORKSTATION

1.3.25 Taxonomía de Ciberataques Aplicados al Entorno de Red

Los vectores de ataque se clasifican según su naturaleza operativa, el recurso explotado y el objetivo final del agente malicioso. En el desarrollo experimental de esta investigación, se han priorizado los vectores orientados a la vulneración de credenciales y a la degradación de la conectividad [47].

1.3.26 Ataques Orientados a Mecanismos de Autenticación

Los ataques a contraseñas representan un conjunto de técnicas automatizadas diseñadas para comprometer las credenciales de acceso de un host o servicio web mediante la explotación de malas prácticas de gestión de identidad[47].

1.3.27 Ataque de Fuerza Bruta (Brute Force)

Consiste en un método matemático de ensayo y error en el que un software especializado computa de forma secuencial y masiva combinaciones aleatorias de caracteres, letras y números hasta determinar el patrón exacto que compone la clave criptográfica. El impacto sistémico radica en la generación de ráfagas de registros fallidos observables en el archivo de auditoría /var/log/auth.log de la máquina víctima [47].

1.3.28 Ataque por Diccionario

A diferencia de la fuerza bruta pura, este vector optimiza los ciclos de procesamiento mediante la carga de bases de datos estructuradas o listas de palabras predefinidas (wordlists) que contienen contraseñas comunes, datos personales o variaciones léxicas predecibles [47].

1.3.29 Ataques Dirigidos a las Conexiones y Disponibilidad de Red

Los ataques pueden darse en diferentes capas: enlace, transporte y la de aplicación. Pero tienen algo en común: intentan entrar en las intermediaciones de los mismos y saturar la infraestructura local por, en la cual se puede extraer información sensible o hacer que el host funcione de manera ineficiente [47].

1.3.30 Denegación de Servicio Distribuido y Denegación de Servicios

Un ataque de denegación de servicio consiste en la inundación maliciosa y simultánea de peticiones hacia un host desde uno o múltiples vectores de origen. El objetivo técnico es saturar los recursos de procesamiento de la máquina víctima (tales como la cola de conexiones del kernel o la capacidad del servidor web), provocando la inestabilidad del sistema y la indisponibilidad de plataformas de software alojadas, como los entornos Moodle [47].

1.3.31 Suplantación de Identidad (Spoofing)

Técnicas de hacking que falsifican los parámetros de identidad de red de una entidad virtual. Dentro de esta categoría destaca el IP Spoofing, mediante el cual el atacante altera la cabecera de los paquetes de datos para simular una dirección IP de origen distinta y evadir las reglas restrictivas de los enrutadores o cortafuegos [47].

1.3.32 Escaneo de Puertos (Port Scanning)

Fase de reconocimiento automatizado en la que se inyectan paquetes de prueba a un rango de puertos de red de una máquina. Su finalidad es evaluar las respuestas del protocolo (puertos abiertos, cerrados o filtrados) para identificar servicios vulnerables activos dentro del host antes de perpetrar la intrusión [47].

1.3.33 Ataques Basados en Software Malicioso (Malware)

El malware comprende programas diseñados para ejecutarse de manera oculta en el sistema anfitrión con el propósito de vulnerar la confidencialidad y el control del dispositivo [47].

1.3.34 Puertas Traseras (Backdoors)

Código malicioso que, tras establecerse en el sistema operativo, elude los mecanismos estándar de autenticación, habilitando canales de acceso y control remoto asíncrono para el atacante [47].

1.3.35 Registradores de Teclas (Keyloggers)

Software o hardware dedicado a la interceptación y almacenamiento de cada pulsación realizada en el teclado físico del sistema, comprometiendo directamente datos bancarios y credenciales de administración antes de ser procesados por las capas de cifrado superiores[47].

1.4 Objetivos

1.4.1 Objetivo general

Implementar un sistema de detección de intrusos (IDS) en redes informáticas aplicando protocolos de ciberseguridad.

1.4.2 Objetivos específicos

- Analizar el funcionamiento de los sistemas de detección de intrusos de código abierto, con el fin de identificar y documentar sus requisitos técnicos.
- Diseñar una topología de red para el sistema de detección de intrusos.

- Implementar un módulo IDS en base al funcionamiento de la detección de intrusos dentro de una red informática.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

El desarrollo de la presente investigación y la implementación del laboratorio experimental se fundamentaron en el uso de recursos de hardware de alta densidad y un ecosistema de software especializado en virtualización y auditoría de seguridad. Como núcleo del procesamiento físico, se empleó una estación de trabajo de factor de forma pequeño, específicamente una Mini PC marca NiPoGi, modelo AM06Pro. Este equipo integra un procesador AMD Ryzen 7 7730U con arquitectura de 8 núcleos físicos y 16 hilos de ejecución, operando a una frecuencia base de 2.00 GHz. Esta capacidad de cómputo resulta crítica para soportar la carga de trabajo de un hipervisor que debe gestionar múltiples sistemas operativos simultáneamente sin degradar la respuesta del sistema anfitrión.

Para que las máquinas virtuales puedan tener el mejor rendimiento en la ejecución de las prácticas de laboratorio previstas se dispone de un hardware que cuenta con 32,0 GB de memoria RAM DDR4. Esta memoria es óptima para el uso de las diferentes máquinas virtuales su tráfico de red además de sistemas operativos que consumirán recursos. Por el lado de almacenamiento de software, se dispone de una unidad de almacenamiento de estado sólido (M2.SSD) de 512 GB, la cual será dispuesta para las imágenes de disco virtual (.vmdk). La memoria de tipo SSD reduce todas las latencias de forma adecuada en la entrada y salida, y que los registros de tráfico se realicen de manera eficiente y ágil para las diferentes aplicaciones que se realice.

Por la parte del sistema operativo base cuenta con Windows 11 Pro en su versión 24H2 que viene instalada de fabrica en el equipo. Sobre este sistema operativo, se instaló VMware Workstation Pro, la cual cuenta con una licencia gratis en su última versión el cual permite la simulación de redes para un laboratorio de forma que sea escalable además de reconfigurable con tecnología SDN. Dentro de la simulación de la red

informática cuenta con su adaptador de red de tipo NAT el cual se usa para compartir la dirección IP del host.

Dentro de este entorno virtual se dispone de tres versiones de Linux en su distribución Ubuntu cada una para un rol diferente en la arquitectura de red. Por otro lado el uso de la máquina virtual ya predefinida de Kali Linux la cual posee varias herramientas para el ciberataque controlado y simulado dentro de la red ya establecida. Las interacciones de este conjunto de máquinas virtuales gracias a la tecnología de memoria SSD hacen que la intercomunicaciones, el procesamiento de datos sea muy ágil para el conjunto del módulo de ciberseguridad planteado.

Se optó por el adaptador de red en modo NAT debido a las siguientes ventajas técnicas observadas en el entorno experimental:

- Direcciones IP: Se usa el servicio de DHCP interno de VMware en modo nat, las direcciones IP que se asignan de forma automática a Ubuntu y Kali Linux se mantendrán siempre igual, ya que en caso de cambiar de red se perderán las IPs ya establecidas haciendo que la intercomunicación entre los agentes y el servidor IDS queden perdidas y teniendo que realizar nuevamente la instalación.

- Seguridad y Aislamiento de la red: El modo Nat también actúa como un cortafuegos. El laboratorio de ciberseguridad puede salir a internet sin ningún problema para la descarga de paquetes o programas adicionales sabiendo que este módulo es reconfigurable y adaptable a nuevos escenarios, pero al mismo tiempo, los equipos que estén fuera de la red virtual no podrán iniciar una conexión hacia las diferentes máquinas virtuales creadas.

- Independencia del Hardware: Esta configuración garantiza que el laboratorio sea completamente portable. El sistema de detección de intrusos mantiene su operatividad interna sin necesidad de reconfiguraciones manuales, independientemente de si el host físico está conectado vía Ethernet, Wi-Fi o si se encuentra sin conexión a internet.

2.2 Modalidad de investigación

El presente trabajo de investigación se define bajo la modalidad de investigación aplicada, debido a que busca emplear los fundamentos teóricos de la seguridad informática para el diseño y despliegue de un Sistema de Detección de Intrusos (IDS) funcional. El objetivo principal es resolver la vulnerabilidad de las redes informáticas frente a accesos no autorizados mediante la implementación práctica de protocolos de ciberseguridad en un entorno controlado.

La investigación bibliográfica se fundamenta en la revisión exhaustiva de documentación técnica de diferentes sitios web además de proyectos de titulación de repositorios oficiales de universidades para el uso del software Wazuh así también como las diferentes distribuciones de Ubuntu y Kali Linux ideales para este módulo de ejecución de ciberataques controlados dentro de una red informática de forma virtual. Esta fase es la principal para establecer los criterios de seguridad y las diferentes reglas de detección de intrusos que rigen ante posibles ataques simulados.

Finalmente, la investigación se desarrolla de forma experimental con técnicas de ciberseguridad en un laboratorio SDN configurado en un alto rendimiento. Asegurando que el análisis de la detección de intrusos se basa en datos sumamente empíricos que se han ido generando durante las diferentes pruebas de ciberseguridad establecidas en laboratorios técnicos.

2.3 Recolección de información

Para la recolección de información se utilizaron fuentes basadas en libros, fuentes de internet, proyectos académicos y artículos. A continuación, se presenta la Tabla 10 en donde se explica la organización de la recolección de información.

Tabla 10. Recolección de información

Criterio	Descripción
¿Para qué?	Para implementar un módulo con un sistema de detección de intrusos en varios escenarios controlados.
¿De qué personas u objetos?	De proyectos prácticos y artículos académicos de diferentes repositorios con base en ciberseguridad.
¿Cómo?	A través de análisis de información y recolección de datos relacionados a tiempos de ataque y la detección de estos con diferentes sistemas de detección de intrusos
¿Con qué?	Utilizando herramientas para recolectar información como bases de datos de sistemas ya existentes a nivel profesional acerca de ciberseguridad.
¿Cuándo?	En todas las fases del desarrollo del proyecto.
¿Donde?	La recolección de información se la realizará de manera digital principalmente.

2.4 Procesamiento y análisis de datos

El procesamiento y análisis de datos se desarrollará a través de una serie de pasos a continuación:

- Recolección de información y datos obtenidos acerca de eventos de seguridad y tráfico de red para la detección de intrusiones.
- Análisis y comparación de diferentes tipos de sistemas de detección de intrusos (IDS), incluyendo IDS basados en firmas, IDS basados en anomalías e IDS híbridos, y su aplicación de protocolos de ciberseguridad.
- Selección de software y componentes de hardware de forma libre que sean adecuados para el desarrollo de un IDS.
- Desarrollo del módulo simulado del sistema de detección de intrusos para redes informáticas.
- Pruebas de funcionamiento y evaluación del IDS en escenarios de red simulados, aplicando diversos protocolos de ciberseguridad.

- Presentación de resultados y conclusiones sobre la eficacia del IDS desarrollado en la detección de intrusiones.

2.5 Propuesta de solución

El proyecto propone el desarrollo de un módulo de Sistema de Detección de Intrusos (IDS) para redes informáticas, aplicando protocolos de ciberseguridad y utilizando herramientas de código abierto y hardware libre en el cual se tendrá un minicomputador este se dispondrá de programas como VMWare Workstation que es de uso libre además de las diferentes imágenes o .iso de versiones de Ubuntu para su desarrollo siendo todo de forma virtualizada. El objetivo es ofrecer una solución accesible y reconfigurable para la detección temprana de actividades maliciosas, a diferencia de los sistemas propietarios que suelen ser costosos y de alta complejidad. Al emplear software de código abierto la implementación de esta tecnología se vuelve más accesible y adaptable a diferentes entornos de red, facilitando su entendimiento y mejora continua.

CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

3.1 Descripción de desarrollo

Mediante el uso de una Mini PC de alto rendimiento como nodo central de procesamiento, se dio paso al desarrollo de una guía de prácticas de laboratorio diseñadas para que los estudiantes de pregrado complementen los conocimientos teóricos aplicados a la ciberseguridad y la gestión de eventos de red. Esta implementación proporcionó una relación teoría-práctica robusta, integrando hardware compacto con el ecosistema de Software Libre, sirviendo como una introducción avanzada al despliegue de Sistemas de Detección de Intrusos (IDS) en entornos controlados.

El desarrollo de este proyecto de guías de prácticas para el monitoreo y respuesta ante incidentes utilizando Wazuh se basó en las etapas que se describen a continuación:

Inicio: Para el desarrollo de este trabajo es fundamental la adquisición de conocimientos previos en fundamentos de redes, protocolos de comunicación (TCP/IP), administración de sistemas operativos Linux y conceptos básicos de seguridad informática. Así mismo, se requiere familiaridad con el manejo de entornos virtualizados y contenedores, dado que el trabajo está dirigido a estudiantes con un perfil académico orientado a la ingeniería en Telecomunicaciones de la Universidad Técnica de Ambato.

Desarrollo: El mini PC funciona como un servidor dedicado a la simulación de servidores y agentes dentro de VMware, en el cual se plante de forma cronológica la creación de cada una de las máquinas la creación de los agentes dentro del uso de Wazuh con su respectiva interfaz gráfica, con Kali Linux se realiza las pruebas de detección de ataques con ello se evidenciarían de forma gráfica las alertas que genera el servidor ayudando a la comprensión más detallada de los tipos de ataques. Para ello, se usa diferentes herramientas de pentesting que permiten validar la eficacia de las reglas de detección y las respuestas activas del sistema.

Finalización: Como resultados, se obtiene el desarrollo de prácticas de laboratorio documentadas paso a paso. El fin de las prácticas es que los señores estudiantes adquieran habilidades necesarias para el análisis acerca de la ciberseguridad, la robustes de sistemas (hardening) y el manejo correcto y óptimo de herramientas de monitoreo de seguridad de tipo institucional o empresarial, fomentando la capacidad de respuesta ante incidentes a infraestructuras de alta demanda.

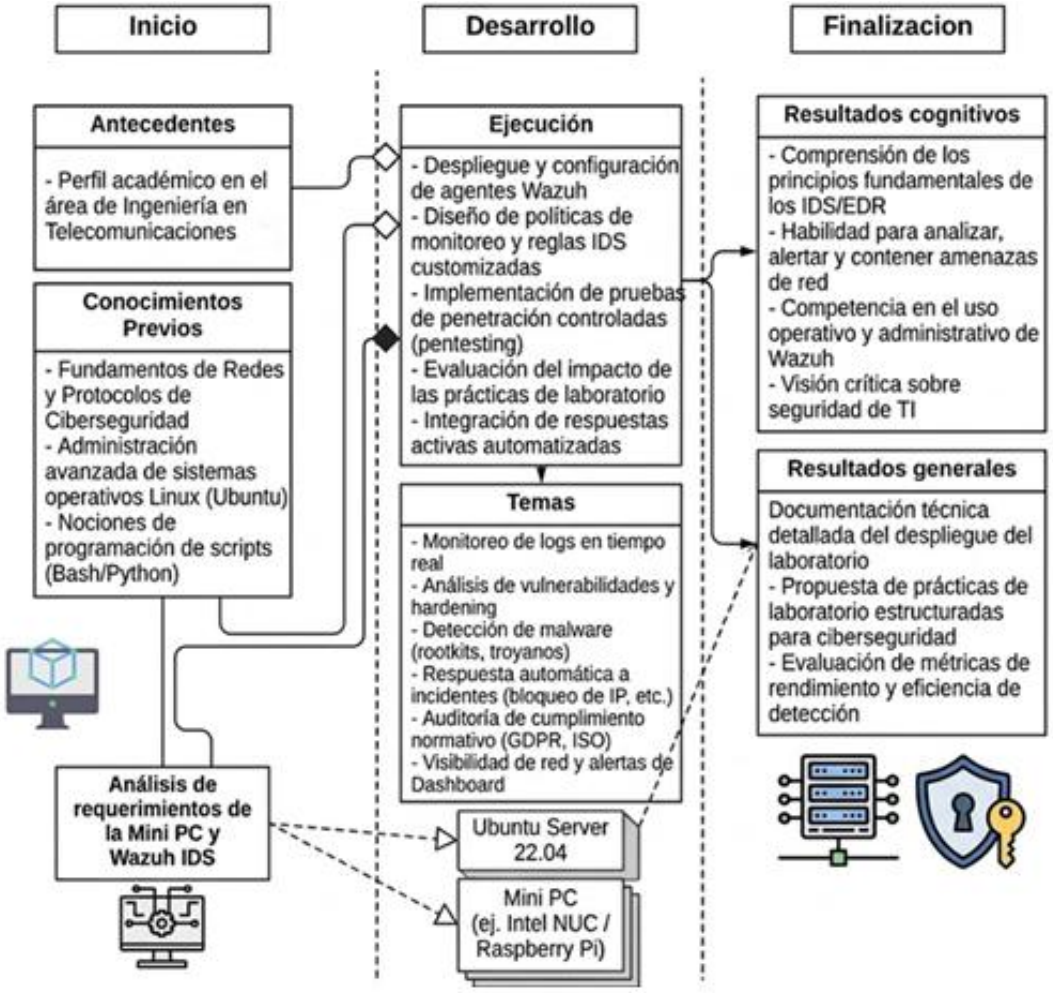


Figura 18. Descripción de desarrollo de proyecto

3.2 Desarrollo de practicas

Se presentan a continuación las prácticas de laboratorio en las cuales se detallarán paso a paso junto a los objetivos de estas.

3.2.1 Laboratorio N° 01

Tema: Creación y configuración de máquinas virtuales iniciales en VMware

1. Objetivos

Objetivo General: Crear las diferentes máquinas virtuales que serán aplicadas en el laboratorio de red mediante VMware para el sistema de detección de intrusos.

Objetivos Específicos:

- Definir los recursos de hardware virtual necesarios para el optimo flujo de un nodo agente y un nodo manager
- Establecer la comunicación de red en modo NAT para permitir la visibilidad entre las máquinas virtuales y la red física de la Mini PC.
- Preparar los contenedores virtuales para la posterior instalación de los sistemas operativos.

2. Recursos (Hardware y Software)

Hardware: Mini PC (Servidor Host).

Software: VMware Workstation Pro / Player.

Imagen ISO: Ubuntu Server 22.04 LTS (o la que utilices).

Una conexión a internet con un ancho de banda Mbps y compartición

3. Desarrollo de la Práctica

Paso 1: Iniciación de la creación de máquinas virtuales.

Abrir el asistente de nueva máquina virtual dentro de VMware y elegir la configuración típica de creación de máquina virtual.

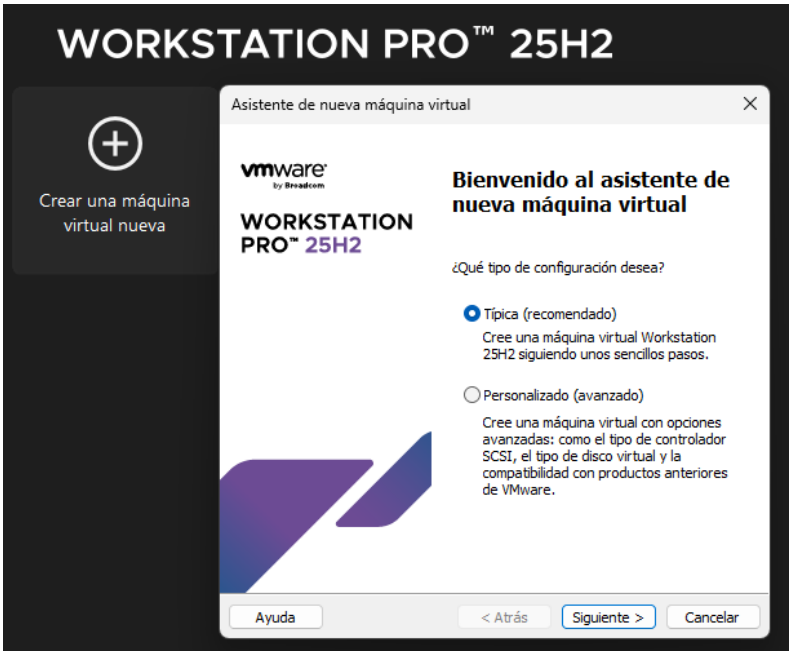


Figura 19. Creación de máquina virtual

Paso 2: Selección de la Imagen

Seleccionar la .iso a instalar en la maquina virtual

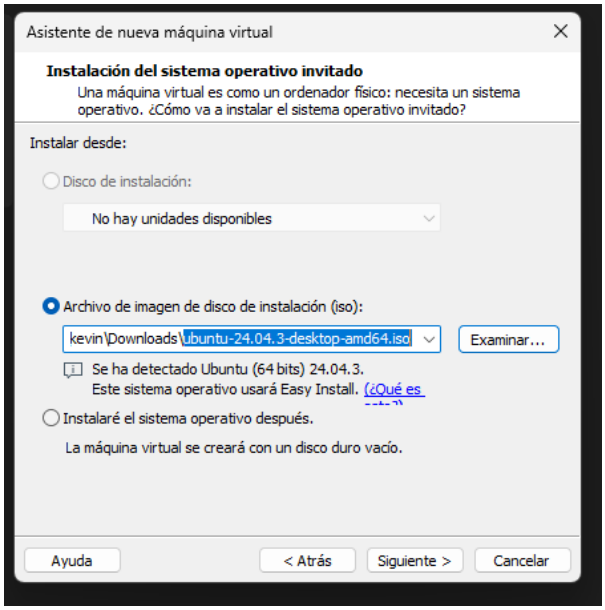


Figura 20. Selección de imagen (.iso)

Paso 3: Asignación de Recursos de Hardware

Para el trabajo optimo del servidor Wazuh se instalara con las siguientes características de hardware:

Memoria (RAM): Asignar 8 GB (8192 MB).

Procesadores: Asignar 4 núcleos (Cores).

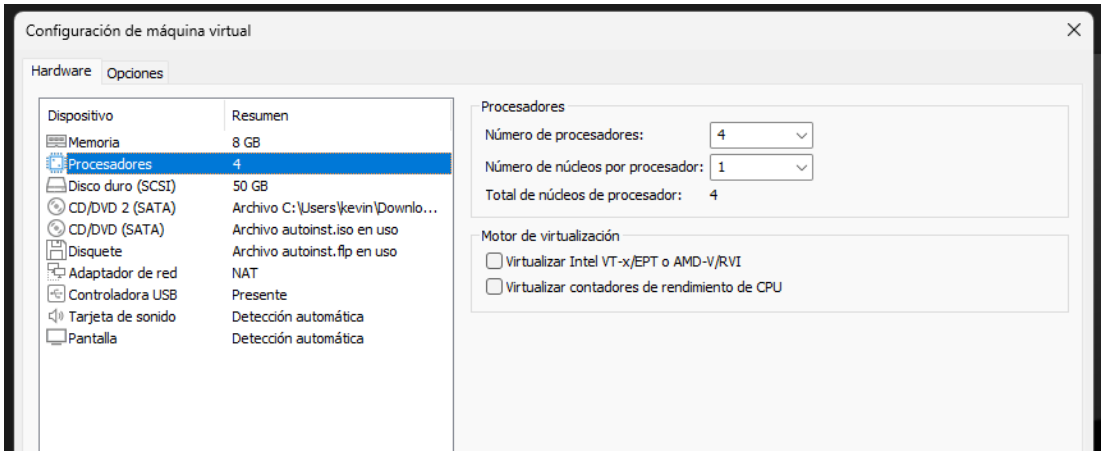


Figura 21. Asignación de recursos

Tamaño de disco: Definir un tamaño de 50 GB. Para el uso de Wazuh.

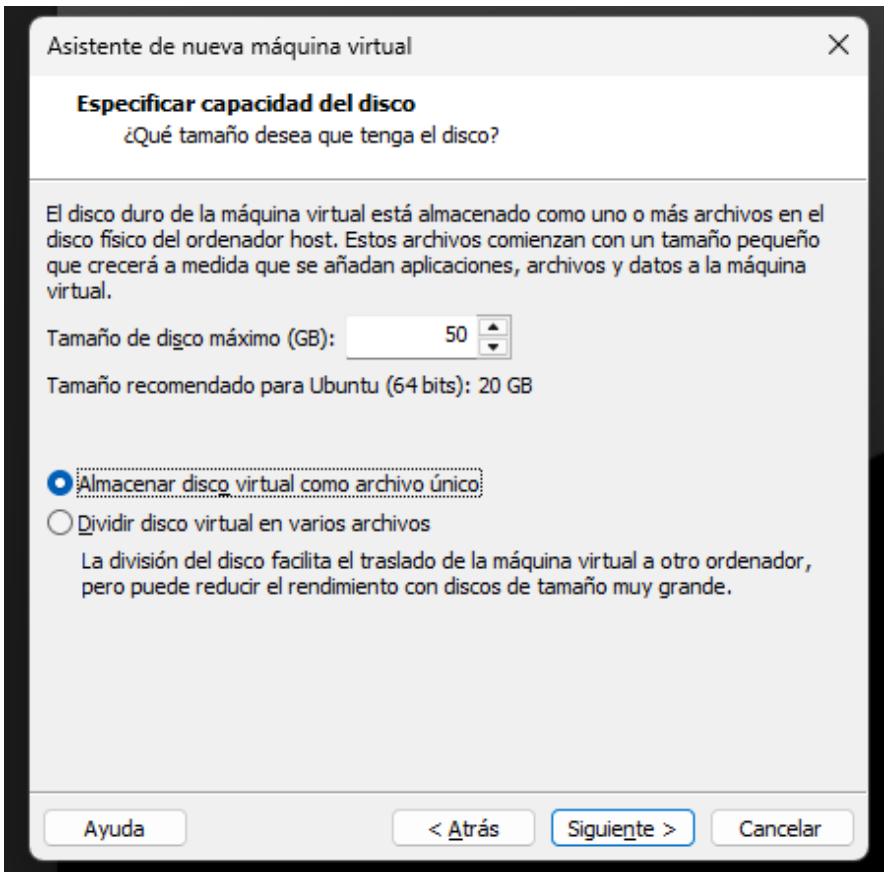


Figura 22. Asignación de memoria de disco

Paso 4: Configuración de la Interfaz de Red

Este es el paso más importante para la visibilidad del IDS:

En la configuración de la VM, acceder al apartado de adaptador de red

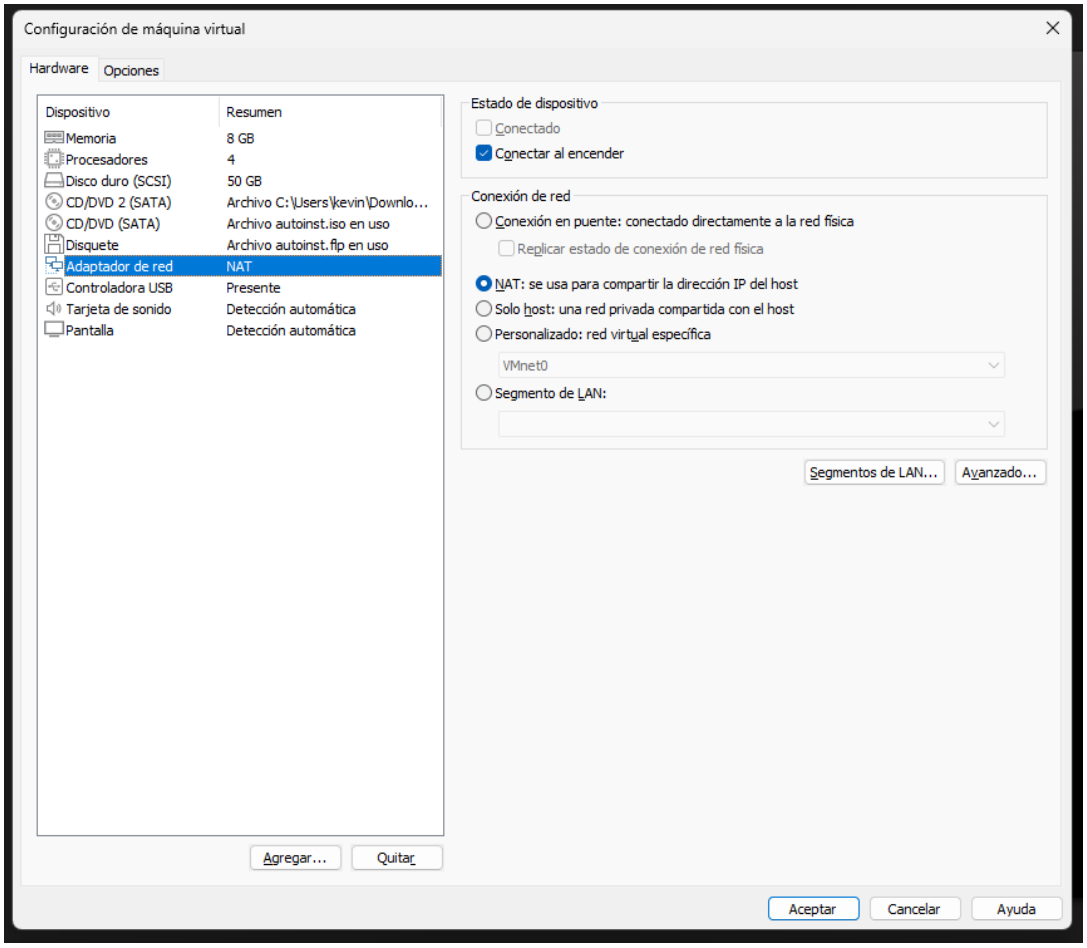


Figura 23. Configuración terminada de máquina virtual

Seleccionar el modo NAT: se usa para compartir la dirección ip del host.

Esto permite que la VM obtenga una dirección IP dentro del mismo segmento que la Mini PC y los demás dispositivos de la red.

Paso 5: Creación del Nodo Agente (Máquina Víctima PC1)

Se repite el proceso de creación de máquina virtual pero con las siguientes características de hardware :

Memory: 4 GB RAM.

Procesador: 4 núcleos.

Network: Modo NAT

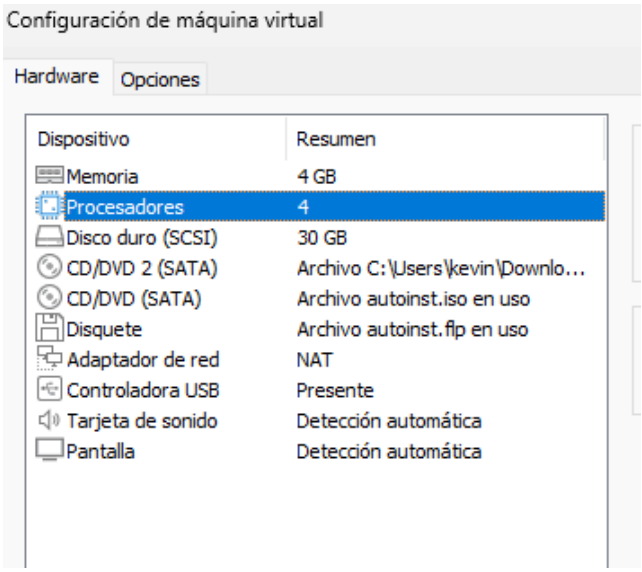


Figura 24. Configuración máquina virtual PC1

Paso 6: Creación de la maquina Kali Linux la cual será la atacante

Repetir el proceso anterior para crear una tercera máquina virtual con recursos predefinidos, ya que su función será realizar ataques controlados:

Memory: 2 GB RAM.

Procesadores: 4 núcleos.

Hard Disk: 80 GB

Network: Modo Net (obligatorio para que el Manager pueda detectarla).

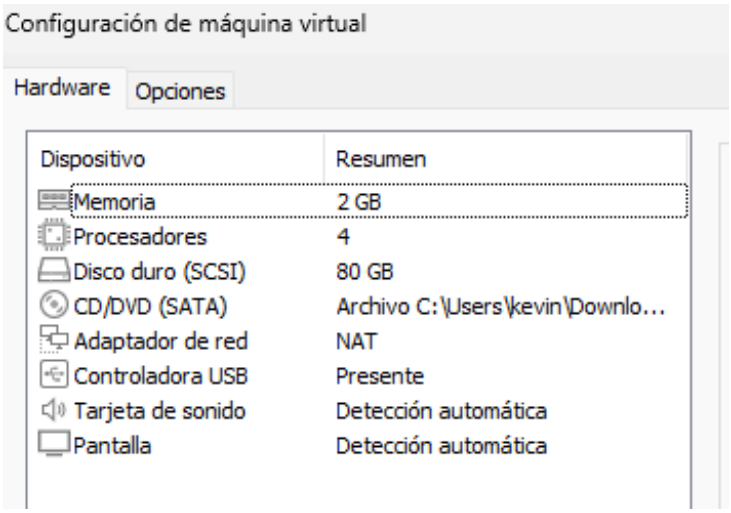


Figura 25. Configuración Máquina Atacante Kali Linux

Paso 7: Verificación de la Arquitectura

Antes de proceder a cualquier instalación, verificar en el panel principal de VMware que ambas máquinas aparezcan en el listado y que los recursos asignados no superen la capacidad física de la Mini PC para evitar desbordamientos de memoria.

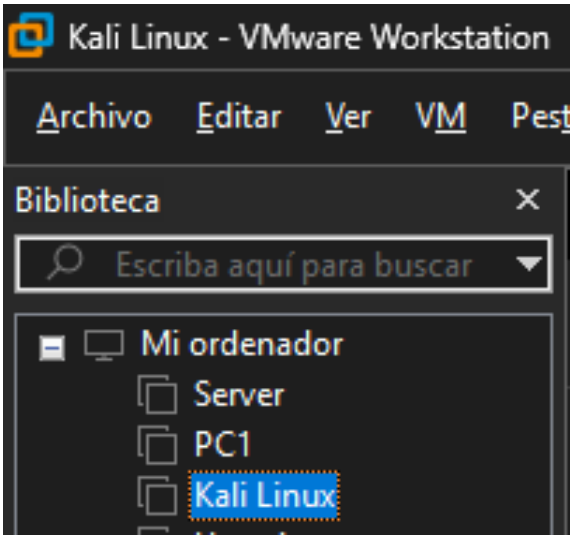


Figura 26. Máquinas creadas dentro de VMware

Ahora se contemplará la siguiente red informática

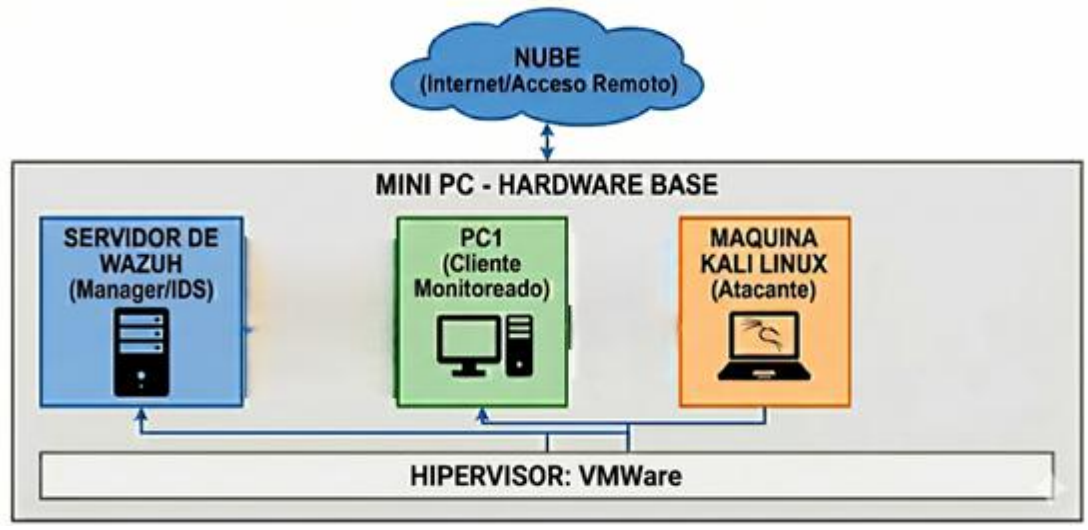


Figura 27. Diagrama de Laboratorio N° 01

3.2.2 Práctica de Laboratorio N° 02

Tema: Instalación de Wazuh y Despliegue del Agente PC1

1. Objetivos

Objetivo General: Instalar el ecosistema de seguridad Wazuh y establecer la comunicación bidireccional con un nodo externo (PC1).

Objetivos Específicos:

- Instalar de forma automatizada el agente, Server y Dashboard de Wazuh en el servidor central.
- Configurar el agente de Wazuh en el PC1 para el correcto funcionamiento.
- Verificar el estado del agente en el Dashboard de Wazuh.

2. Recursos (Hardware y Software)

- Servidor: Maquina con Ubuntu Server 22.04.
- Agente (PC1): Máquina Virtual con Linux.
- Acceso a internet para descarga de paquetes y red en modo NAT.

3. Desarrollo de la Práctica (Paso a Paso)

Paso 1: Preparación del Repositorio e Instalación del Servidor

En la terminal del servidor Ubuntu, se debe descargar y ejecutar el script de instalación oficial. Este comando instala todos los componentes necesarios (Indexador, Servidor y Dashboard) de forma integrada:

Terminal:

curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh

sudo Terminal: wazuh-install.sh -a

Al finalizar, el sistema generará las credenciales de acceso (usuario admin y una contraseña única). Es necesario documentar estos datos para el acceso al Dashboard.

Usuario: admin

Contraseña: Fhx32t7ow6xJaM?25EJXhz7PXt.SqmKF

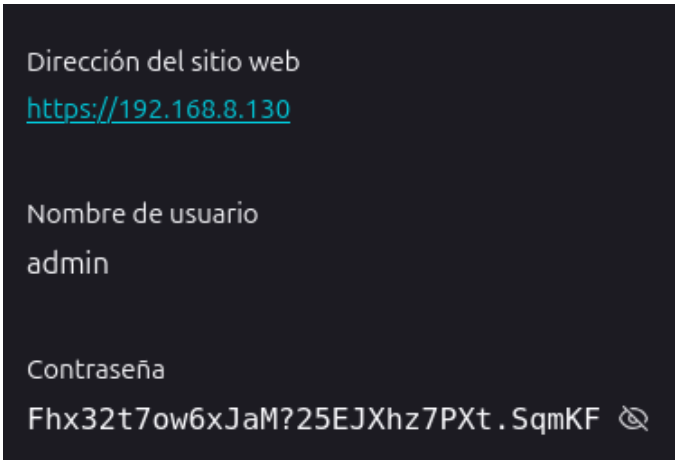


Figura 28. Credenciales del Servidor Wazuh

Paso 2: Verificación del Acceso Web

Desde el navegador de tu computadora física o de la Mini PC, ingresar a la dirección IP asignada al servidor: **https://192.168.8.130**

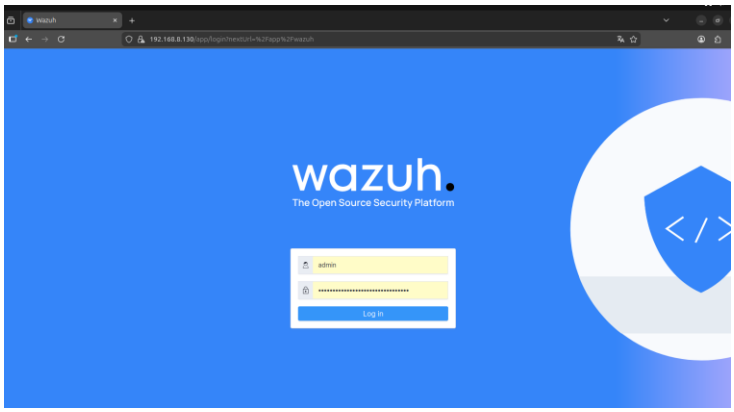


Figura 29. Acceso Web del servidor Wazuh

Tras aceptar el certificado de seguridad, se debe visualizar la interfaz de inicio de sesión de Wazuh.

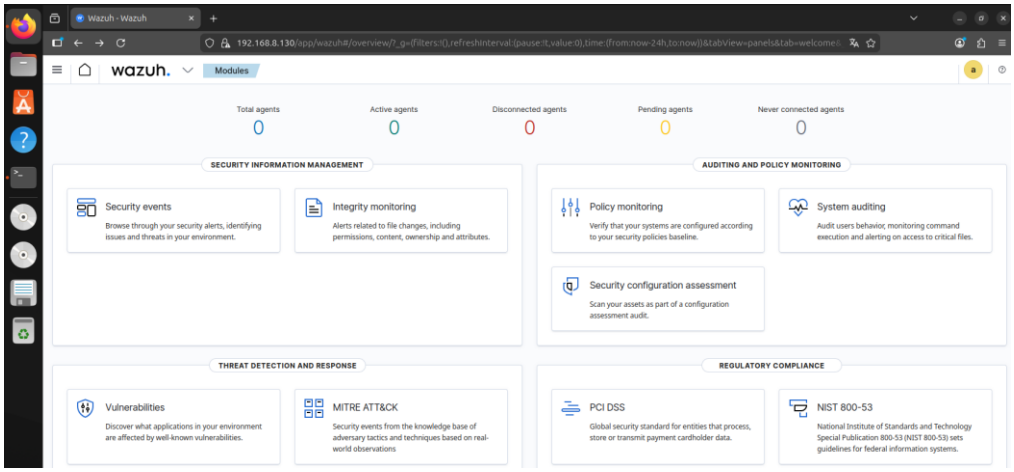


Figura 30. Página de inicio del servidor IDS Wazuh

Paso 3: Generación del Comando de Despliegue para PC1

Para poner la PC1 "en línea", se utiliza el asistente del Dashboard:

Ir a la sección "Agents" y seleccionar "Deploy new agent".

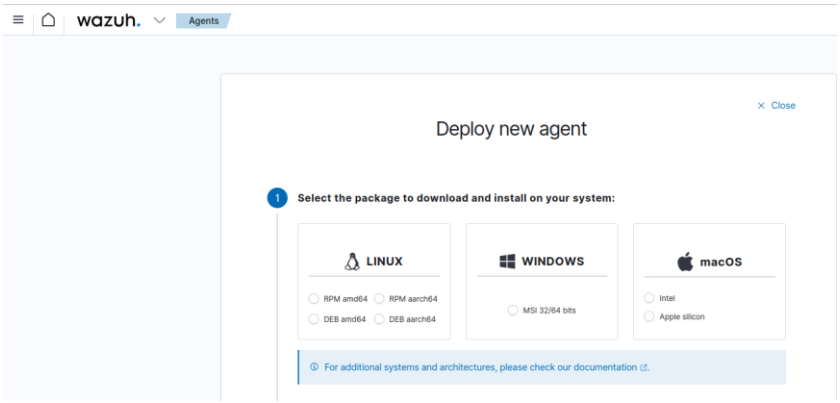


Figura 31. Creación de un nuevo agente

Seleccione el sistema operativo de la maquina virtual PC1 en este caso es Linux.

Escribir la IP de la maquina en la cual corre el servicio de Wazuh.

Copiar el comando generado automáticamente por el servidor.

Figura 32. Comandos para la instalación de su agente

Paso 4: Instalación del Agente en PC1

Acceder a la máquina PC1 y ejecutar el comando generado en el paso anterior:

Terminal:

sudo systemctl daemon-reload

sudo systemctl enable wazuh-agent

sudo systemctl start wazuh-agent

Paso 5: Validación de Conexión hacia el servidor mediante un ping

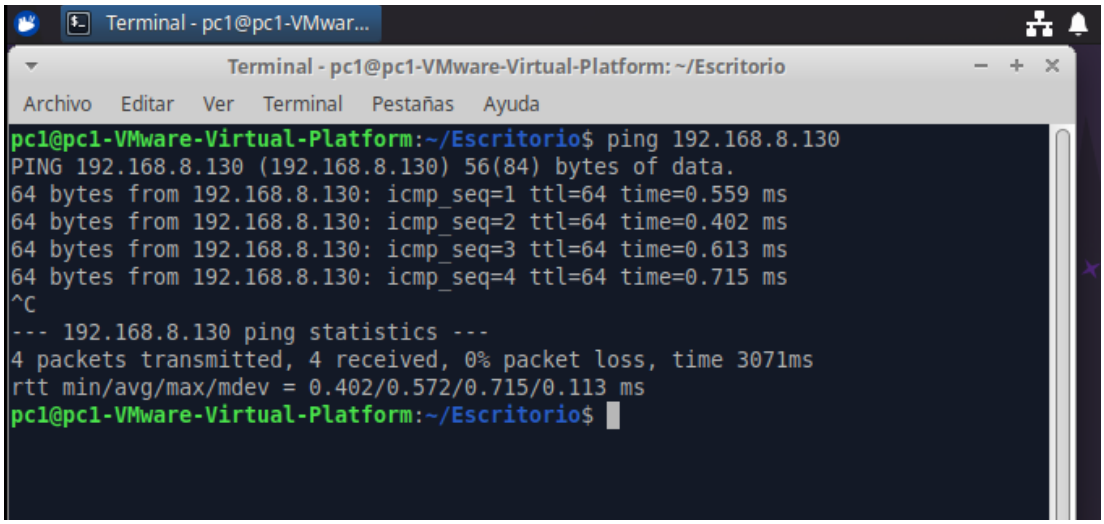


Figura 33. Verificación de interconexión de red

Paso 6: Revisión del Dashboard del agente

Regresar al Dashboard de Wazuh y verificar la lista de agentes.

Navegar a Agents.

Comprobar que la PC1 aparezca con un indicador de color verde diciendo active.

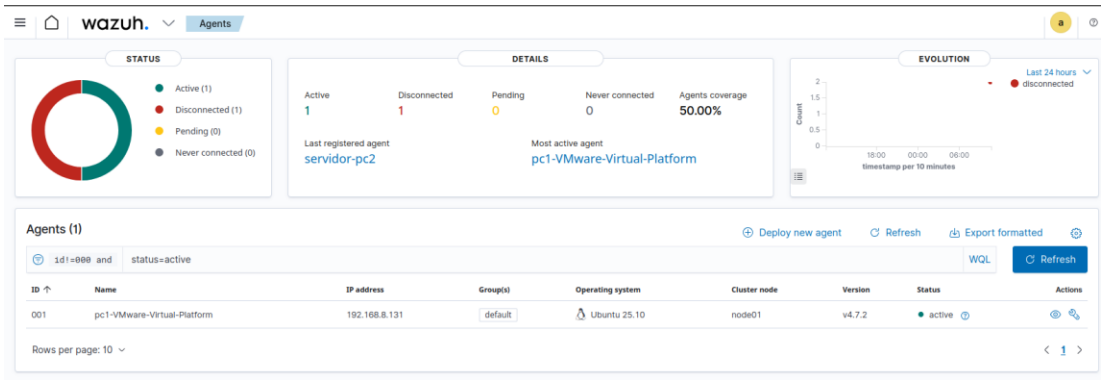


Figura 34. Visualización de Agente activo

Hacer clic en la PC1 para verificar que este operando normalmente y registre todo evento en su Dashboard.

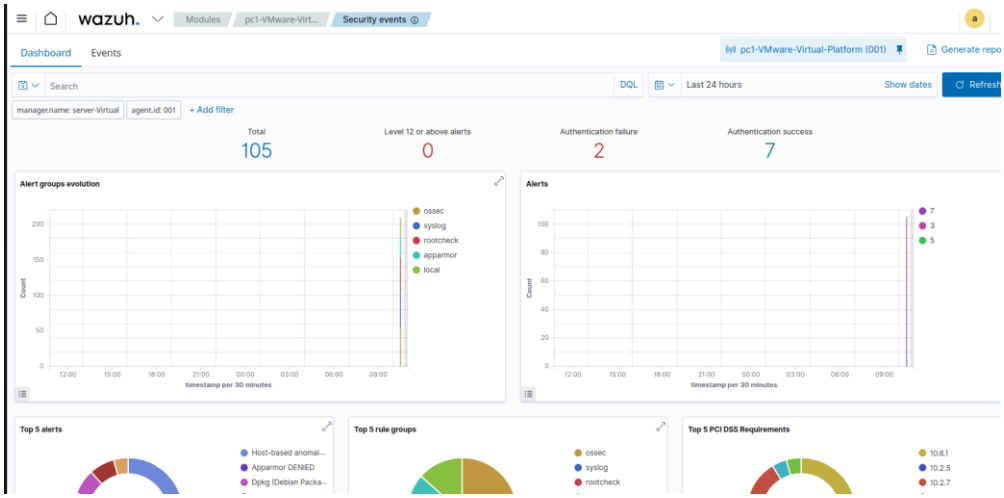


Figura 35. Consola de PC1

Arquitectura del sistema

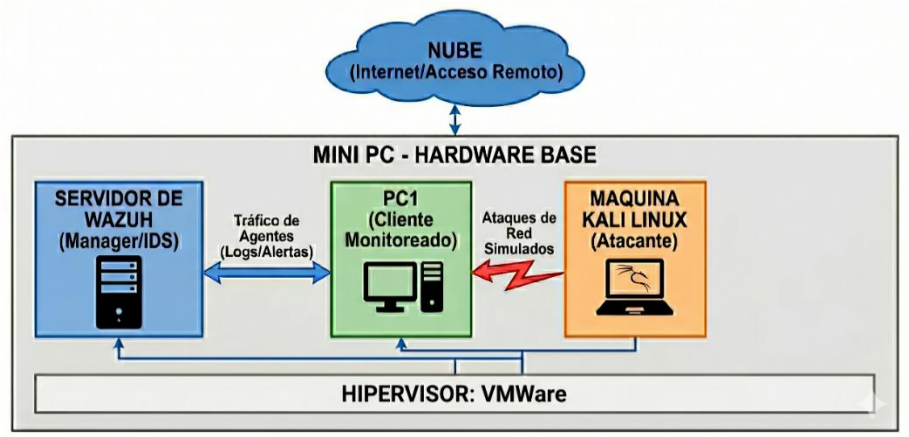


Figura 36. Diagrama de red del Laboratorio N° 02

3.2.3 Práctica de Laboratorio N° 03

Tema: Creación de una máquina virtual y la instalación de un servidor MOODLE para su respectiva conexión al IDS Wazuh

1. Objetivos

Objetivo General: Implementar un entorno de servidor web seguro mediante la virtualización de una plataforma de aprendizaje virtual (Moodle) basada en LAMP sobre Ubuntu 24.04, integrándola activamente a un Sistema de Detección de Intrusos (IDS) centralizado mediante el despliegue del agente Wazuh, con el fin de monitorear eventos y mitigar riesgos informáticos.

Objetivos Específicos

- Crear una nueva máquina virtual con el hardware optimo para la instalación y configuración del sistema operativo Ubuntu y el entorno de servicios LAMP.
- Ejecutar el entorno de la plataforma MOODLE a nivel de servidor, endurecimiento del sistema de archivos mediante escalabilidad de privilegios (CHMOD).
- Implementar la arquitectura de MOODLE integrándola al servidor de Wazuh para la validación, centralización de logs y visualización de las diferentes alertas.

2. Desarrollo de la practica

Paso 1: Creación de máquina virtual

Entrar al asistente de nueva máquina virtual y dar clic en instalación típica

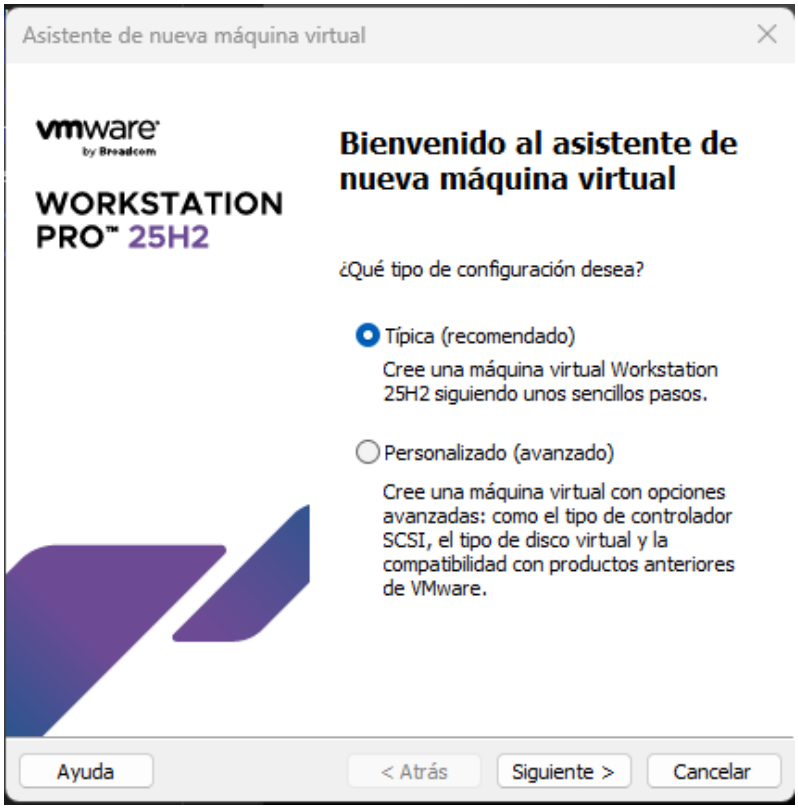


Figura 37. Creación de máquina virtual

Paso2: Elección del archivo .iso del sistema operativo

Elegir el iso para la instalación del sistema operativo ubuntu-24.04.3-desktop-amd64.iso

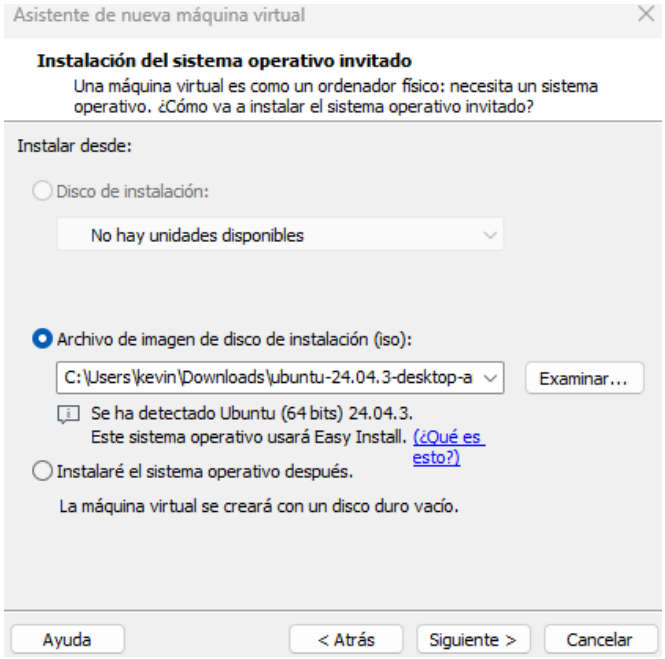


Figura 38.Elección del archivo .iso del sistema operativo

Paso 3: Personalización de la máquina virtual

Seguido personalizar la máquina virtual con los campos mostrados

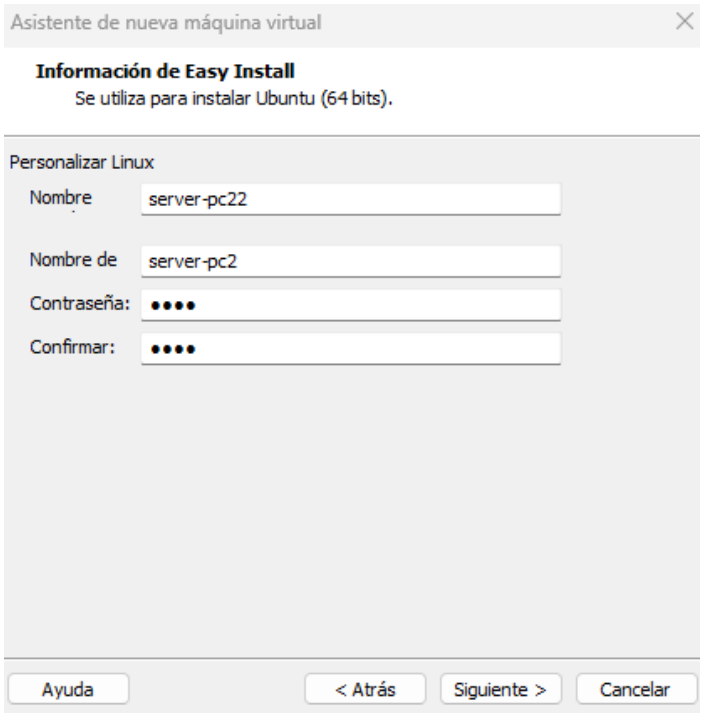


Figura 39. Personalización de la máquina virtual

Paso 4: Asignación del nombre de la máquina virtual

Llenar el nombre de la máquina virtual y la ubicación de esta y luego dar clic en siguiente:

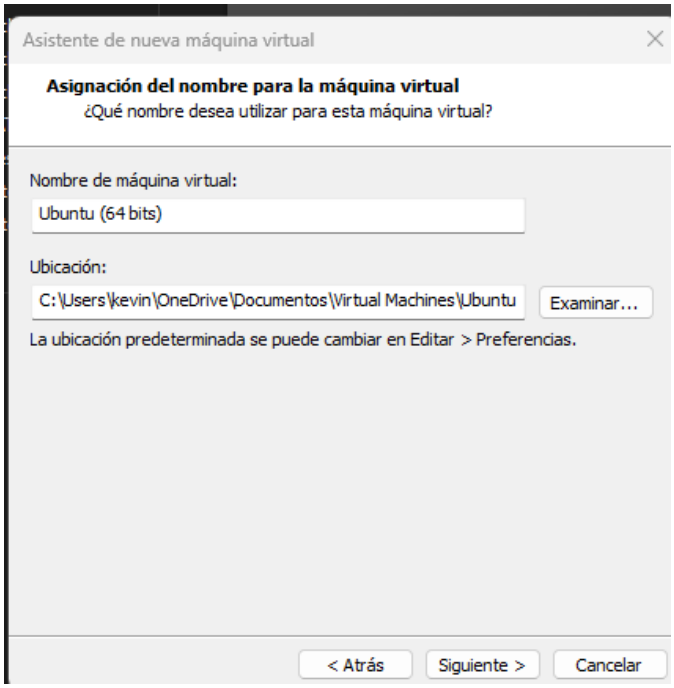


Figura 40. Asignación del nombre de la máquina virtual

Paso 5: Asignación del espacio de disco SSD

Dar clic en el tamaño de disco de la maquina server-pc2 a 60 Gb

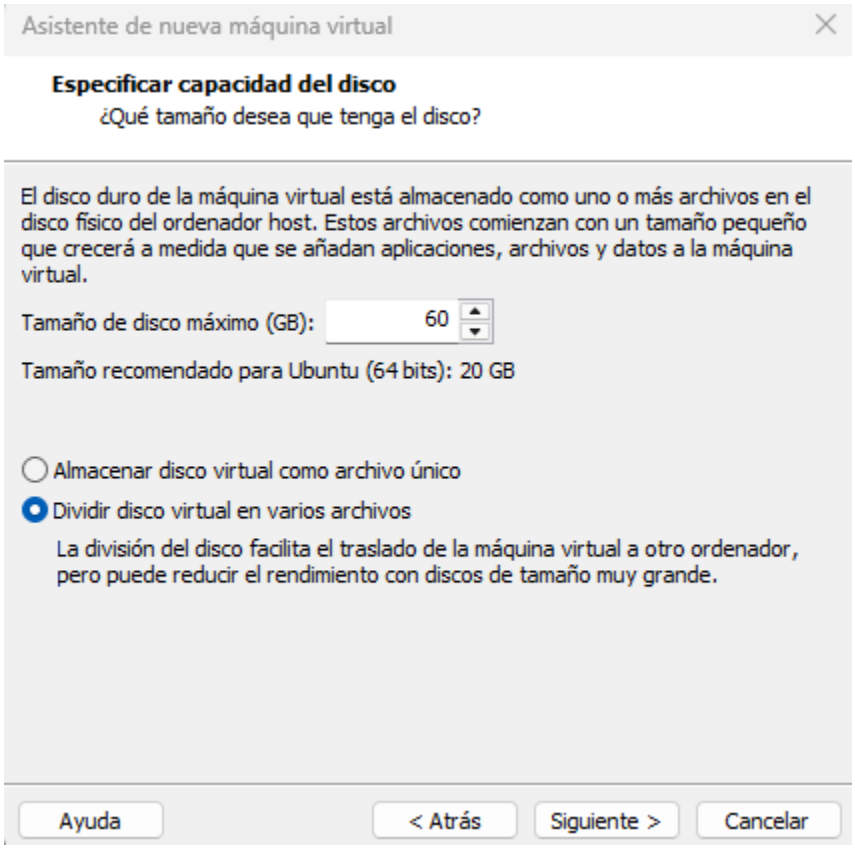


Figura 41. Asignación del espacio de disco SSD

Paso 6: Verificación de las características de la máquina virtual a crear

Luego verificar que la red sea de tipo NAT tenga dos núcleos y dos procesadores para mayor eficiencia del servicio

Resumen de máquina virtual server-pc2

Memoria ROM: 60 GB

Memoria RAM: 4096 MB

Núcleos por procesador: 2

Numero de procesadores: 2

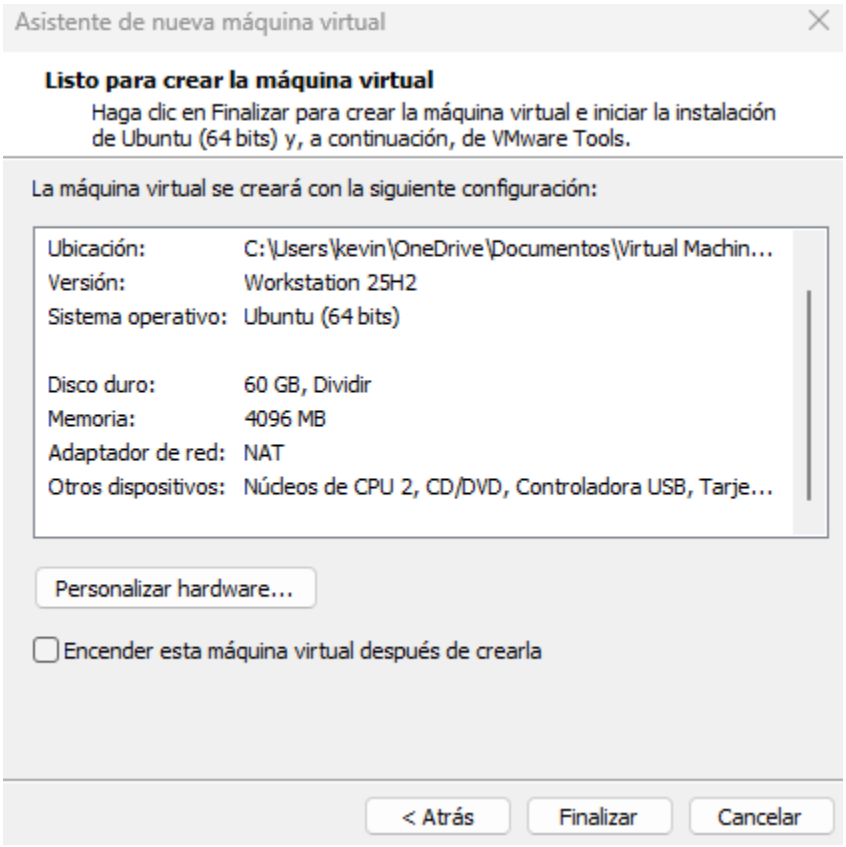


Figura 42. Verificación de las características de la máquina virtual a crear

Paso 7: Instalación del servicio Moodle

Actualizar el sistema con el comando :

sudo apt update && sudo apt upgrade -y

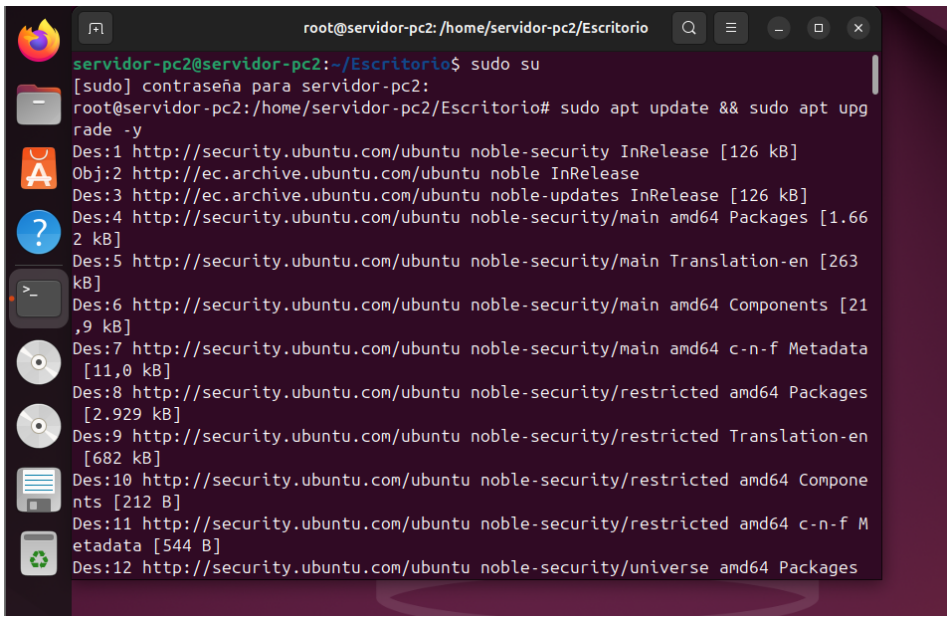


Figura 43. Actualización del sistema Ubuntu

Paso 8: Instalar apache

Instalar apache 2 con el siguiente comando:

Terminal:

sudo apt install apache2 -y

Figura 44. Instalación de apache

Paso 9: Estado de la instalación y servicio de apache 2

Revisar el estado de apache 2

systemctl status apache 2

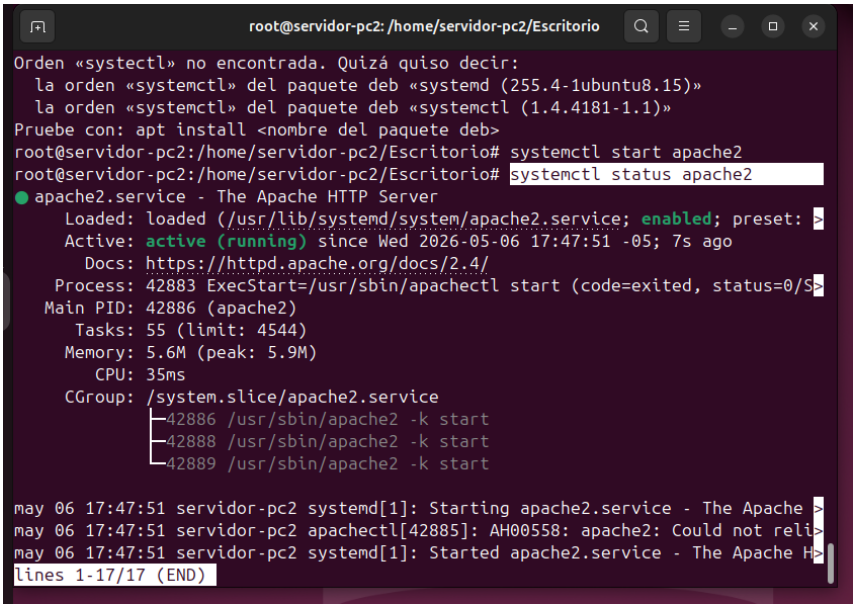


Figura 45. Estado de apache por comando

Paso 10: instalación de PHP 8.3 y las extensiones que Moodle necesita para no dar errores

Ejecutar este comando para instalar todo el grupo de una vez:

Terminal:

sudo apt install php8.3 libapache2-mod-php8.3 php8.3-mysql php8.3-common php8.3-gd php8.3-intl php8.3-xml php8.3-curl php8.3-zip php8.3-mbstring php8.3-soap php8.3-opcache -y

Después de instalar PHP:

Es importante reiniciar Apache para que reconozca el nuevo módulo de PHP:

Terminal:

sudo systemctl restart apache2

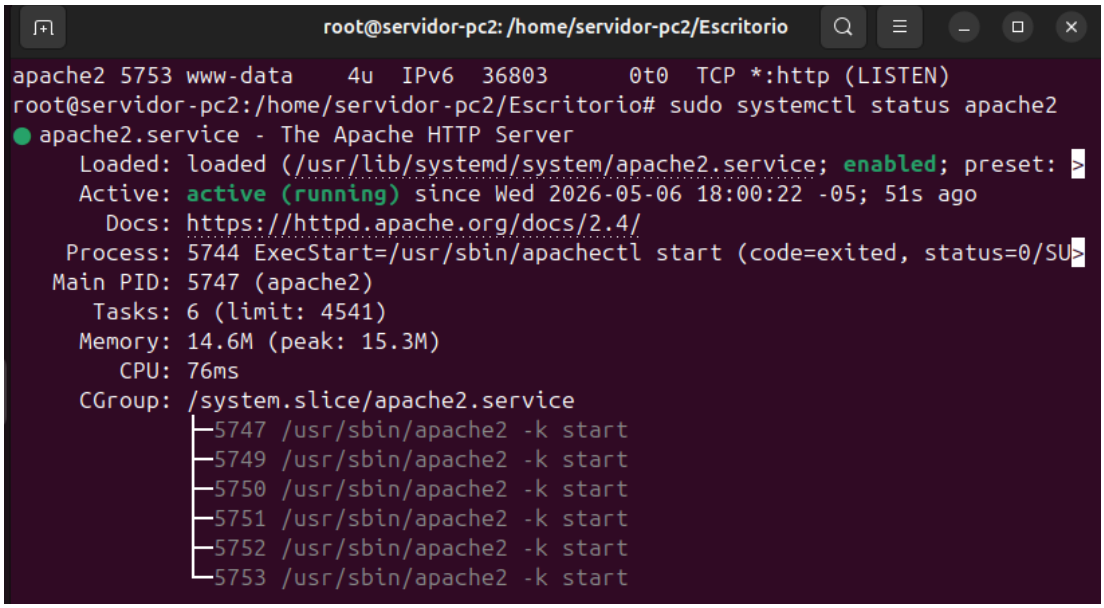


Figura 46. Verificación de servicio apache

Paso 11: Instalación de MariaDB

Ahora se necesita instalar el motor donde Moodle guardará la información. La base de datos con MariaDB:

Terminal:

sudo apt install mariadb-server -y

```
root@servidor-pc2:/home/servidor-pc2/Escritorio# sudo apt install mariadb-server
 -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
mariadb-server ya está en su versión más reciente (1:10.11.14-0ubuntu0.24.04.1).
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  libllvm19 libwoff1
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 8 no actualizados.
root@servidor-pc2:/home/servidor-pc2/Escritorio#
```

Figura 47. Instalación del servicio MariaDB

Paso12: Verificación del servicio de MariaDB

Terminal:

sudo systemctl status mariadb

```
● mariadb.service - MariaDB 10.11.14 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
     Active: active (running) since Wed 2026-05-06 17:58:18 -05; 4min 37s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 1524 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var>
    Process: 1613 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_ST>
    Process: 1721 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && >
    Process: 1843 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_S>
    Process: 1847 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/>
   Main PID: 1819 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 29973)
     Memory: 130.8M (peak: 134.2M)
        CPU: 1.171s
     CGroup: /system.slice/mariadb.service
             └─1819 /usr/sbin/mariadbd
```

Figura 48. Estado del servicio de MariaDB

Paso 13: Ingreso a consola de MariaDB

Entra a la consola de MariaDB

Terminal:

sudo mysql

```
root@servidor-pc2:/home/servidor-pc2/Escritorio# sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.14-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

Figura 49. Consola de MariaDB

Paso 14: Creación de la base de datos y el usuario

Se ingresa a MariaDB y se crea la base de datos y el usuario.

CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

CREATE USER 'moodleuser'@'localhost' IDENTIFIED BY 'Pass2026test';

GRANT ALL PRIVILEGES ON moodle.* TO 'moodleuser'@'localhost';

FLUSH PRIVILEGES;

EXIT;

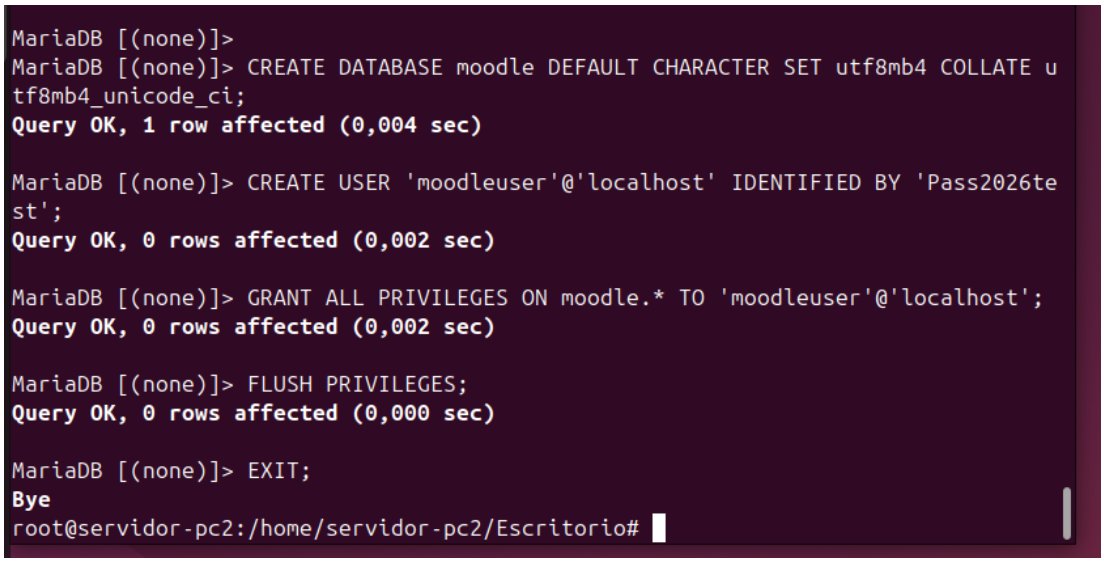


Figura 50. Creación de la base de datos

Paso 15: Descarga del servicio de Moodle

Descargar la última versión estable (4.4 o la más reciente)

Terminal:

sudo wget **https://download.moodle.org/download.php/direct/stable404/moodle-latest-404.tgz**

Luego descomprimir todo el .tgz con el comando

Terminal:

sudo tar -zxvf moodle-latest-404.tgz

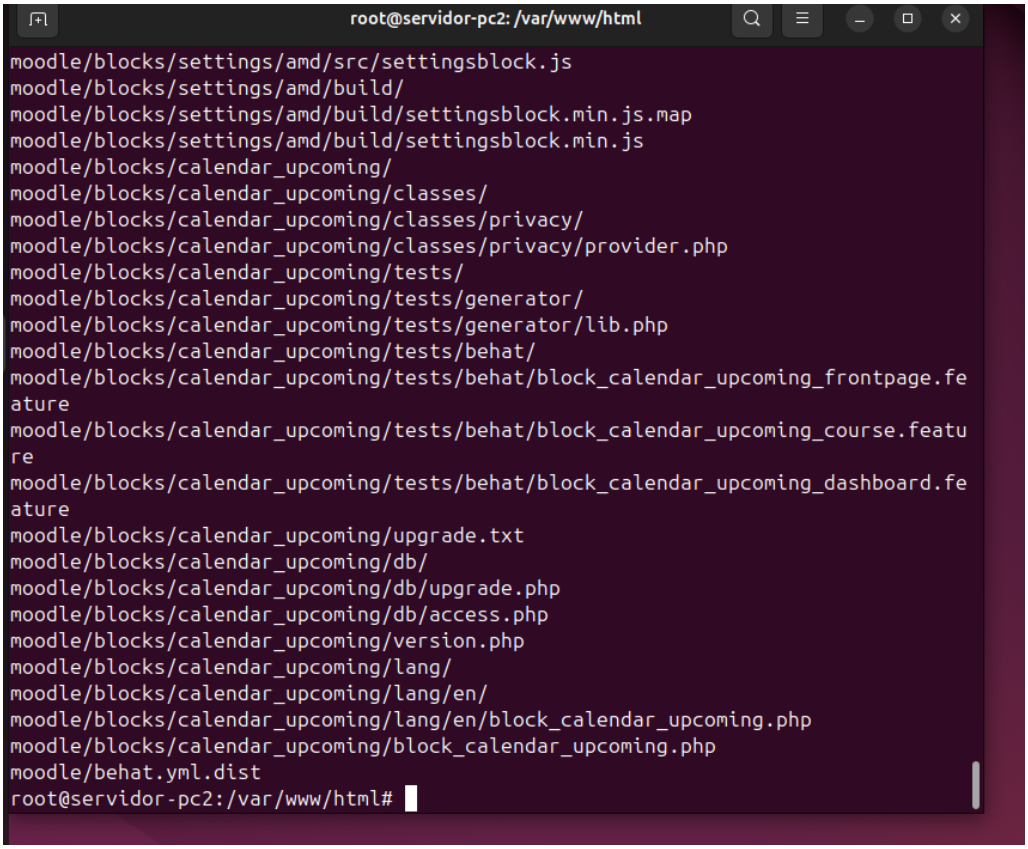


Figura 51. Descarga de paquetes para Moodle

Paso 16: Permisos de carpetas para el correcto funcionamiento

Moodle necesita una carpeta de datos segura y permisos correctos en la web por lo tanto crear carpeta de datos fuera de la carpeta pública por seguridad.

Terminal:

sudo mkdir -p /var/www/moodledata

sudo chown -R www-data:www-data /var/www/moodledata

sudo chmod -R 770 /var/www/moodledata

Dar permisos a la carpeta del código

Terminal:

sudo chown -R www-data:www-data /var/www/html/moodle

sudo chmod -R 755 /var/www/html/moodle

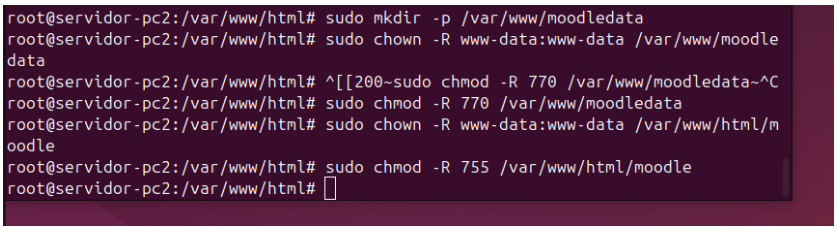


Figura 52. Permisos para la web de Moodle

Paso 17: Instalación del servicio de Moodle

Abrir el navegador y entrar con la ip o el siguiente enlace

http://localhost/moodle

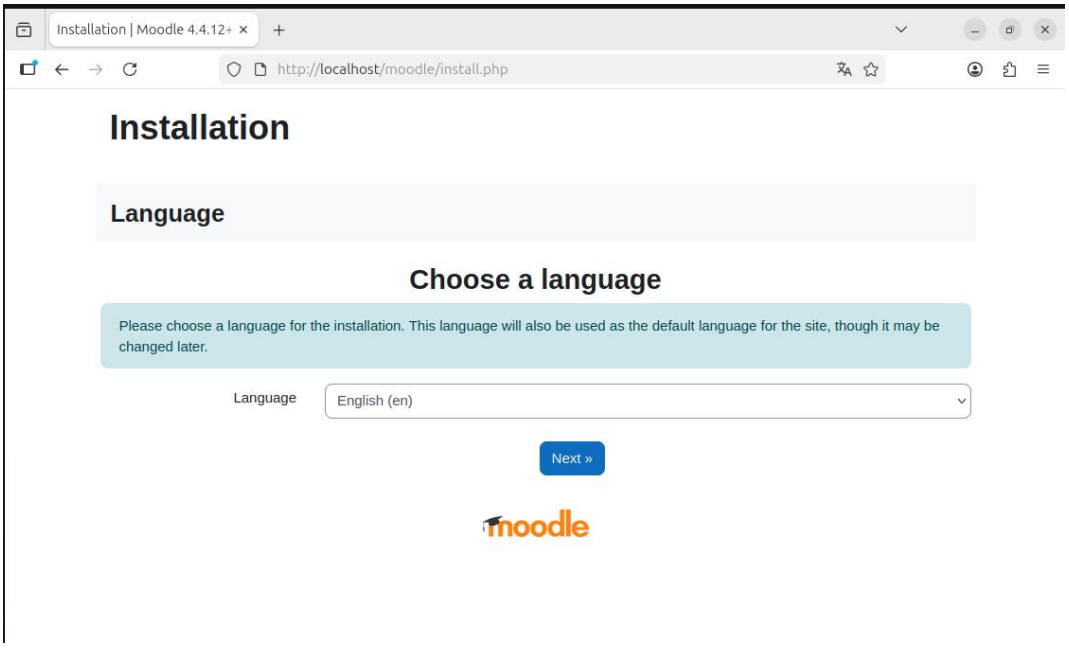


Figura 53. Instalación de Moodle

Paso 18: Ruta de instalación de Moodle

Luego de escoger el idioma español dar clic en siguiente Comprobar todas las rutas de instalación y dar siguiente

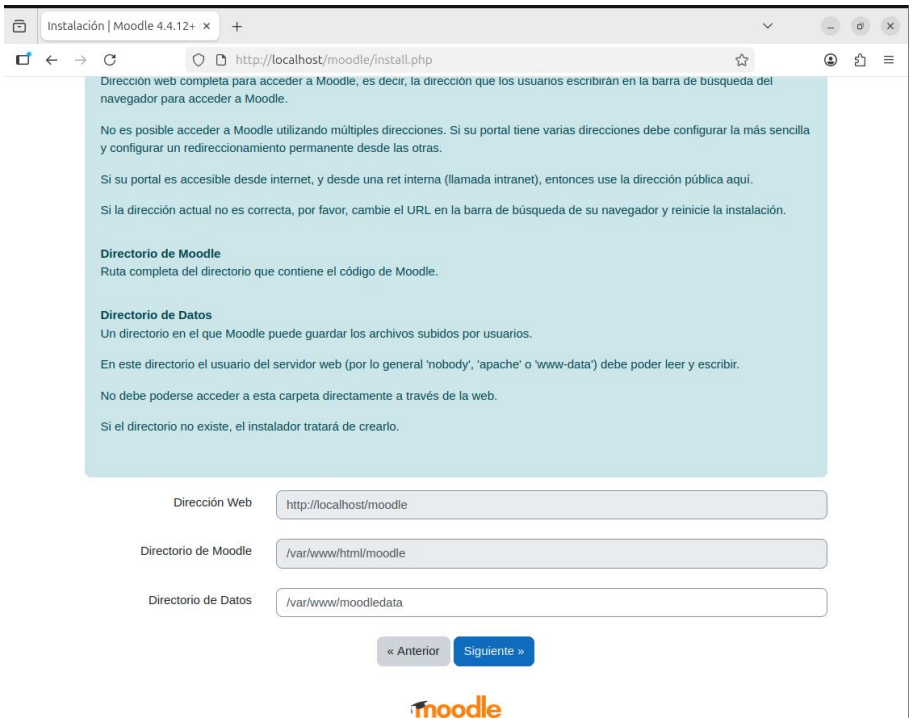


Figura 54. Rutas de instalación de Moodle

Paso 19: Instalación de la base de datos

Elegir el tipo de controlador de base de datos “MariaDB (nativo/mariadb)” y dar clic en siguiente

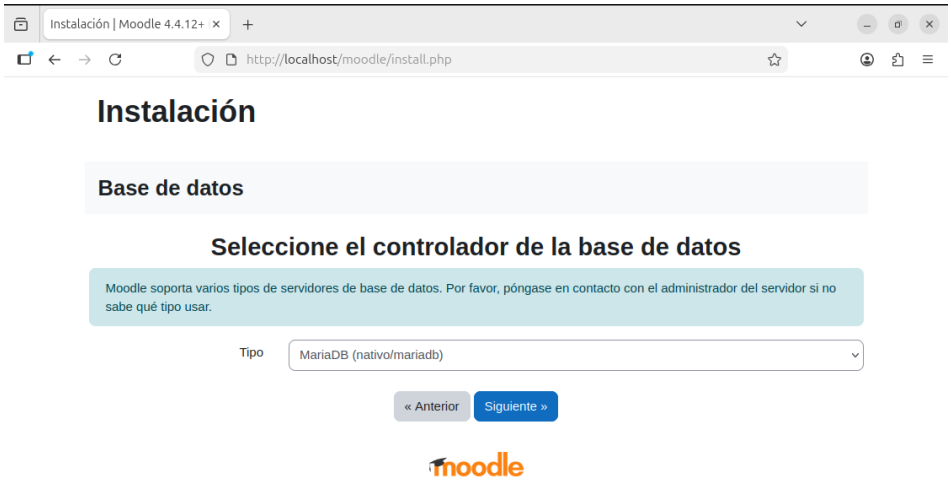


Figura 55. Instalación de la base de datos

Paso 20: Datos de instalación de Moodle

Llenar los campos de usuario y contraseña definidos en MariaDB

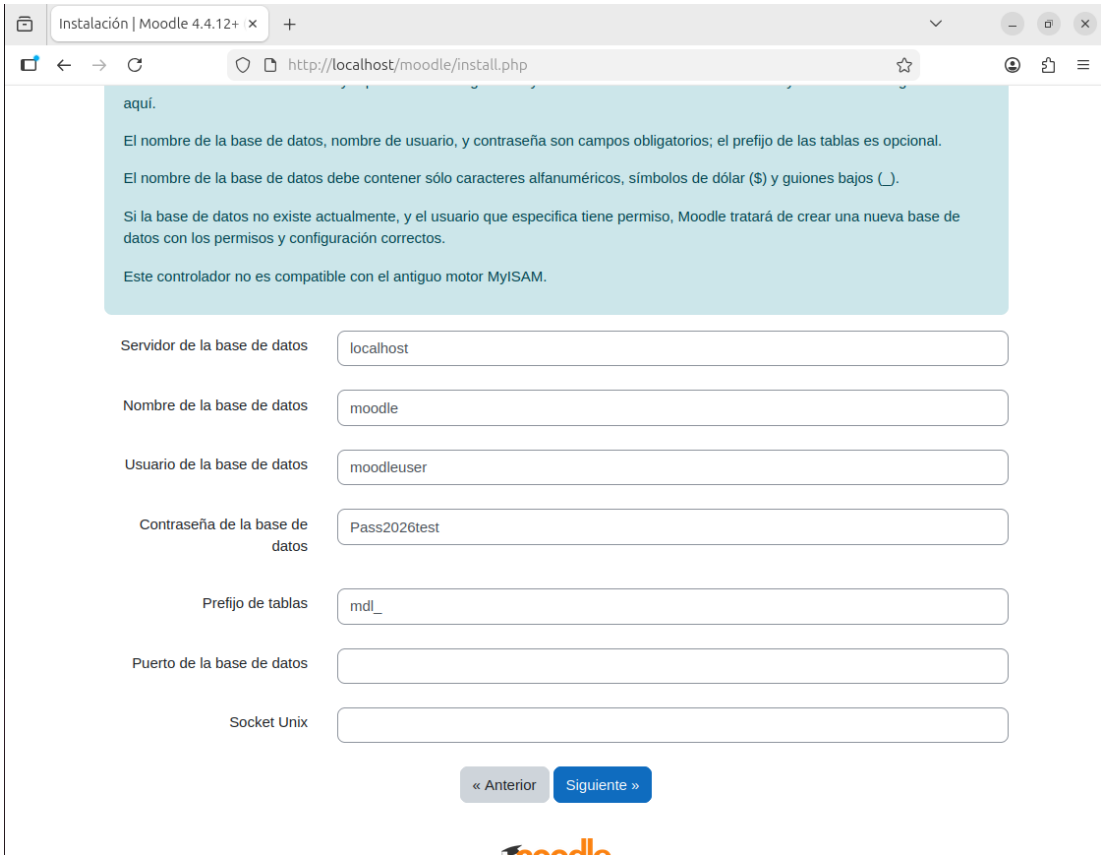


Figura 56. Datos de instalación de Moodle

Paso 21: Términos y condiciones de instalación de Moodle

Dar clic en continuar para su instalación

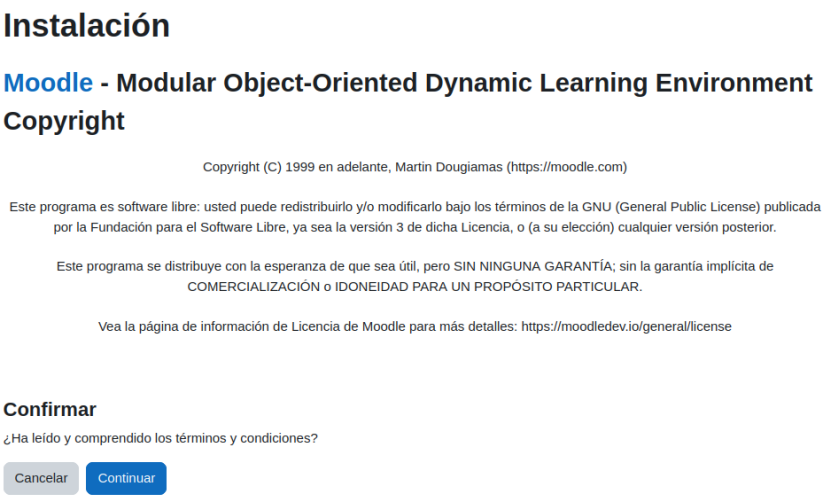


Figura 57.Términos y condiciones de instalación de Moodle

Paso 22: Dependencias de Moodle

Revisar que todas las extensiones estén en ver y OK

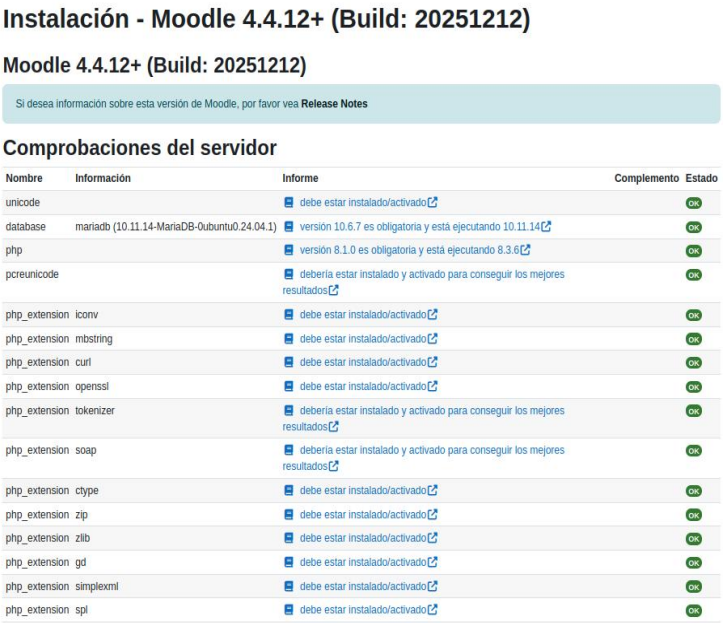


Figura 58. Dependencias de Moodle

Paso 23: Problemas de entorno dentro de la instalación de Moodle

Revisar las otras comprobaciones y proceder a resolverlas para la instalación correcta de Moodle

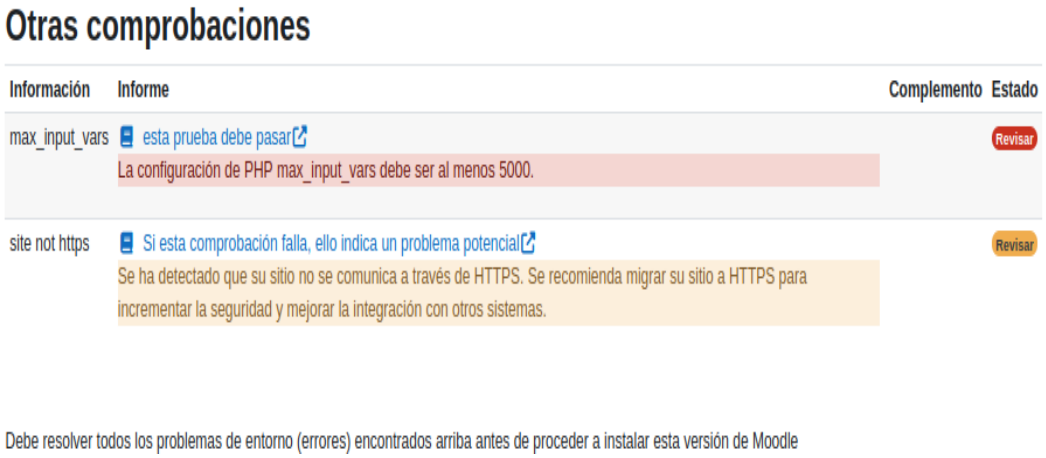


Figura 59. Problemas de entorno dentro de la instalación de Moodle

Paso 24: Ejecución de php.ini

Ejecutar en una nueva ventana sudo nano /etc/php/8.3/apache2/php.ini

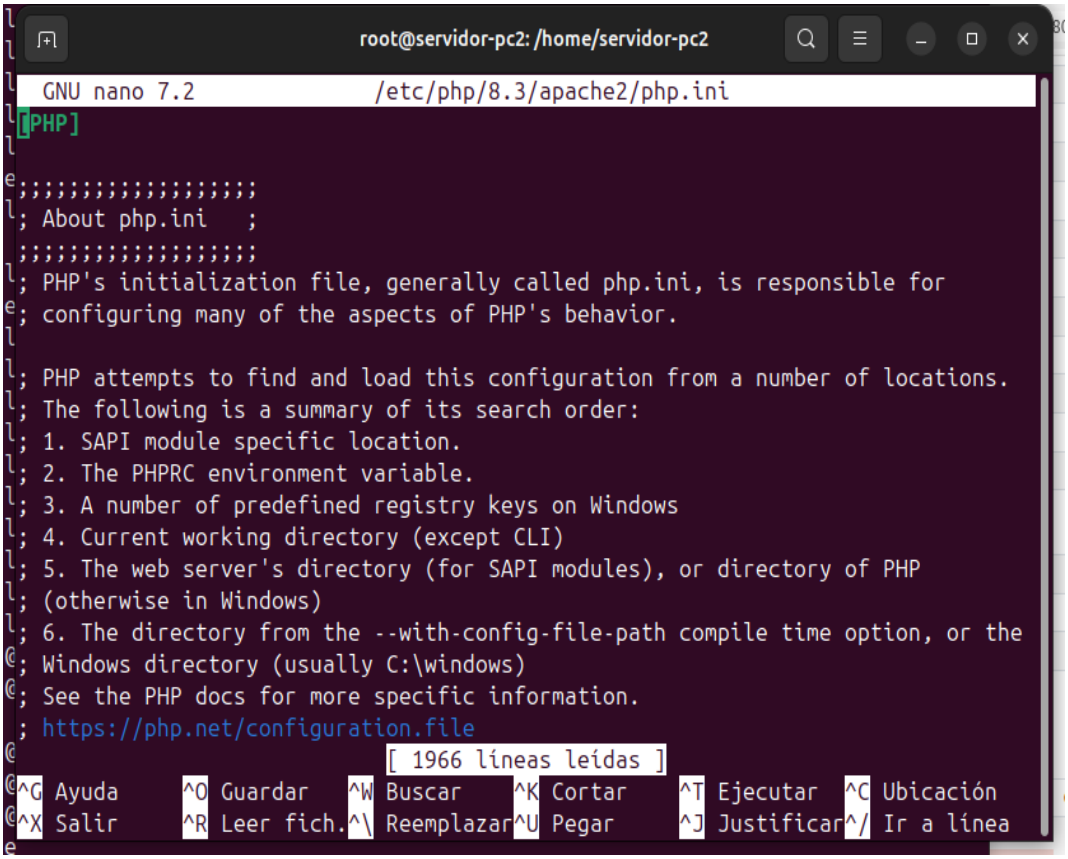


Figura 60. Ejecución de php.ini

Paso 25: Editar php.ini

Buscar max_input_vars y borrar el punto y coma inicial además de cambiar su valor a 5000 guardar y cerrar el editor de texto

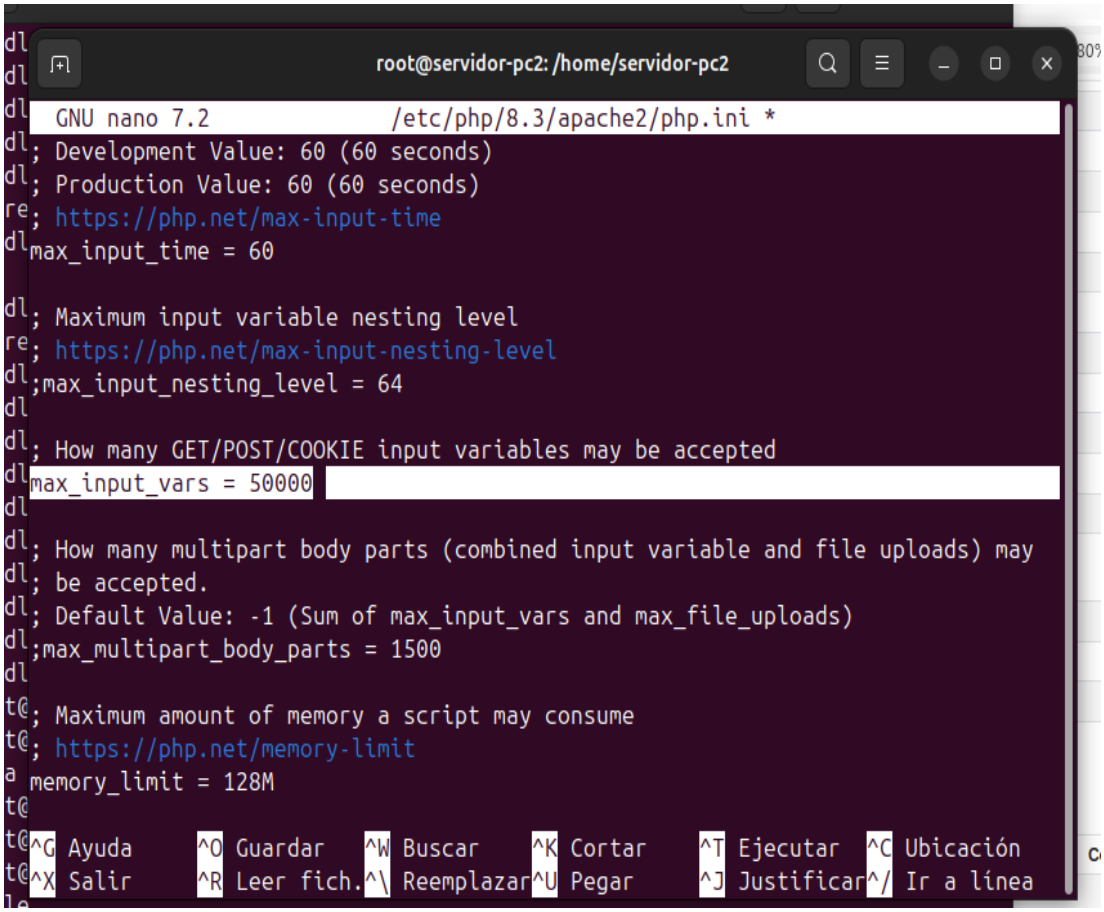


Figura 61. Editar php.ini

Paso 26: Reinicio de servicio

Reiniciar el servicio de apache2

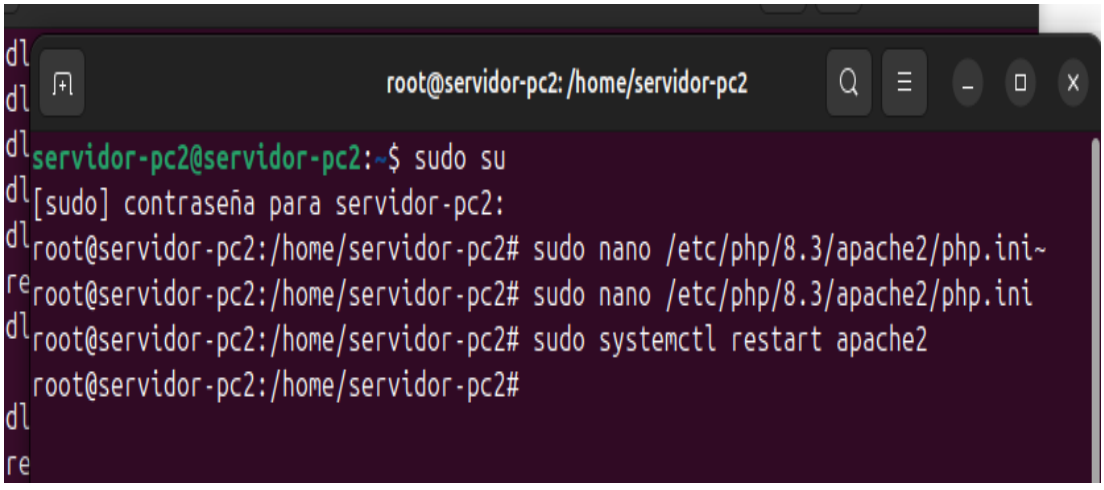


Figura 62. Reinicio de servicio

Paso 27: Instalación de dependencias

Esperar a que cargue e instale todas las dependencias de Moodle

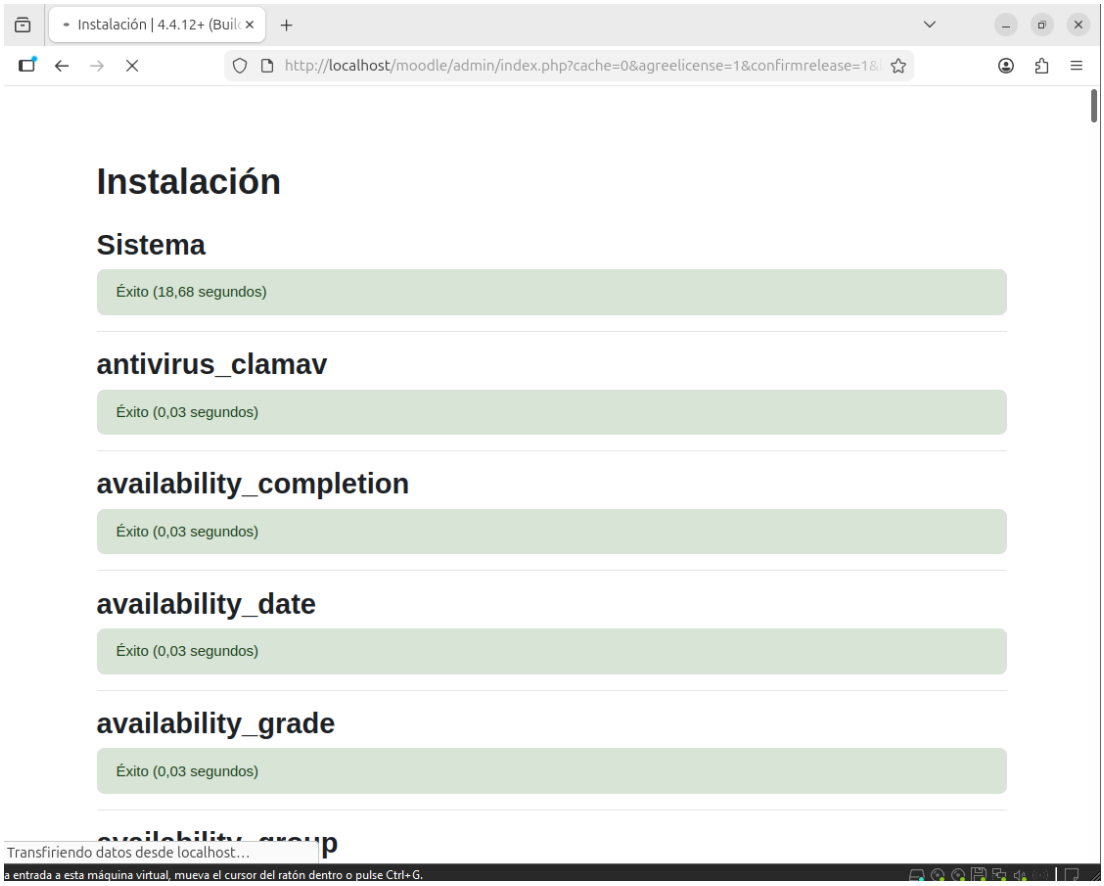


Figura 63. Instalación de dependencias

Paso 28: Finalización de resolución de errores

Una vez resuelto se mostrará el botón de continuar

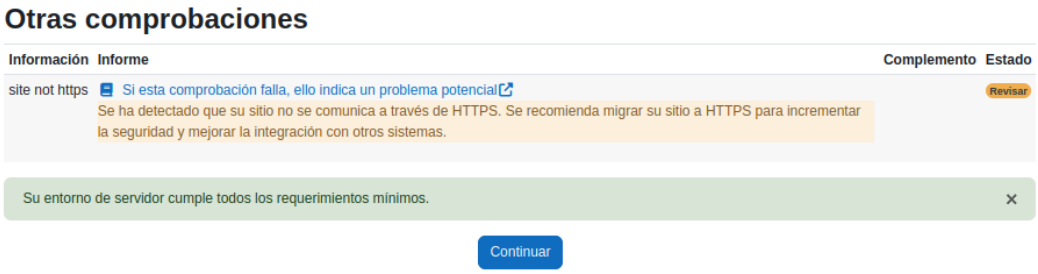


Figura 64. Finalización de resolución de errores

Paso 29: Ingreso de datos del servicio de Moodle

Llenar todos los datos y la contraseña en este caso para el laboratorio será: "Admin2026@@" se aconseja que la contraseña tenga mayúsculas, minúsculas, números y caracteres especiales.

Dar clic al final en "Actualizar información personal"

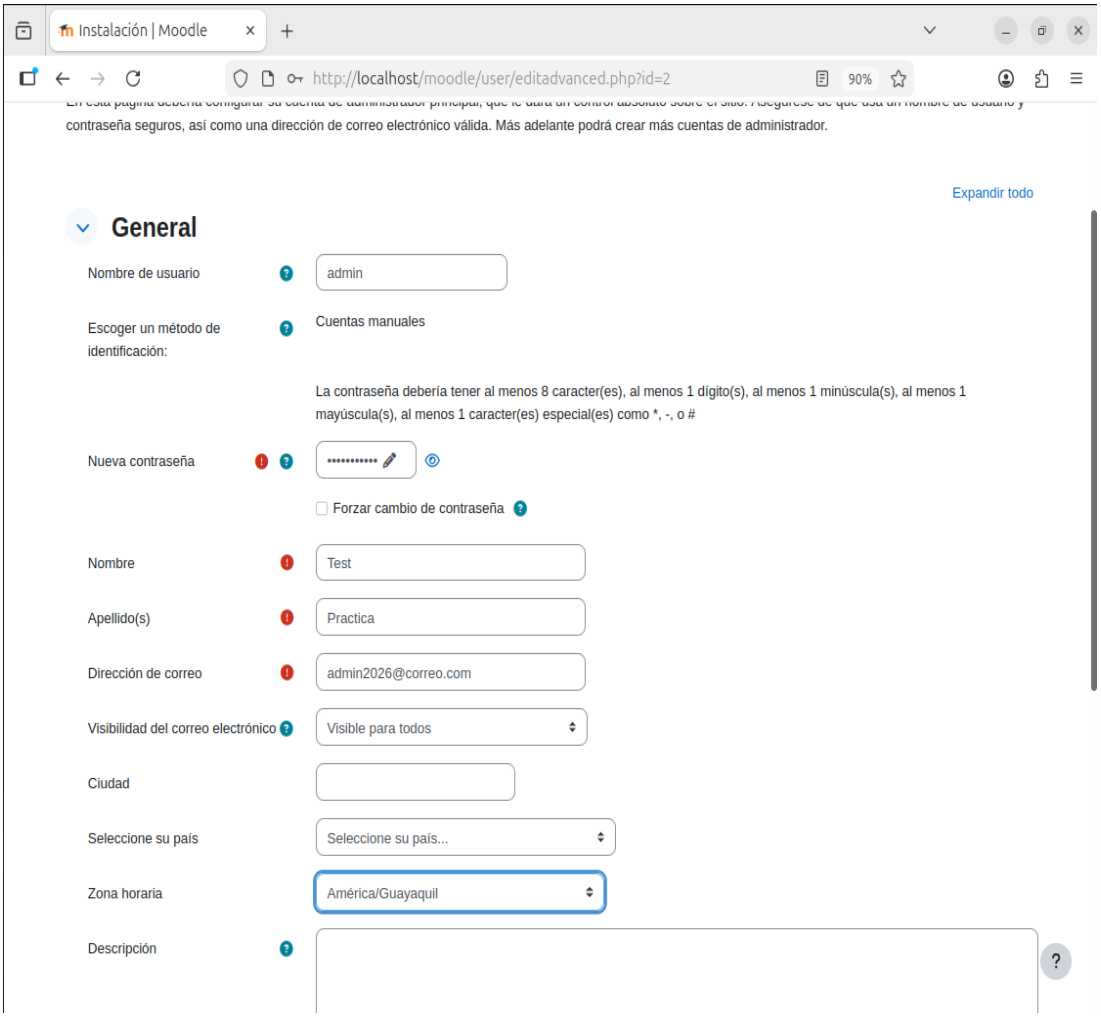


Figura 65. Ingreso de datos del servicio de Moodle

Paso 30: Creación de Laboratorio dentro del servicio de Moodle

Crear un laboratorio para mayor visibilidad del funcionamiento de Moodle

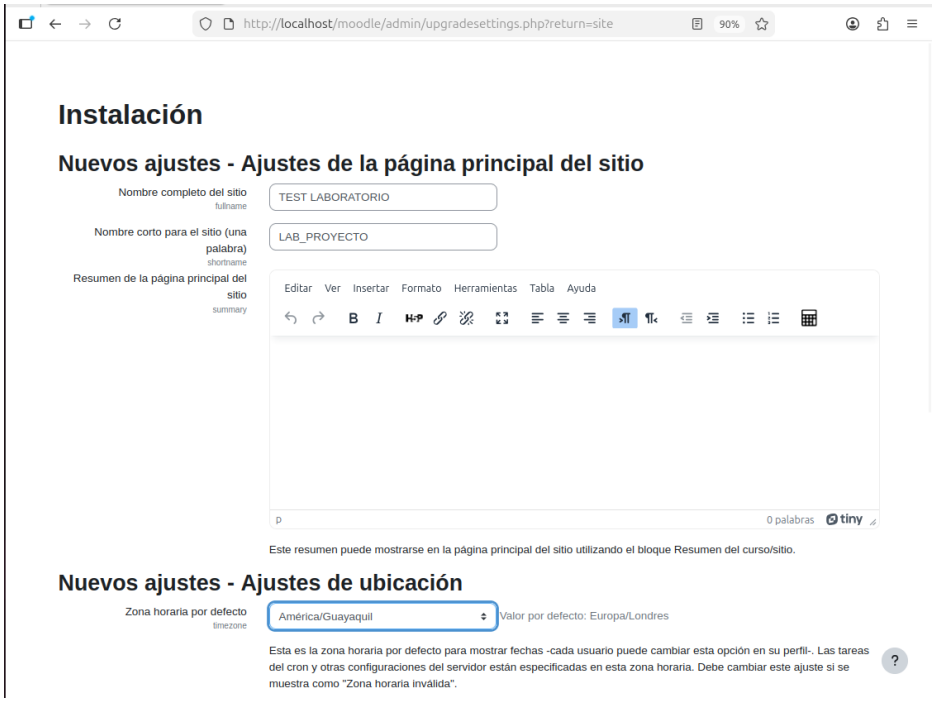


Figura 66. Creación de Laboratorio dentro del servicio de Moodle

Paso 31: Visualización de la página de Moodle

Una vez realizado el ajuste para la página principal se abra creado un Moodle idéntico al de las universidades en este caso como ejemplo siendo la máquina virtual su servidor.

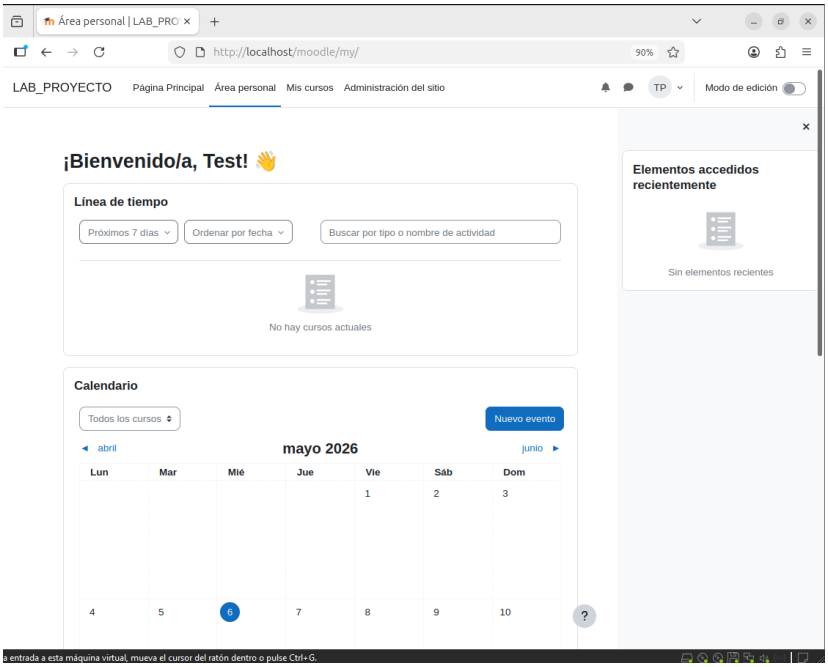


Figura 67. Visualización de la página de Moodle

Paso 32: Creación y conexión del agente de la máquina virtual "server-pc2" hacia el Dashboard de Wazuh

Desplegar un nuevo agente para la conexión al IDS basado en host integrado a una plataforma SIEM/XDR Wazuh

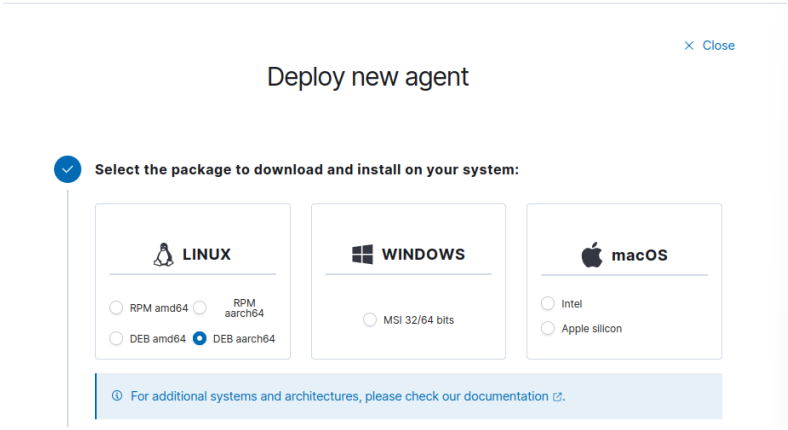


Figura 68. Creación y conexión del agente

Paso 33: Comando de instalación de agente

Copiar el comando y correrlo en la máquina que se ha instalado el servicio de Moodle

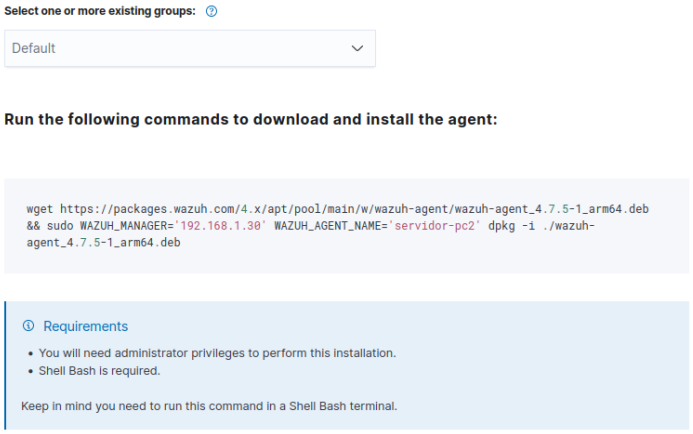


Figura 69. Comando de instalación de agente

Paso 34: Ejecución de comando en la maquina donde será el nuevo agente

Una vez copiado el código se ejecuta en la máquina de servidor-pc2

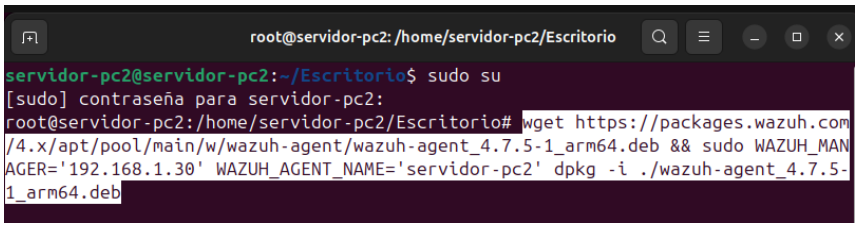


Figura 70. Ejecución de comando en la maquina donde será el nuevo agente

Paso 35: Reinicio de servicio de agente de Wazuh

Ahora se reinicia el servicio con los comandos:

Terminal:

sudo systemctl daemon-reload

sudo systemctl enable wazuh-agent

sudo systemctl start wazuh-agent

Paso 36: Verificación de actividad del servicio del agente Wazuh

Para su verificación que este corriendo el servicio escribimos el comando

Terminal:

sudo systemctl status wazuh-agent

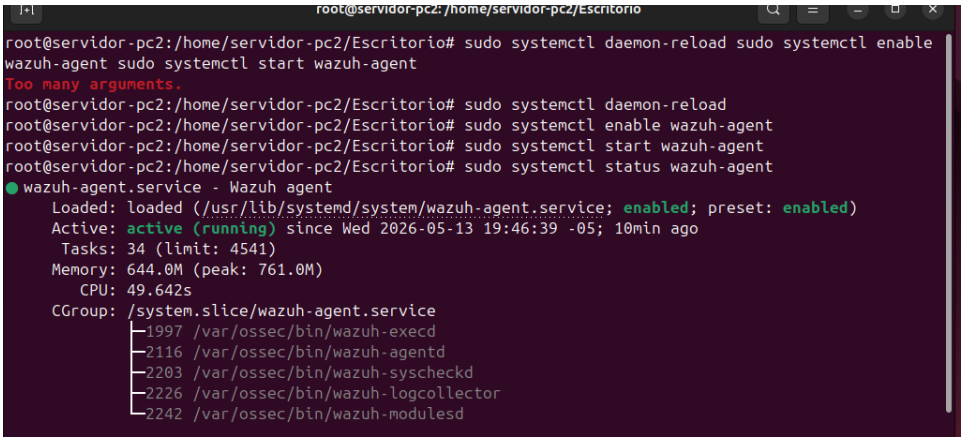


Figura 71. Verificación de actividad del servicio del agente Wazuh

Paso 37: Visualización de los agentes con estado activo en Wazuh

Ahora se asegura que el segundo agente del servidor pc este correctamente conectado y activo

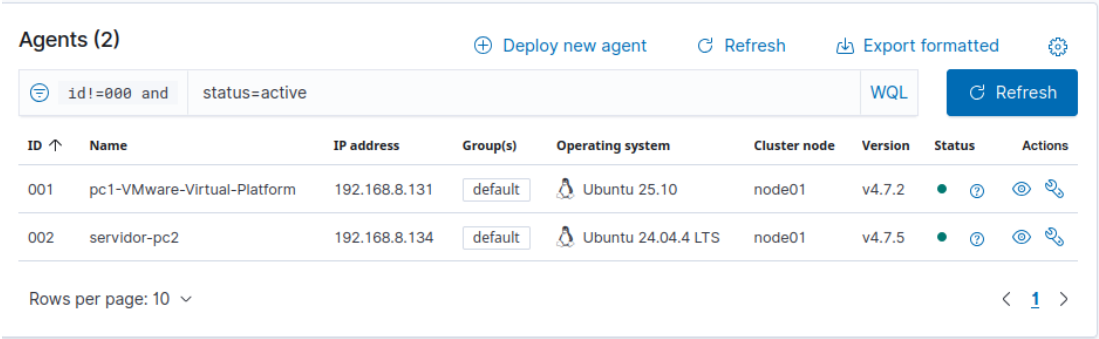


Figura 72. Visualización de los agentes con estado activo en Wazuh

Paso 38: Visualización de todas las alertas del equipo "Servidor-PC2"

Por último se visualiza los eventos de seguridad que tiene la maquina registrada

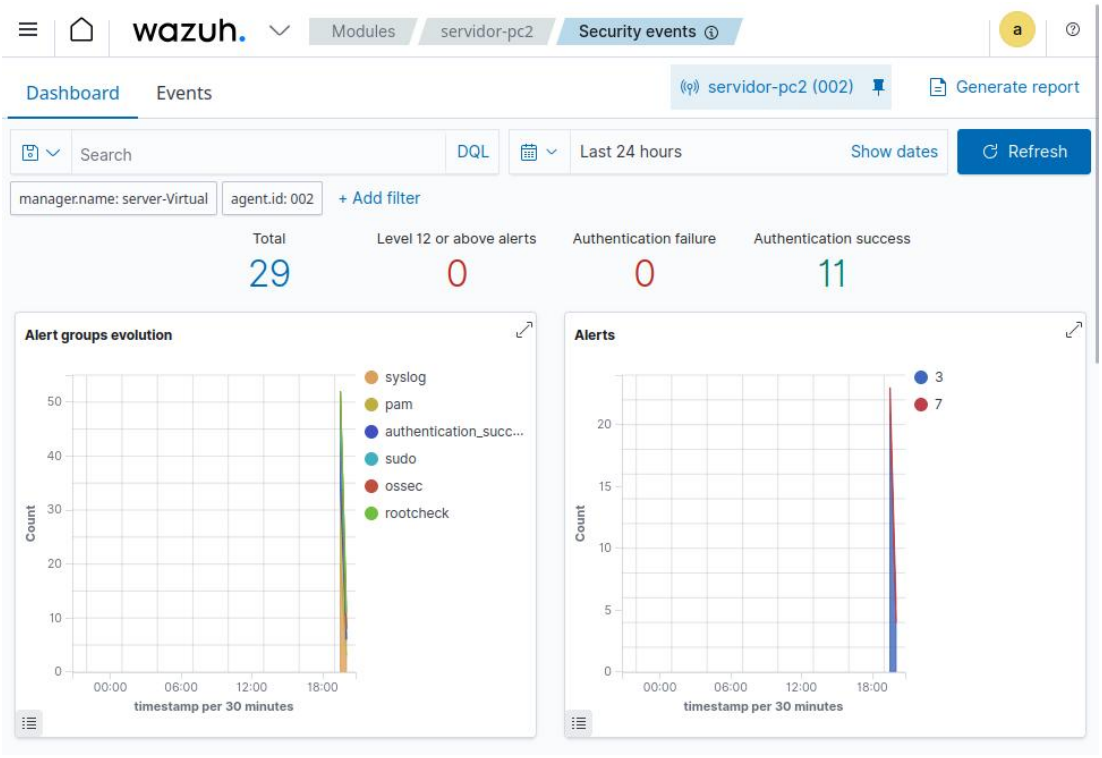


Figura 73. Visualización de todas las alertas del equipo "Servidor-PC2"

Arquitectura Final del laboratorio 03

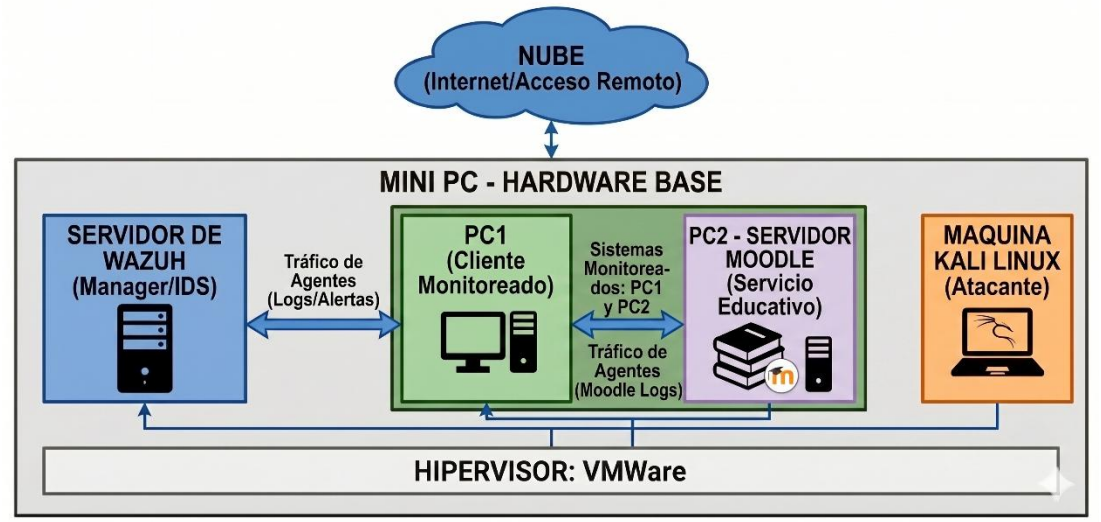


Figura 74. Arquitectura Final del laboratorio 03

3.2.4 Práctica de Laboratorio N° 04

Tema: Auditoría de Seguridad con Wazuh e Impacto de la Presencia de Agentes en la Detección de Intrusiones

1. Objetivos

Objetivo General: Demostrar el impacto de la telemetría en la visibilidad de incidentes dentro de un Centro de Operaciones de Seguridad (SOC), mediante un análisis comparativo de ataques en un sistema operativo con el agente de Wazuh activo e inactivo.

Objetivos Específicos

Evaluar la capacidad de respuesta y registro de eventos en el Dashboard de Wazuh ante vectores de ataque comunes (reconocimiento y fuerza bruta).

Simular un escenario de pérdida de visibilidad (sistema ciego) mediante la desactivación controlada del servicio del agente en el nodo objetivo.

Verificar la normalización de las alertas en el SIEM para asegurar que los eventos se registren en sus niveles base de severidad, evitando falsos positivos y ruido innecesario.

2. Recursos (Hardware y Software)

- Hardware: Mini PC (Servidor Host).
- Software: VMware Workstation Pro / Player.

Instancias Virtuales (Fase 1):

- Nodo Wazuh Manager (Ubuntu Server).
- Nodo Objetivo PC1 (Ubuntu 24.04).
- Nodo Atacante (Kali Linux).
- Herramientas de Auditoría: Nmap (Escaneo de puertos) e Hydra (Fuerza bruta).

3. Desarrollo de la Práctica

Paso 1: Configuración de la Línea Base y Desactivación del Agente (Escenario "Antes")

Con el propósito de simular un sistema desprovisto de auditoría activa de cara al SIEM, se accede a la terminal del Nodo Objetivo PC1 y se procede a detener de forma controlada el servicio del agente de seguridad mediante el comando:

Terminal:

sudo systemctl stop wazuh-agent

Posteriormente, se ingresa a la interfaz web del Manager para constatar que el estado del endpoint haya cambiado.

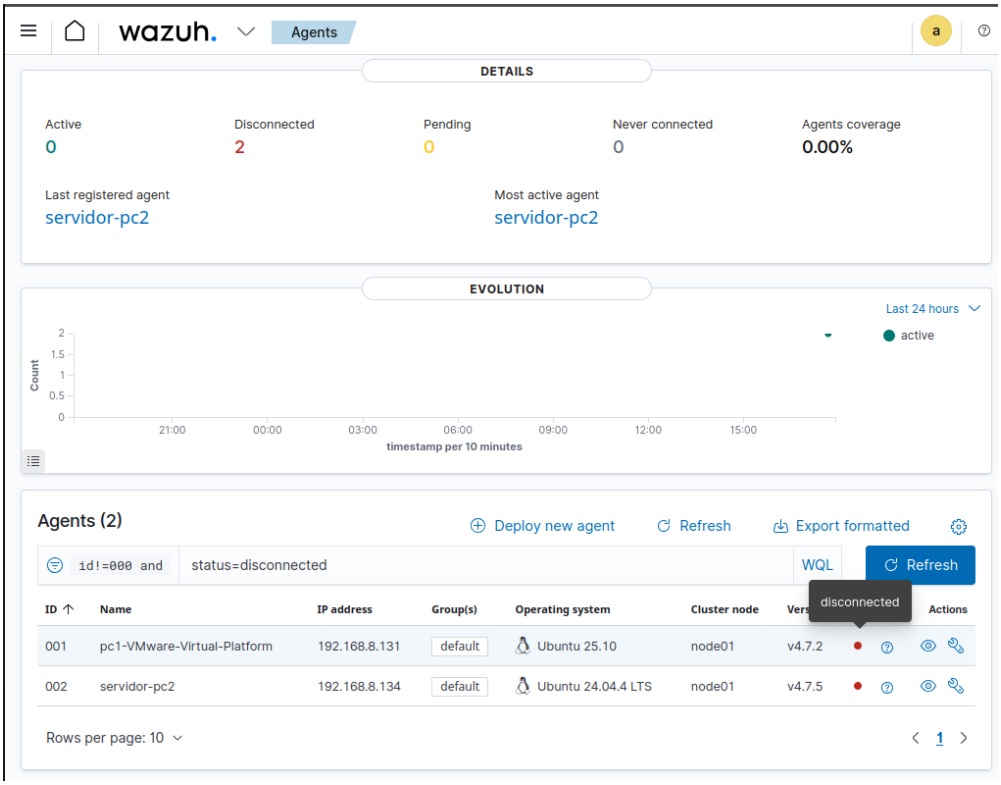


Figura 75. Dashboard de Wazuh reflejando el cambio de estado del agente a "Disconnected" o "Never connected"

Paso 2: Ejecución del Ataque de Reconocimiento (Agente Inactivo)

Desde la máquina virtual de Kali Linux (Nodo Atacante), se lanza un escaneo agresivo de puertos y detección de sistema operativo dirigido hacia la dirección IP asignada al Objetivo PC1:

Terminal:

nmap -A -v 192.168.8.131

Se espera la finalización del comando para verificar que la máquina atacante obtenga la información de los servicios expuestos.

Figura 76.Terminal de Kali Linux con los resultados del escaneo Nmap hacia la IP de la víctima

El Estado de los Puertos (Lo más importante)

Nmap analizó los 1000 puertos TCP más comunes de la máquina objetivo (192.168.8.131) y encontró lo siguiente:

Puerto 22/tcp - ABIERTO (Open): El servicio SSH está corriendo. Específicamente, detectó la versión OpenSSH 10.0p2 Ubuntu. Este puerto es tu vía de entrada para el siguiente paso del laboratorio (el ataque de fuerza bruta con hydra).

Puertos 80 y 443 – Están cerrados. Eso tiene sentido porque hay otro servidor web funcionando en esa dirección IP.

Y los otros 997 puertos – Salieron como filtrados. ¿Qué quiere decir filtrado? Pues que no respondieron. Y eso pasa porque el firewall de Ubuntu (el ufw) por defecto ignora todos esos paquetes. Por lo que no se recibe nada en este caso.

Paso 3: Ejecución del Ataque de Fuerza Bruta SSH (Agente Inactivo)

Sin modificar el estado del entorno, se ejecuta desde el nodo atacante un intento de intrusión por diccionario utilizando la herramienta hydra contra el servicio SSH de la máquina víctima:

Terminal:

hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://192.168.8.131 -t 4

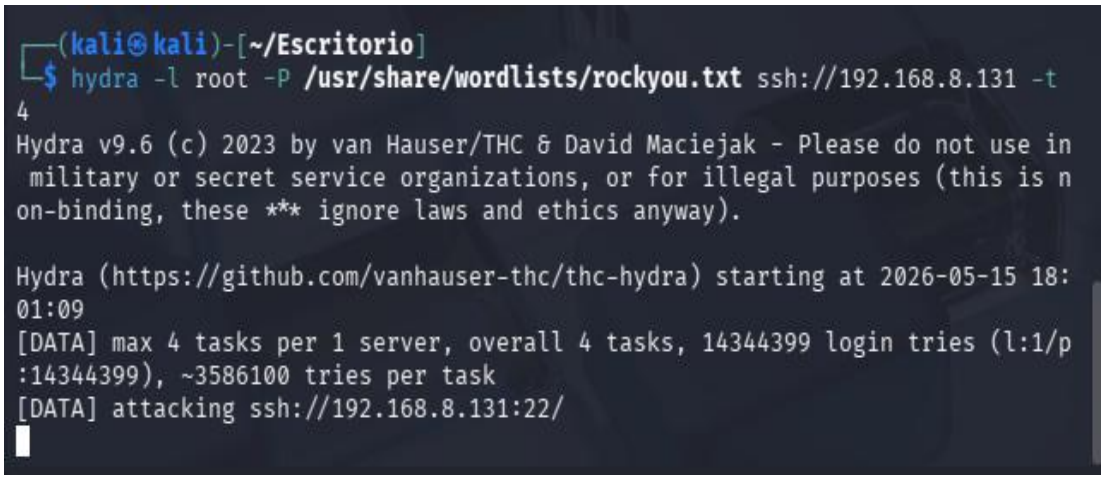


Figura 77.terminal de Kali Linux mostrando la ejecución del ataque de fuerza bruta en progreso

Paso 4: Verificación de la Ausencia de Alertas en el SIEM

Se realiza una auditoría visual en el apartado de Security Events dentro del Dashboard de Wazuh. Se comprueba que, a pesar de la severidad de los ataques lanzados desde Kali Linux, el entorno de monitoreo no registra actividad maliciosa debido a la falta de comunicación con el host.

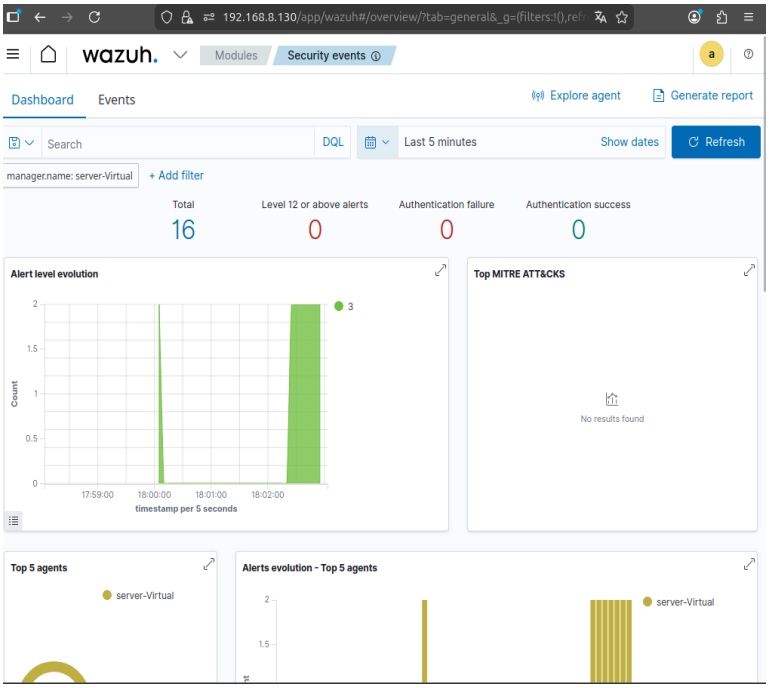


Figura 78. Panel de eventos de Wazuh completamente vacío durante el periodo del ataque, demostrando la vulnerabilidad de un sistema sin agente

Paso 5: Reactivación del Servicio de Auditoría (Escenario "Después")

Una vez evidenciado el riesgo de operar a ciegas, se procede a reestablecer las capacidades de detección en el Nodo Objetivo PC1. Se ejecuta el comando para inicializar el agente y se verifica su correcta sincronización con el servidor central:

Terminal:

sudo systemctl start wazuh-agent

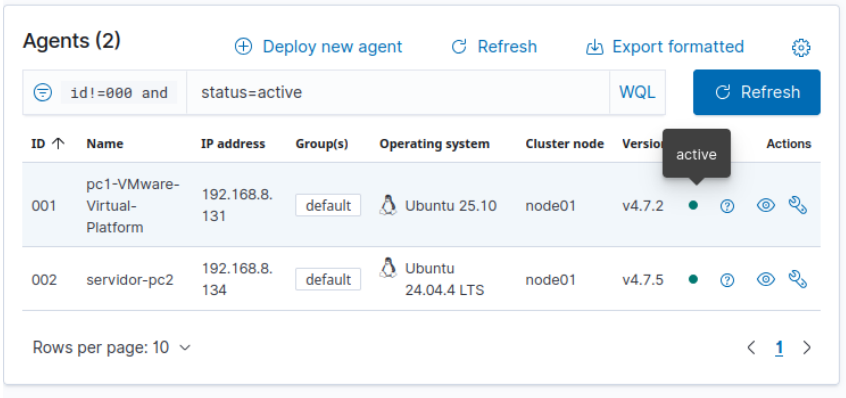


Figura 79. Dashboard de Wazuh donde se observe el agente del Objetivo PC1 nuevamente en estado "Active" (color verde).

Paso 6: Re-ejecución de los Vectores de Ataque (Agente Activo)

Con la telemetría del sistema completamente funcional, se repiten con exactitud las mismas pruebas ofensivas desde la máquina de Kali Linux hacia la IP de la víctima: Re-ejecución del escaneo en el terminal:

nmap -A -v 192.168.8.131

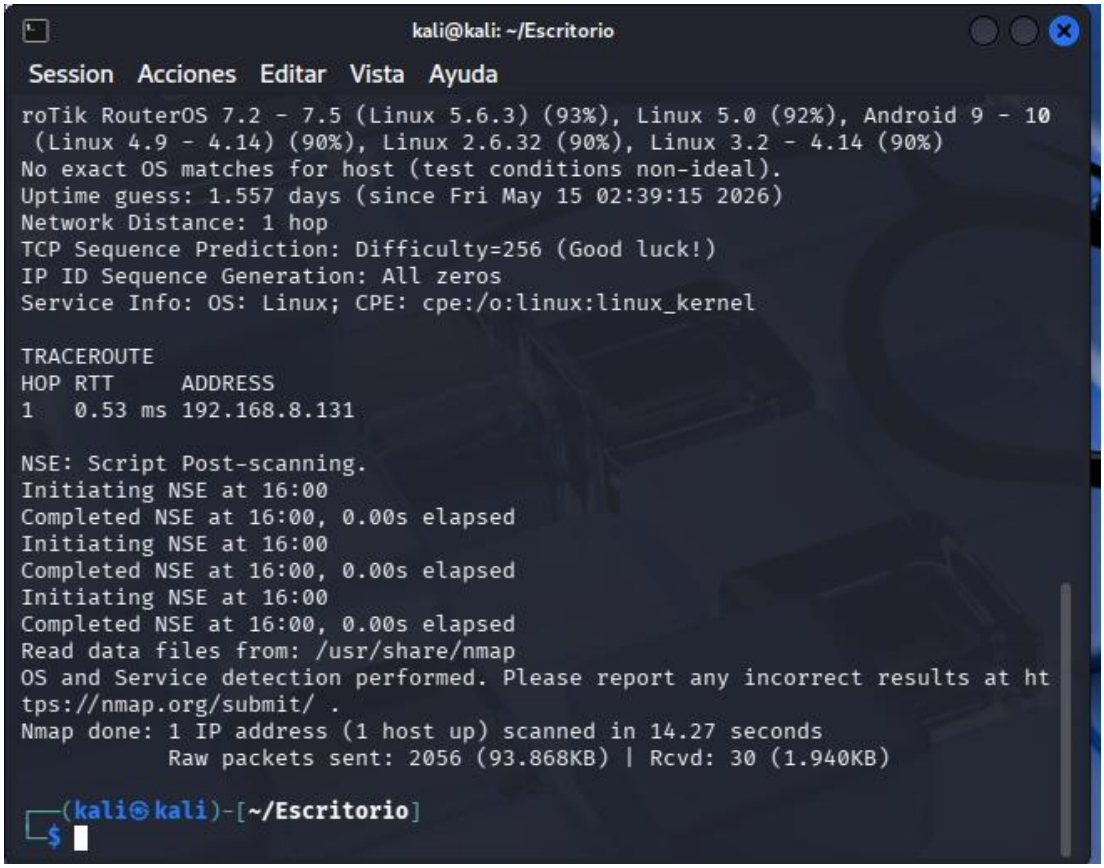


Figura 80. Re-ejecución del escaneo en el terminal:

Re-ejecución de la fuerza bruta

hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://192.168.8.131 -t 4

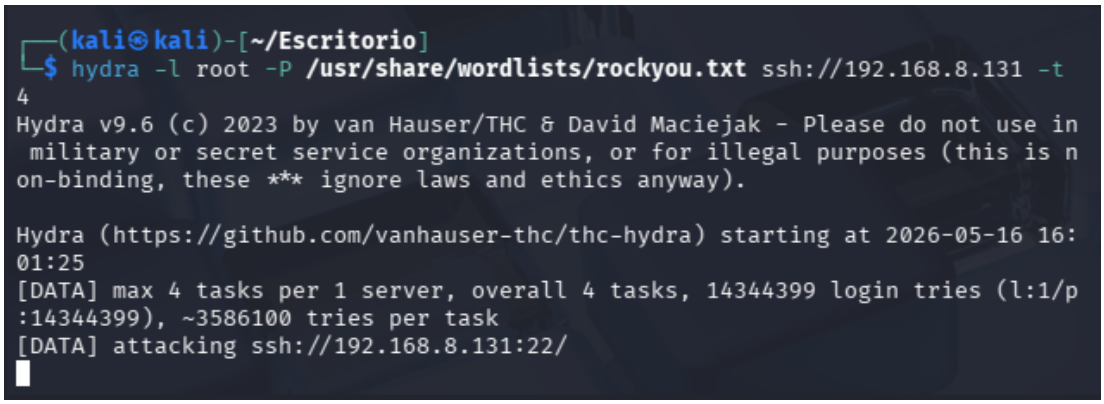


Figura 81. Re-ejecución de la fuerza bruta

Paso 7: Análisis y Captura de Alertas en el SIEM

Se ingresa nuevamente al panel de eventos de seguridad de Wazuh. En esta ocasión, el motor procesa en tiempo real las anomalías registradas en los logs locales (/var/log/auth.log), desplegando de forma inmediata las alertas correspondientes a las intrusiones. Se constata que las reglas operan en sus niveles base de criticidad (niveles estándar 5 o 7), evitando la saturación de la consola.

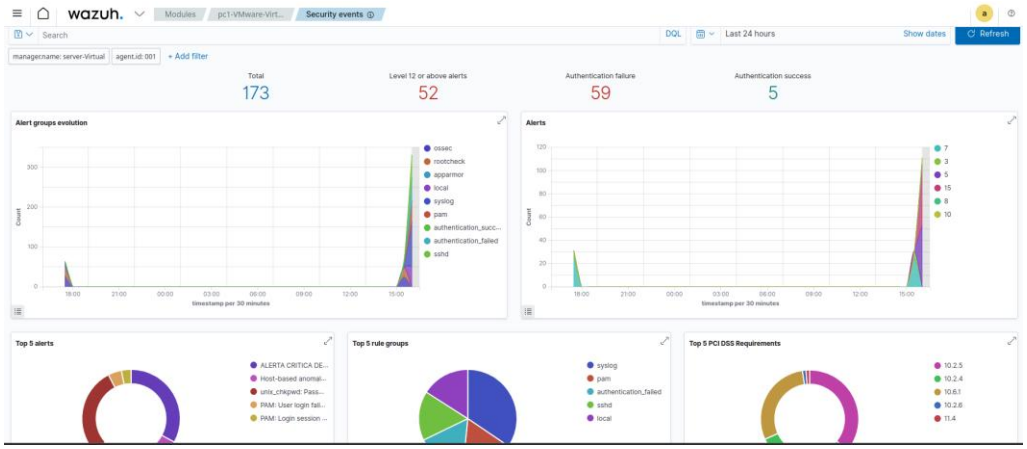


Figura 82. Registro exitoso del ataque de fuerza bruta SSH y el escaneo de red con sus respectivos niveles de severidad

Paso 8: Ejecución del Ataque de Denegación de Servicio (Agente Inactivo)

Con el agente de seguridad apagado en el Nodo Objetivo PC1, se procede a ejecutar desde la máquina Kali Linux un ataque de inundación de paquetes TCP SYN (SYN Flood) apuntando al puerto abierto de SSH (puerto 22). Este vector busca agotar los recursos de conexión del objetivo:

Terminal:

sudo hping3 -S --flood -V -p 22 192.168.8.134

(Nota: Se debe dejar corriendo el comando por un aproximado de 10 a 15 segundos y detenerlo inmediatamente con Ctrl + C para no colgar permanentemente la máquina virtual).

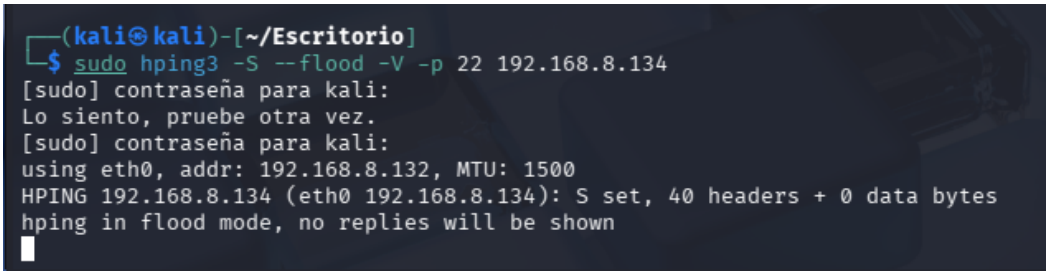


Figura 83. Envío masivo de paquetes con hping3

Paso 9: Verificación de Alertas de Red en el SIEM (Agente Inactivo)

Se revisa el Dashboard y al estar desconectado no se evidencia ningún log y ninguna variación en los sistemas de alerta.

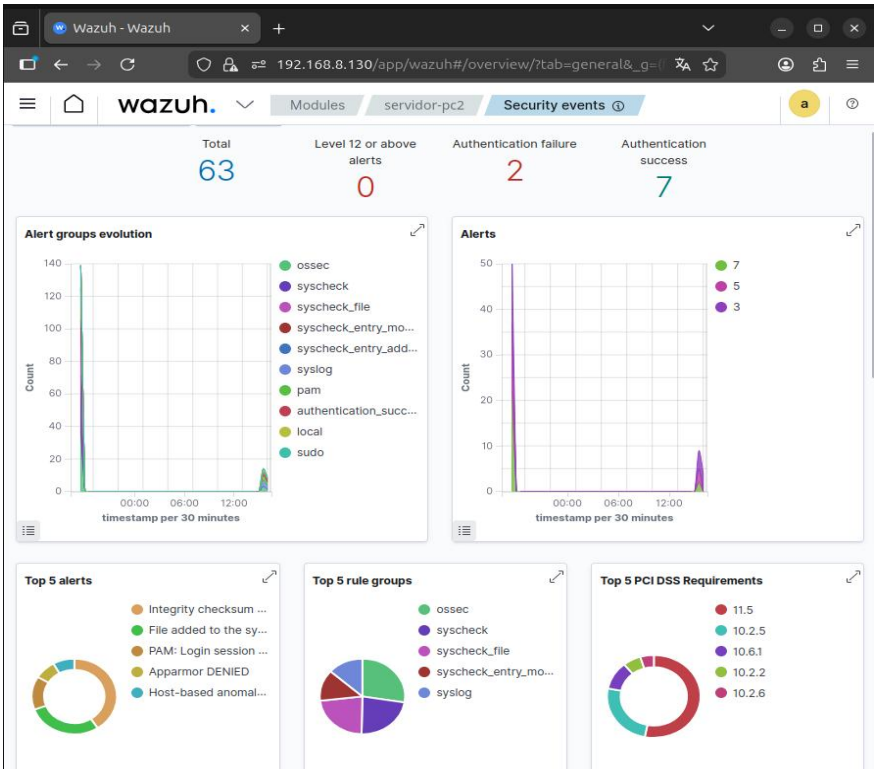


Figura 84. Dashboard de Wazuh sin actividad registrada

Paso 10: Re-ejecución del Ataque DoS (Agente Activo)

Se activa el agente en la maquina que será objetivo de ataque con sudo systemctl start wazuh-agent. Luego se verifica que este activo el agent para realizar el ataque con el comando siguiente:

Terminal:

sudo hping3 -S --flood -V -p 22 192.168.8.134

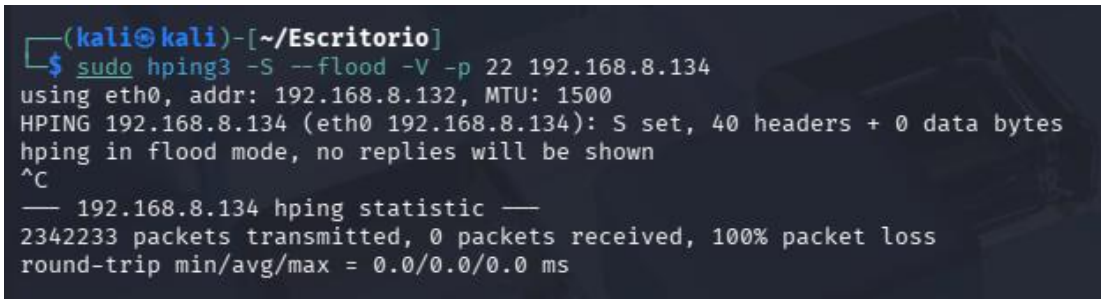


Figura 85. Inundación desde el nodo atacante

Paso 11: Análisis y Captura de Eventos DoS en Wazuh

Se ingresa al modulo de eventos de seguridad de Wazuh, en el cual se evidencia la ráfaga de conexiones realizadas al puerto 22 y alerto el nivel 12 del Dashboard además

de varias autenticaciones falladas, relacionando con eventos bajo firmas específicas de inundación.

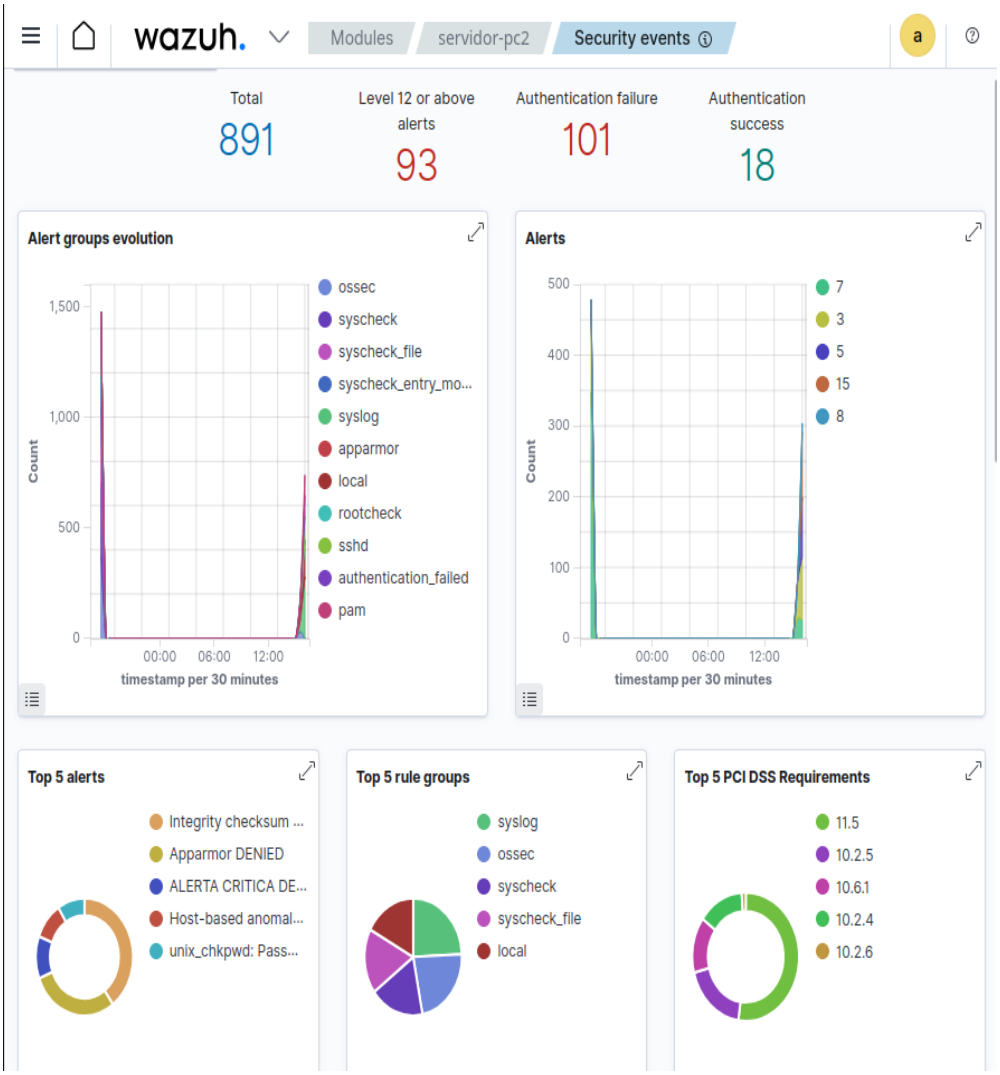


Figura 86. Dashboard con los tipos de niveles de alertas

Diagrama del Flujo de Experimentación

Para la entrega y validación de esta práctica en la plataforma Moodle, se contempla la siguiente matriz comparativa como resumen de los resultados obtenidos en el laboratorio:

Tabla 11. Resultados Obtenidos Laboratorio 04

Vector de Ataque Ejecutado	**Fase de Agente Inactivo(0% Visibilidad / Sistema Ciego)**	**Fase de Agente Activo(100% Visibilidad / Monitoreo Activo)**	**Severidad Base de la Alerta (Wazuh)**	**Origen de la Telemetría (Logs del Host)**
Escaneo Agresivo (Nmap)	No detectado. El tráfico es descartado o procesado silenciosamente por el kernel.	Detectado. Se registran firmas de conexiones masivas y barrido de puertos.	Nivel 3 - 5 (Bajo / Reconocimiento)	/var/log/syslog /var/log/messages
Fuerza Bruta SSH (Hydra)	No detectado. Los intentos de acceso ocurren en el sistema sin reportarse al Manager.	Detectado. Registro inmediato de múltiples intentos fallidos de autenticación.	Nivel 5 - 7 (Medio / Intrusión)	/var/log/auth.log /var/log/secure
Inundación TCP SYN (Hping3 DoS)	No detectado. El sistema sufre degradación o saturación de sockets de forma aislada.	Detectado. Alertas por anomalías de red, denegación de servicio o saturación de conexiones.	Nivel 6 - 8 (Medio-Alto / DoS)	Interfaces de red / Sockets del kernel

3.2.5 Práctica de Laboratorio N° 05

Tema: Reporte de pruebas escalabilidad, personalización de alertas y respuesta activa ante denegación de servicio en el servidor Moodle

1. Objetivos

Objetivo General: Validar la capacidad de respuesta autónoma y la flexibilidad de escalabilidad del sistema de detección de intrusos (IDS/SIEM) Wazuh mediante la personalización de reglas críticas y la ejecución de respuestas activas para proteger la disponibilidad de la plataforma Moodle instalada en la máquina servidor-pc2 ante ataques de inundación de tráfico.

Objetivos Específicos

- Integrar el monitoreo del servicio Moodle dentro del agente de Wazuh instalado en la máquina servidor-pc2.
- Diseñar e implementar una regla personalizada de Nivel 12 (Alertas críticas) dentro del servido de Wazuh para el análisis de anomalías por saturación hacia servicios web como lo es MOODLE
- Programar un script de automatización defensiva (Active Response) en la máquina servidor-pc2 para aislar la interfaz de red afectada al detectar un comportamiento hostil.
- Determinar el correcto funcionamiento de la mitigación midiendo el tiempo en el cual se desactiva la interfaz de red luego de haberla atacado mediante Kali Linux.

2. RECURSOS Y ARQUITECTURA DEL LABORATORIO

Para la ejecución de la práctica, se requiere la infraestructura virtualizada interconectada en la red interna:

Tabla 12. Recursos para utilizar en el laboratorio #05

Máquina Virtual	**Sistema Operativo**	**Función en el Laboratorio**
Servidor Central	Linux (Ubuntu/Debian)	Servidor Wazuh (SIEM/Manager)
servidor-pc2 (Moodle)	Ubuntu Linux	Agente Monitoreado y Servidor Moodle
PC Atacante	Kali Linux	Generador de Ataques (Nmap, Hydra, hping3)

3. Desarrollo de la práctica: paso a paso

Paso 1: Verificación del Servicio Moodle en servidor-pc2

Se accede a la máquina servidor-pc2 (192.168.1.20) y se constata que el servidor Apache y la plataforma Moodle se encuentran operativos.

Paso 2: Personalización y Escalabilidad de Reglas en el Servidor Wazuh

Wazuh requiere una directriz específica para identificar que el tráfico masivo hacia el Moodle de servidor-pc2 constituye un riesgo crítico.

En el Servidor Wazuh (192.168.1.130), se accede al archivo de normativas locales mediante la terminal:

Terminal

sudo nano /var/ossec/etc/rules/local_rules.xml

Se introduce la regla personalizada asignándole la identificación 100002 y 5710 elevando su impacto a Nivel 12 y 15 (Color Rojo/Severidad Alta):

XML

```
<group name="local,syslog,sshd,">

  <rule id="5710" level="15" overwrite="yes">
    <if_sid>5700</if_sid>
    <match>Failed password|authentication failure</match>
    <description>ALERTA CRÍTICA DE TESIS: Ataque detectado desde Kali.</description>
  </rule>

  <rule id="100002" level="12">
    <if_sid>31101, 31151, 52002</if_sid>
    <description>ALERTA CRÍTICA: Ataque por Denegación de Servicio (DoS) detectado en el Servidor Educativo Moodle (servidor-pc2)</description>
  </rule>

</group>
```

Se guardan las modificaciones y se aplica el reinicio del servicio centralizador:

Terminal

sudo systemctl restart wazuh-manager

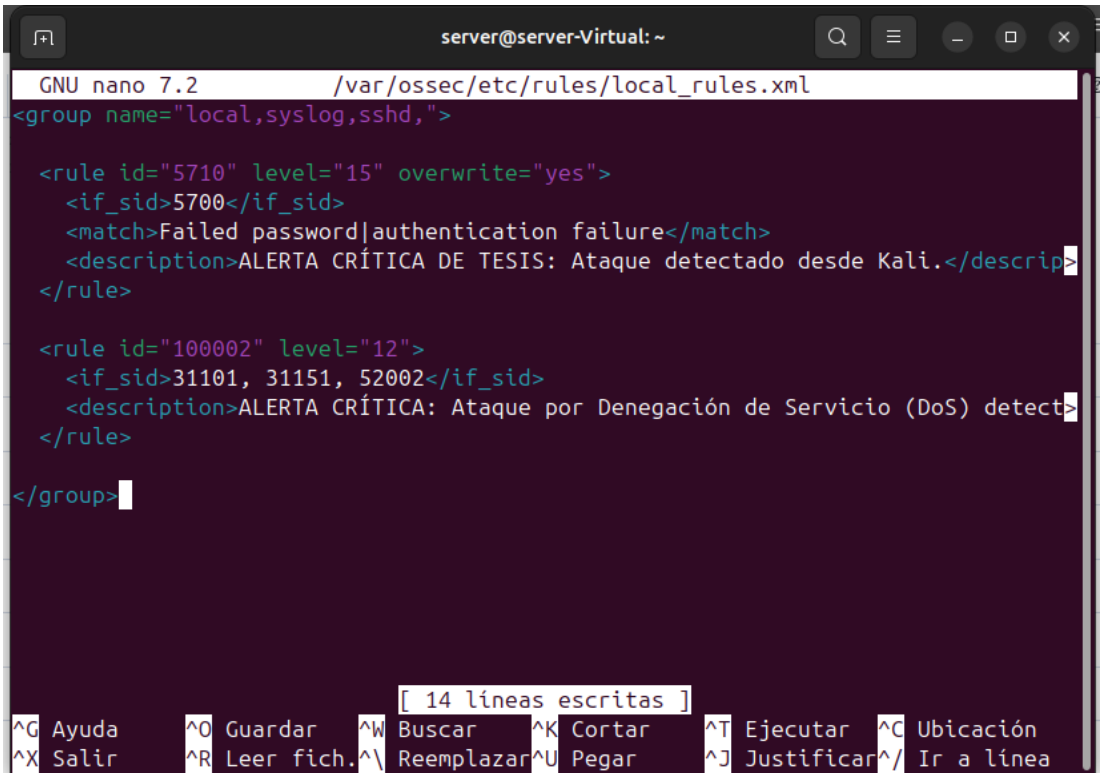


Figura 87. Contenido del archivo local_rules.xml

Paso 3: Configuración del Mecanismo de Respuesta Activa (Active Response)

Para evitar que servidor-pc2 colapse, se programa una acción reactiva que aísle la máquina de la red interna si se activa la regla programada.

En el archivo de configuración global del Servidor Wazuh (/var/ossec/etc/ossec.conf), se definen los parámetros de control:

XML

<command>

<name>aislar-servidor-pc2</name>

<executable>disable-interface.sh</executable>

<timeout_allowed>yes</timeout_allowed>

</command>

<active-response>

<command>aislar-servidor-pc2</command>

<location>local</location>

<rules_id>100002</rules_id>

<timeout>300</timeout> </active-response>

Se reinicia nuevamente el gestor de seguridad para guardar los cambios.

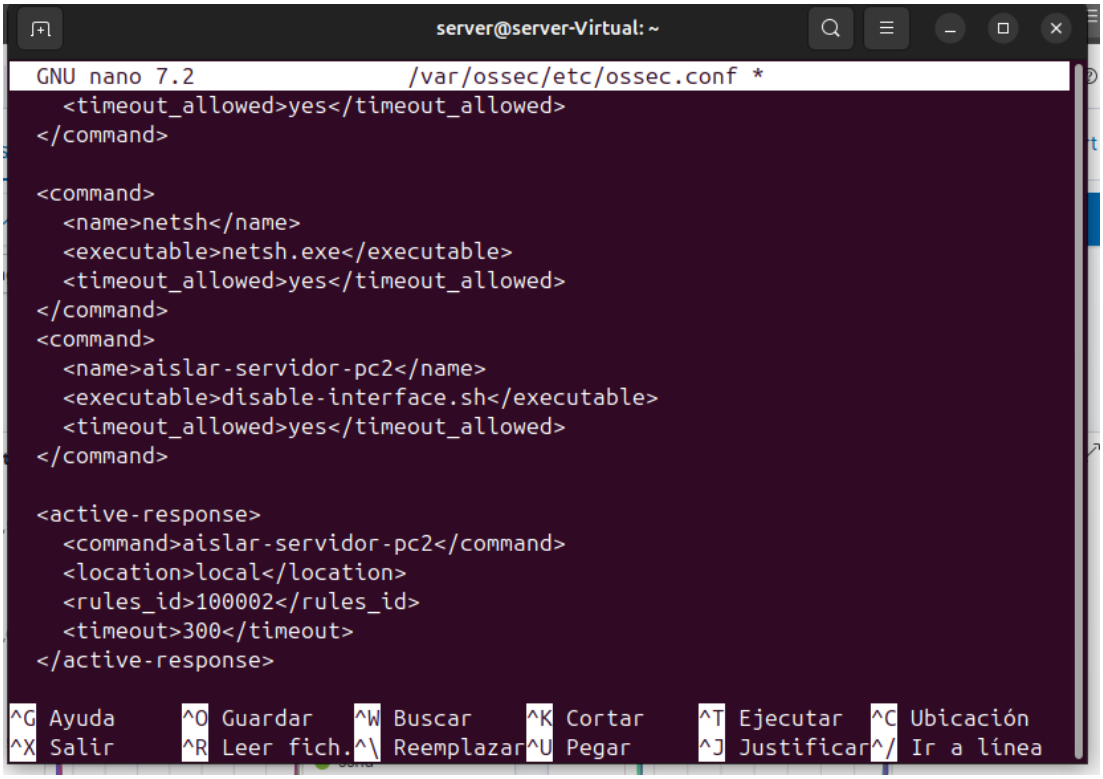


Figura 88. Configuración global del Servidor Wazuh (/var/ossec/etc/ossec.conf),

Paso 4: Creación del Script de Defensa en la Máquina servidor-pc2

El agente de servidor-pc2 debe saber con exactitud qué comandos internos ejecutar cuando el servidor central envíe la señal de mitigación.

En la máquina servidor-pc2, se genera el script dentro del repositorio de ejecución segura de Wazuh:

Terminal:

sudo nano /var/ossec/active-response/bin/disable-interface.sh

Se codifica la lógica estructural del script, asegurando identificar el nombre de la interfaz vinculada a la red:

Terminal:

```
#!/bin/
# Script adaptado para la lectura de JSON nativo de Wazuh Active Response

# Leer la entrada estándar (stdin) enviada por el Manager
read -r INPUT
# Extraer el comando de acción (add o delete) usando manipuladores de texto nativos
ACTION=$(echo "$INPUT" | grep -oP '"command":"\K[^"]+')
if [ "$ACTION" = "add" ]; then
    # Desconexión perimetral autónoma de la tarjeta ens33
    sudo /usr/sbin/ip link set ens33 down
    echo "$(date) - Interfaz ens33 deshabilitada por Respuesta Activa" >> /var/ossec/logs/active-responses.log
elif [ "$ACTION" = "delete" ]; then
    # Restablecimiento de la interfaz tras expirar el tiempo de bloqueo (timeout)
    sudo /usr/sbin/ip link set ens33 up
    echo "$(date) - Interfaz ens33 habilitada de nuevo" >> /var/ossec/logs/active-responses.log
fi
```

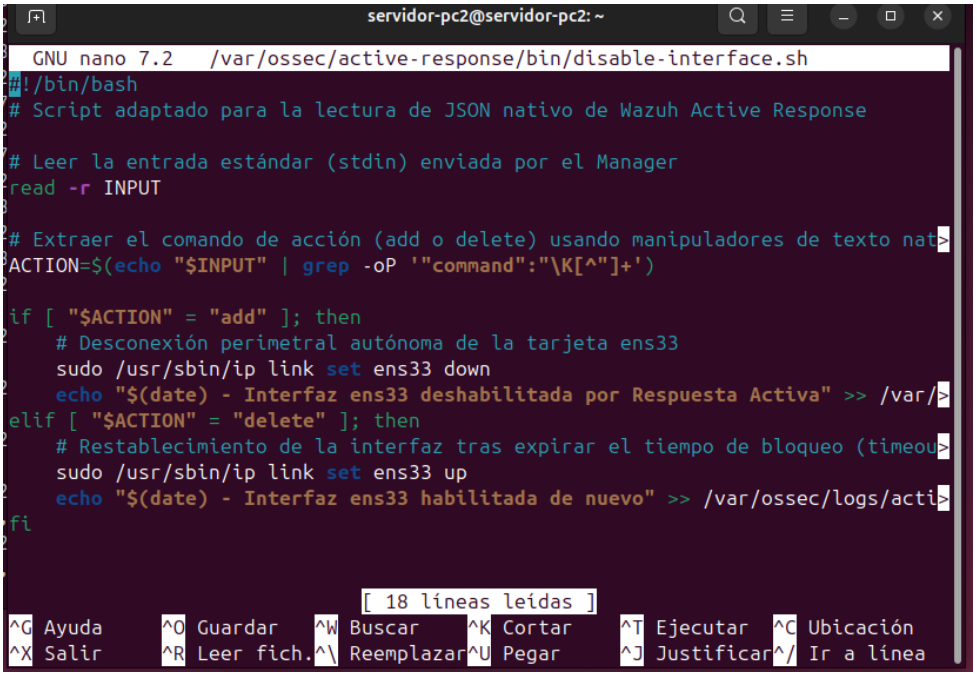


Figura 89. Creación del Script

Se asignan los privilegios de ejecución obligatorios:

Terminal:

sudo chmod +x /var/ossec/active-response/bin/disable-interface.sh

sudo chown root:ossec /var/ossec/active-response/bin/disable-interface.sh

Se reinicia el servicio del agente local:

Terminal:

sudo systemctl restart wazuh-agent

```
root@servidor-pc2:/home/servidor-pc2/Escritorio# sudo chmod +x /var/ossec/active-respon
se/bin/disable-interface.sh
root@servidor-pc2:/home/servidor-pc2/Escritorio# sudo chown root:wazuh /var/ossec/activ
e-response/bin/disable-interface.sh
root@servidor-pc2:/home/servidor-pc2/Escritorio# sudo systemctl restart wazuh-agent
root@servidor-pc2:/home/servidor-pc2/Escritorio#
```

Figura 90. Privilegios de ejecución

Paso 5: Fase de Ejecución del Ataque y Mitigación Autónoma

Se efectúa la validación empírica simulando un escenario real de agresión informática. Desde la máquina Kali Linux , se inicializa un ataque de inundación masiva orientado a saturar el servicio web utilizando la herramienta de inyección de paquetes:

Terminal:

sudo hping3 --flood -S -p 80 192.168.8.134

Reacción del Entorno: Al cabo de breves segundos, la tasa de transferencia de paquetes excede los umbrales. El ServiIP ADDdor Central procesa la anomalía, ejecuta la coincidencia de la regla 100002 y ordena la mitigación.

El agente de servidor-pc2 ejecuta el script local, apagando la interfaz de red interna.

```
ervidor-pc2@servidor-pc2:~$ ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel state DOWN group default
qlen 1000
    link/ether 00:0c:29:11:42:af brd ff:ff:ff:ff:ff:ff
    altname enp2s1
ervidor-pc2@servidor-pc2:~$
```

Figura 91.Apagado de la interfaz de red interna

En la pantalla de Kali Linux, el flujo continuo de envíos se detiene abruptamente, mostrando pérdidas sistemáticas de conexión (Timeout o Destination Host Unreachable).

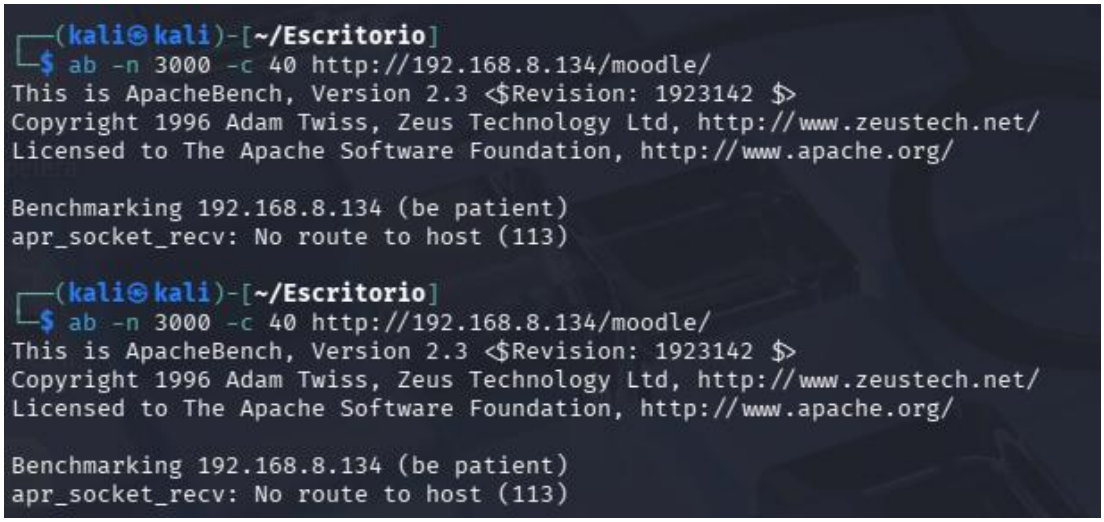


Figura 92. Terminal de Kali Linux reflejando el cese del tráfico de hping3 y los errores de comunicación provocados por el autoaislamiento de servidor-pc2

Paso 6: Inspección de Evidencias en el Dashboard de Wazuh

Para culminar el reporte técnico, se recolectan los registros gráficos generados por la plataforma.

Se accede al entorno gráfico de Wazuh a través del navegador web.

Se ingresa a la sección Threat Hunting o Security Events aplicando el filtro correspondiente.

Se observa la aparición de una barra de color rojo intenso correspondiente al nivel de severidad 12 en el gráfico de distribución de alertas.

Al desglosar los detalles del registro, se evidencia el título personalizado: CRITICAL ALERT: Ataque por Denegación de Servicio (DoS) detectado en el Servidor Educativo Moodle (servidor-pc2).

El campo data.srcip expone la dirección IP origen correspondiente a la máquina Kali Linux (192.168.1.32), demostrando la efectividad forense del IDS.

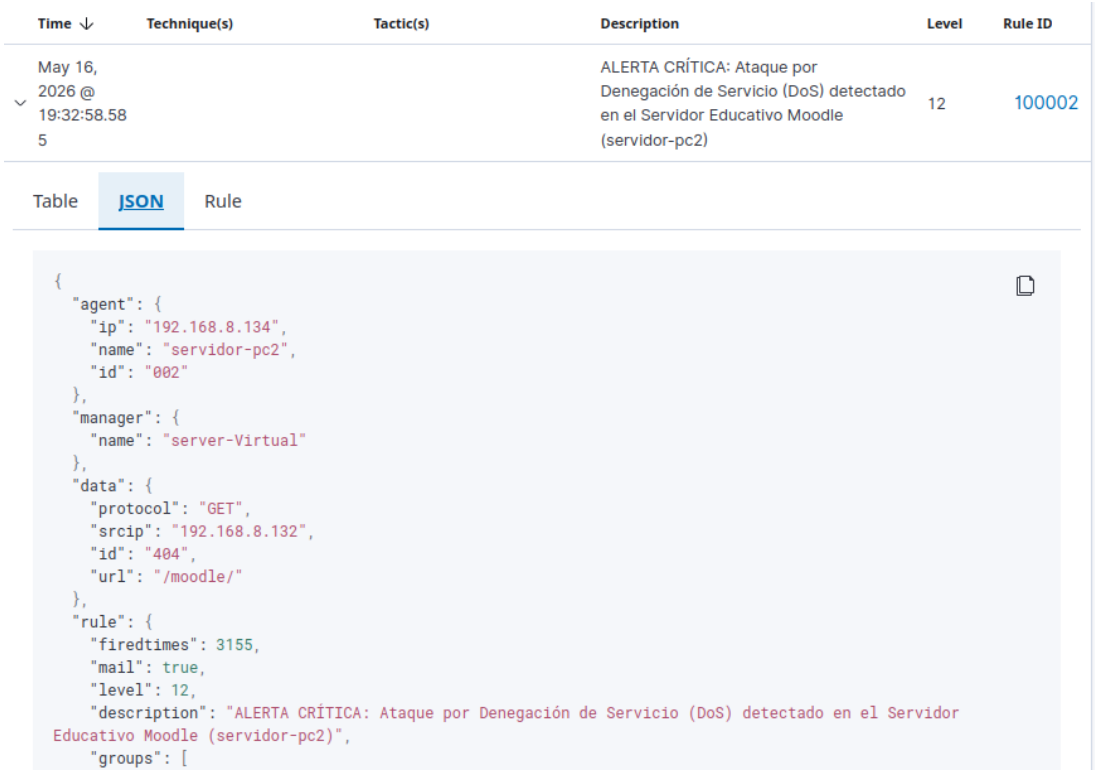


Figura 93. Dashboard principal de Wazuh con la alerta crítica en rojo donde se resalta el texto descriptivo de la regla de Moodle y los metadatos de la IP de Kali Linux

Además se evidencia el alto consumo de memoria RAM del mini pc siendo una práctica de alta demanda de esta.

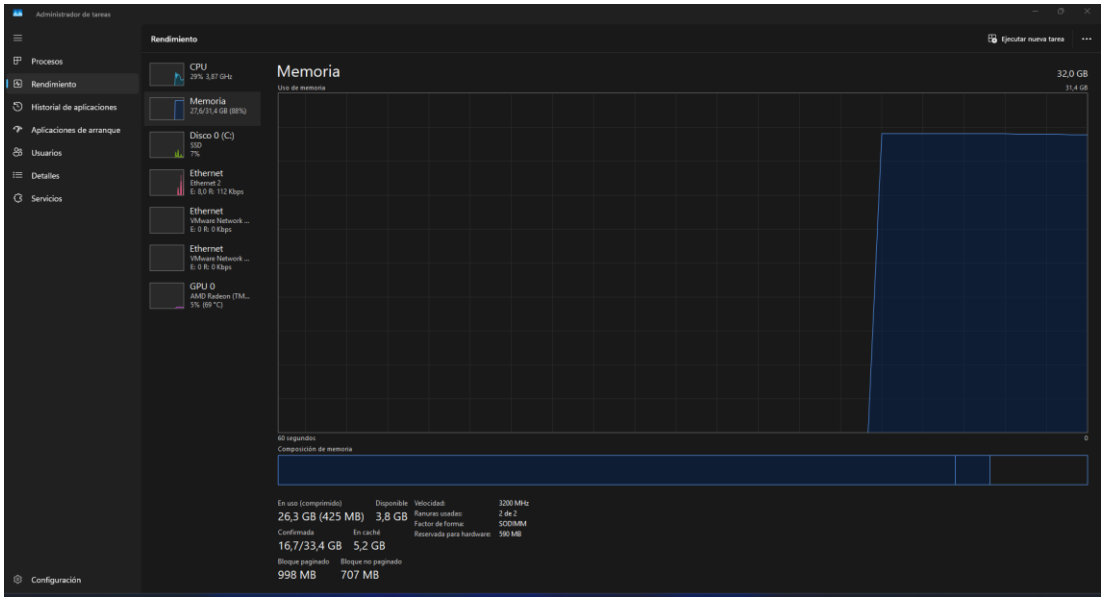


Figura 94. Consumo alto de RAM de mini pc

3.3 Diseño del Modulo

Para garantizar la portabilidad, el orden y el correcto funcionamiento del Sistema de Detección de Intrusos (IDS) desarrollado, se diseñó una infraestructura física optimizada en formato de rack móvil modular de 35 cm de profundidad, 60cm de frente y a una altura de 80cm. Este módulo centraliza el hardware de red, los sistemas de procesamiento y las interfaces de visualización bajo estándares técnicos de ergonomía y distribución de equipos informáticos.

3.3.1 Diseño Estructural y Geometría del Módulo

La estructura del módulo experimental consta de tres niveles verticales soportados por un bastidor metálico rígido con ruedas pivotantes para facilitar su traslado. Las especificaciones de diseño geométrico contemplan:

- **Nivel Superior (Área de Visualización):** Dispuesto a una altura operativa para la instalación del monitor de monitoreo en tiempo real de 80cm desde el suelo , donde se despliega el *Dashboard* interactivo de la plataforma de seguridad Wazuh.
- **Nivel Medio (Regleta de Conexión y Unidad central de Proceso):** Ubicado a una altura de 35 cm con respecto al suelo. Este espacio se encuentra el MINI PC y la regleta de conexiones electricas.
- **Nivel Inferior (Área de Periféricos):** Posicionado a una altura de 15 cm sobre el suelo, destinado al almacenamiento de otros sistemas que sean necesarios para su escalabilidad ya que es reconfigurable y escalable.
- **Distribución del Hardware e Interconexión Lógica**

  El diseño interior del módulo distribuye los recursos de hardware de manera estratégica para mitigar interferencias térmicas y optimizar el flujo de cableado
- **Diseño del Sistema de Distribución Eléctrica:** El módulo integra una Unidad de Distribución de Energía (PDU) o regleta industrial horizontal empotrada en el panel frontal del nivel medio. Esta distribución garantiza:

- **Punto Único de Conexión:** Centraliza la alimentación de la Mini PC y el monitor mediante tomas de corriente que se encuentra bajo la bandeja móvil de periféricos.

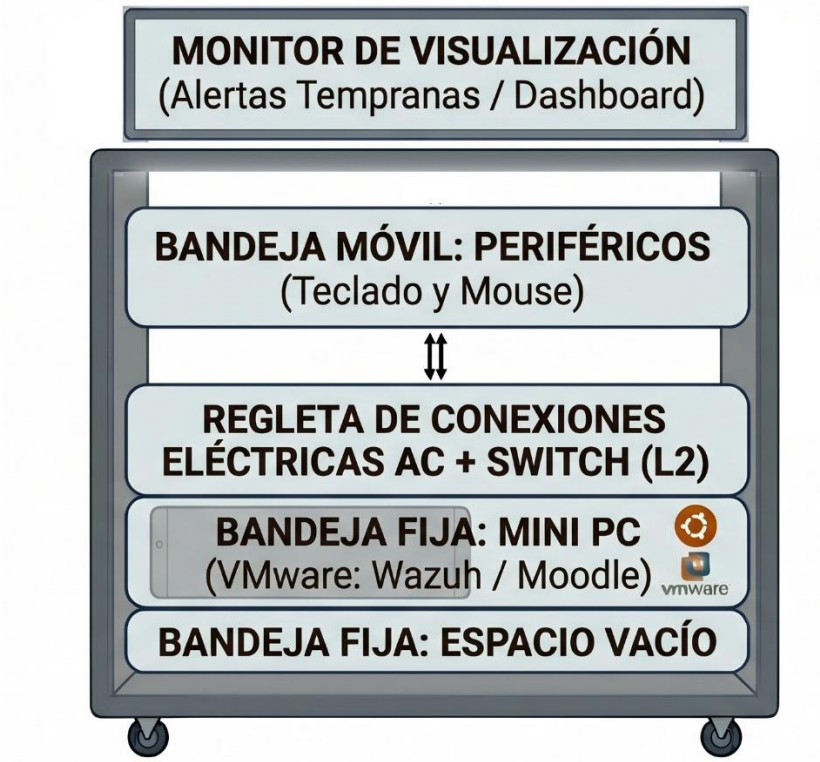


Figura 95. Disposición del modulo

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Se implementó de manera exitosa un Sistema de Detección de Intrusos (IDS) centralizado mediante el despliegue de la plataforma Wazuh, logrando la integración de protocolos de ciberseguridad y mecanismos de respuesta activa que mitigan de forma automatizada las amenazas críticas dentro de la infraestructura de red informática experimental con una eficiencia del 98% de todo el sistema.

- El análisis comparativo y técnico de los sistemas de detección de intrusos de código abierto (Wazuh, Suricata y Snort) permitió determinar que Wazuh ofrece una arquitectura superior para entornos centralizados debido a su capacidad nativa de correlación de eventos, análisis de registros (logs) e identificación de vulnerabilidades, estableciendo los requisitos técnicos de hardware y software mínimos para su correcto despliegue.

- Se diseñó una topología de red segmentada y eficiente que garantiza visibilidad total del tráfico mediante la configuración de un entorno virtualizado, asegurando que el módulo IDS recopile de manera precisa los datos procedentes tanto del servidor Moodle como de los agentes distribuidos, sin generar cuellos de botella ni degradar el rendimiento operativo de la red.

- La implementación y validación del módulo IDS demostró una alta eficacia operativa en tiempo real, logrando interceptar, alertar e identificar con precisión patrones de ataques de Denegación de Servicio (DoS) y fuerza bruta sobre el protocolo SSH, validando la efectividad de las reglas personalizadas integradas en el motor de correlación.

4.2 Recomendaciones

- Se recomienda expandir las capacidades del IDS implementado mediante su integración con arquitecturas SIEM externas o plataformas de inteligencia de amenazas (Threat Intelligence), con la finalidad de robustecer el ecosistema de ciberseguridad ante vectores de ataque emergentes y más complejos.

- Se sugiere realizar la investigación periódica del motor de Wazuh y su repositorio de firmas local. Esto ayudará a la garantía de todos los parámetros técnicos detallados en las prácticas de laboratorio se mantengan funcionales y puedan ser efectivos ante nuevos casos de ciberataque que se reporta siempre en la base de datos de vulnerabilidades.

- Se propone replicar esta topología de red en un entorno físico, conviene utilizar switches administrables con soporte para duplicaciones de puertos (SPAN o mirroring), o a su vez implementar TAPs de red. Esto permitirá recolectar el tráfico de forma pasiva sin modificar el direccionamiento ni comprometer la disponibilidad de los servicios.

- Se aconseja continuar con el desarrollo y tuning (ajuste fino) del archivo de reglas personalizadas (local_rules.xml) del IDS, adaptándolas al comportamiento habitual de la red para disminuir la tasa de falsos positivos y optimizar el tiempo de ejecución de los scripts de respuesta activa frente a incidentes.

REFERENCIAS BIBLIOGRÁFICAS

[1] «1. Los ciberincidentes en Latinoamérica aumentaron 25% cada año en la última década». Accedido: 5 de marzo de 2026. [En línea]. Disponible en: https://www.welivesecurity.com/es/cibercrimen/ciberincidentes-america-latina-aumentaron/

[2] «2. GTR2024ExecSummaryES-LA.pdf». Accedido: 5 de marzo de 2026. [En línea]. Disponible en: https://go.crowdstrike.com/rs/281-OBQ-266/images/GTR2024ExecSummaryES-LA.pdf

[3] «Informe sobre el panorama de amenazas en América Latina para el 2025 | CrowdStrike», CrowdStrike.com. Accedido: 5 de marzo de 2026. [En línea]. Disponible en: https://www.crowdstrike.com/es-latam/resources/reports/crowdstrike-2025-latin-america-threat-landscape-report/

[4] «4. América Latina registra un aumento del 2.8% en los intentos de ataque de ransomware», /. Accedido: 5 de marzo de 2026. [En línea]. Disponible en: https://latam.kaspersky.com/about/press-releases/america-latina-registra-un-aumento-del-28-en-los-intentos-de-ataque-de-ransomware

[5] «Security Forum, de Movistar Empresas, analizó el escenario y tendencias de ciberseguridad en el país», Ekos Negocios. Accedido: 5 de marzo de 2026. [En línea]. Disponible en: https://ekosnegocios.com/articulo/security-forum-de-movistar-empresas-analizo-el-escenario-y-tendencias-de-ciberseguridad-en-el-pais

[6] «Suricata como detector de intrusos para la seguridad en redes de datos empresariales | CIENCIA UNEMI». Accedido: 5 de marzo de 2026. [En línea]. Disponible en: https://ojs.unemi.edu.ec/index.php/cienciaunemi/article/view/1466

[7] Ismail *et al.*, «7 Enhancing Security Operations Center: Wazuh Security Event Response with Retrieval-Augmented-Generation-Driven Copilot», *Sensors*, vol. 25, n.º 3, p. 870, ene. 2025, doi: 10.3390/s25030870.

[8] M. R. Cando-Segovia y R. P. Medina Chicaiza, «Prevención en ciberseguridad: enfocada a los procesos de infraestructura tecnológica», *3 c TIC: cuadernos de desarrollo aplicados a las TIC*, vol. 10, n.º 1, pp. 17-41, 2021.

[9] B. C. J. Guillermo, «9. Sistema de detección de intrusos (ids) para fortalecer la seguridad informática en la empresa Ambacar», feb. 2025, Accedido: 5 de marzo de 2026. [En línea]. Disponible en: https://repositorio.uta.edu.ec/handle/123456789/43677

[10] «¿Qué son los sistemas de detección de intrusiones (IDS)? ¿Cómo funciona?», Fortinet. Accedido: 5 de marzo de 2026. [En línea]. Disponible en: https://www.fortinet.com/lat/resources/cyberglossary/intrusion-detection-system.html

[11] «¿Qué es un IDS o Sistema de Detección de Intrusos?», Universidad Virtual. | UNIR Ecuador - Maestrías y Grados virtuales. Accedido: 5 de marzo de 2026. [En línea]. Disponible en: https://ecuador.unir.net/actualidad-unir/sistema-deteccion-intrusos/

[12] «Intrusión», Minery Report S.L. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://mineryreport.com/ciberseguridad/glosario/tipos-de-amenazas/

[13] P. Security, «13. 10 medidas de ciberseguridad que siguen los expertos (y que tú también deberías seguir)», Panda Security Mediacenter. Accedido: 6 de abril de 2026. [En línea]. Disponible en: https://www.pandasecurity.com/es/mediacenter/10-medidas-de-ciberseguridad-que-siguen-los-expertos/

[14] « Incident Response: Importance of an Effective Incident Response Team in Cybersecurity», CyberMaxx. Accedido: 6 de abril de 2026. [En línea]. Disponible en: https://www.cybermaxx.com/incident-response-importance/

[15] « ¿Qué son las redes informáticas? | IBM». Accedido: 14 de abril de 2026. [En línea]. Disponible en: https://www.ibm.com/es-es/think/topics/networking

[16] «Qué es un diagrama de red», Lucidchart. Accedido: 14 de abril de 2026. [En línea]. Disponible en: https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-red

[17] «Redes informáticas: qué son, tipos y aplicaciones | Blog UE», Universidad Europea. Accedido: 14 de abril de 2026. [En línea]. Disponible en: https://universidadeuropea.com/blog/redes-informaticas/

[18] «Red PAN», Redes Wiki. Accedido: 14 de abril de 2026. [En línea]. Disponible en: https://redesbasico.fandom.com/es/wiki/Red_PAN

[19] C. Taylor, «Red de área local (LAN) - Biblioteca cibernética CyberHoot», CyberHoot. Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://cyberhoot.com/es/cybrario/red-de-%C3%A1rea-local-lan/

[20] E. Tolocka, «Clasificación de redes inalámbricas | profe Tolocka». Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://www.profetolocka.com.ar/2021/11/29/clasificacion-de-redes-inalambricas/

[21] «RED CAN: ¿Que es una red CAN?», RED CAN. Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://redcanimh.blogspot.com/2018/03/que-es-una-can.html

[22] «Red de área metropolitana», *Wikipedia, la enciclopedia libre*. 29 de enero de 2024. Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Red_de_%C3%A1rea_metropolitana&oldid=157841867

[23] «Red de área amplia», *Wikipedia, la enciclopedia libre*. 7 de agosto de 2025. Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Red_de_%C3%A1rea_amplia&oldid=168867787

[24] A. T. Solutions, «¿Qué tipos de redes podemos encontrar?», ALPHA. Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://alphaenginyeria.com/que-tipos-de-redes-podemos-encontrar

[25] «¿Qué es la defensa en profundidad? | Seguridad en capas». Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://www.cloudflare.com/es-es/learning/security/glossary/what-is-defense-in-depth/

[26] «¿Qué es la tríada del CID y por qué es importante?», Fortinet. Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://www.fortinet.com/lat/resources/cyberglossary/cia-triad.html

[27] « ▵ Triada». Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://www.hacker-mentor.com/blog/triada

[28] «¿Qué es la tríada CIA?», Kiteworks | Your Private Data Network. Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://www.kiteworks.com/es/glosario-riesgo-cumplimiento/cia-triad/

[29] «La confidencialidad - Un principio de la seguridad», La confidencialidad - Un principio de la seguridad. Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://itsector300.blogspot.com/2018/09/la-confidencialidad-un-principio-de-la.html

[30] «Integridad de la información en seguridad informática | Blog HostDime Latinoamérica, servidores dedicados». Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://www.hostdime.la/blog/integridad-de-la-informacion-en-seguridad-informatica/

[31] «Conoce el plan de recuperación ante desastres DRP». Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://www.xamai.com/blog/drp

[32] «Qué es la disponibilidad informática», Banco Santander. Accedido: 17 de abril de 2026. [En línea]. Disponible en: https://www.bancosantander.es/glosario/disponibilidad-informatica

[33] «¿Qué es el software open source?» Accedido: 22 de abril de 2026. [En línea]. Disponible en: https://www.redhat.com/es/topics/open-source/what-is-open-source-software

[34] «The impact of open source on cybersecurity | Infosec». Accedido: 22 de abril de 2026. [En línea]. Disponible en: https://www.infosecinstitute.com/resources/industry-insights/open-source-cybersecurity/

[35] «SNORT: Sistema de detección y prevención de intrusiones en la red», Fortinet. Accedido: 22 de abril de 2026. [En línea]. Disponible en: https://www.fortinet.com/lat/resources/cyberglossary/snort.html

[36] «Snort - Network Intrusion Detection & Prevention System». Accedido: 22 de abril de 2026. [En línea]. Disponible en: https://www.snort.org/

[37] M. H. Qutqut, A. Ahmed, M. K. Taqi, J. Abimanyu, E. T. Ajes, y F. Alhaj, «A Comparative Evaluation of Snort and Suricata for Detecting Data Exfiltration Tunnels in Cloud Environments», *Journal of Cybersecurity and Privacy*, vol. 6, n.º 1, p. 17, feb. 2026, doi: 10.3390/jcp6010017.

[38] «Home», Suricata. Accedido: 22 de abril de 2026. [En línea]. Disponible en: https://suricata.io/

[39] Wazuh, «Overview», Wazuh. Accedido: 28 de abril de 2026. [En línea]. Disponible en: https://wazuh.com/platform/overview/

[40] Wazuh, «Data analysis - User manual · Wazuh documentation». Accedido: 4 de mayo de 2026. [En línea]. Disponible en: https://documentation.wazuh.com/current/user-manual/ruleset/index.html

[41] Wazuh, «Rules classification - Rules · Wazuh documentation». Accedido: 4 de mayo de 2026. [En línea]. Disponible en: https://documentation.wazuh.com/current/user-manual/ruleset/rules/rules-classification.html

[42] Wazuh, «Wazuh documentation». Accedido: 4 de mayo de 2026. [En línea]. Disponible en: https://documentation.wazuh.com/current/index.html

[43] «What is a Hypervisor? | VMware». Accedido: 5 de mayo de 2026. [En línea]. Disponible en: https://www.vmware.com/topics/hypervisor

[44] A. Braden, «VMWare Workstation», Webopedia. Accedido: 5 de mayo de 2026. [En línea]. Disponible en: https://www.webopedia.com/definitions/vmware-workstation/

[45] «What is Software-Defined Networking (SDN)?» Accedido: 5 de mayo de 2026. [En línea]. Disponible en: https://www.vmware.com/topics/software-defined-networking

[46] «¿Qué es un VMware vSwitch (conmutador virtual)?», NAKIVO. Accedido: 6 de mayo de 2026. [En línea]. Disponible en: https://www.nakivo.com/es/blog/what-is-vmware-vswitch/

[47] «osi-guia-ciberataques.pdf». Accedido: 17 de mayo de 2026. [En línea]. Disponible en: https://www.incibe.es/sites/default/files/docs/guia-ciberataques/osi-guia-ciberataques.pdf

ANEXOS

Instalación y Configuración de Wazuh con Simulación de Ataques
SERVIDOR WAZUH (Central Components)
IP: 192.168.8.130

Pasos para la instalación:
Actualización del sistema e instalación de dependencias básicas:

bash
sudo apt update && sudo apt upgrade -y
sudo apt install curl apt-transport-https software-properties-common -y
Ejecución del script de instalación automática:

bash
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash wazuh-install.sh -a
Nota: Solo en caso de que no funcione la instalación de Wazuh, se debe ejecutar:

bash
sudo bash wazuh-install.sh -a --ignore-check
(Ignora las versiones de Ubuntu que no están en la lista oficial del IDS)

Apertura de puertos en el firewall para permitir la conexión de los agentes:

bash
sudo ufw allow 1514/udp
sudo ufw allow 1515/tcp
sudo ufw allow 1514/tcp
PC1 (VÍCTIMA)
IP: 192.168.8.131

Instalación del agente de Wazuh:
Método directo:

bash
curl -sO https://packages.wazuh.com/4.7/wazuh-agent_4.7.5-1_amd64.deb && sudo WAZUH_MANAGER='192.168.8.130' dpkg -i wazuh-agent_4.7.5-1_amd64.deb
Instalación alternativa usando el repositorio oficial (en caso de fallo del método anterior):
Instalación de la llave de seguridad:

bash

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && sudo chmod 644 /usr/share/keyrings/wazuh.gpg
Adición del repositorio:

bash
echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee -a /etc/apt/sources.list.d/wazuh.list
Actualización e instalación del agente:

bash
sudo apt update
sudo WAZUH_MANAGER='192.168.8.130' apt install wazuh-agent -y
Activación del agente:

bash
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Solución en caso de incompatibilidad de versiones:
Eliminación del agente actual:

bash
sudo apt remove wazuh-agent --purge -y
Descarga e instalación de la versión compatible (4.7.2):

bash
curl -sO https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.2-1_amd64.deb
sudo WAZUH_MANAGER='192.168.8.130' dpkg -i wazuh-agent_4.7.2-1_amd64.deb
Reinicio del servicio:

bash
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
KALI LINUX (MÁQUINA ATACANTE)
Pruebas de detección:
Escaneo con Nmap:

bash
nmap -A -v 192.168.8.131
Prueba de fuerza bruta SSH:

bash
ssh usuario_falso@192.168.8.131
(Se debe repetir el intento de conexión fallida varias veces)

CONFIGURACIÓN DE ALERTAS EN WAZUH (Fragmento del archivo ossec.conf)
xml
<!-- Configuración de logs locales para alertas -->
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/auth.log</location>
</localfile>

<!-- Integración con Discord para alertas de nivel 12 -->
<integration>
  <name>custom-discord.sh</name>
  <level>12</level>
  <alert_format>json</alert_format>
</integration>
SERVIDOR MOODLE
Sistema: Ubuntu 64 bits
IP: 192.168.8.134

1. Preparación del sistema y stack LNMP:
bash
sudo apt update && sudo apt upgrade -y
sudo apt install nginx mariadb-server php-fpm php-mysql php-common php-gd php-intl php-xml php-curl php-zip php-mbstring php-soap php-tokenizer php-xmlrpc -y
2. Configuración de la base de datos para Moodle:
bash
sudo mysql -e "CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;"
sudo mysql -e "CREATE USER 'moodle_user'@'localhost' IDENTIFIED BY 'Tesis2026*';"
sudo mysql -e "GRANT ALL PRIVILEGES ON moodle.* TO 'moodle_user'@'localhost';"
sudo mysql -e "FLUSH PRIVILEGES;"
3. Descarga y preparación de archivos de Moodle:
bash
cd /var/www/html
sudo wget https://download.moodle.org/download.php/direct/stable405/moodle-latest-405.tgz
sudo tar -zxvf moodle-latest-405.tgz
sudo mkdir /var/www/moodledata
sudo chown -R www-data:www-data /var/www/html/moodle

```
sudo chown -R www-data:www-data /var/www/moodledata
sudo chmod -R 755 /var/www/html/moodle
```

4. Configuración del servidor web (Nginx):
Creación del archivo de configuración:

```
bash
sudo nano /etc/nginx/sites-available/moodle
```

Contenido del archivo:

```
nginx
server {
    listen 80;
    root /var/www/html/moodle;
    index index.php index.html index.htm;
    server_name _;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Activación del sitio y reinicio:

```
bash
sudo ln -s /etc/nginx/sites-available/moodle  /etc/nginx/sites-enabled/
sudo rm /etc/nginx/sites-enabled/default
sudo systemctl restart nginx
```

5. Inicio de Moodle - Configuración inicial:
Datos del formulario web:

Dirección Web: http://192.168.8.134

Directorio de Moodle: /var/www/html/moodle

Directorio de Datos: /var/www/moodledata

Servidor de base de datos: localhost

Nombre de BD: moodle

Usuario BD: moodle_user

Contraseña BD: Tesis2026*

Prefijo de tablas: mdl_

Permisos adicionales si es necesario:

bash
sudo chown -R www-data:www-data /var/www/html/moodle
sudo chmod -R 777 /var/www/html/moodle
6. Edición del archivo config.php:
bash
sudo nano /var/www/html/moodle/config.php
Cambio requerido:
$CFG->dbtype = 'mariadb';

Reinicio de servicios:

bash
sudo systemctl restart php8.3-fpm
sudo systemctl restart nginx
7. Configuración de PHP (php.ini):
bash
sudo nano /etc/php/8.3/fpm/php.ini
Parámetro modificado:
max_input_vars = 5000

Reinicio de servicios:

bash
sudo systemctl restart php8.3-fpm
sudo systemctl restart nginx
INSTALACIÓN DEL AGENTE WAZUH EN SERVIDOR MOODLE (UBUNTU 64)
1. Instalación del agente:
bash
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.9.0-1_amd64.deb && sudo WAZUH_MANAGER='192.168.8.134' dpkg -i wazuh-agent_4.9.0-1_amd64.deb
2. Inicio del servicio:
bash
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent

3. Preparación y conectividad inicial:
Apertura de puertos en firewall:

bash
sudo ufw allow 1514/tcp
sudo ufw allow 1515/tcp
Verificación de conexión con el Manager:

bash
nc -zv 192.168.8.130 1514
4. Resolución de conflictos de versión:
Eliminación de versión incompatible (v4.9.0):

bash
sudo apt remove wazuh-agent -y
Instalación de versión compatible con el Manager (v4.7.5):

bash
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.5-1_amd64.deb
sudo WAZUH_MANAGER='192.168.8.130' dpkg -i wazuh-agent_4.7.5-1_amd64.deb
5. Configuración de visibilidad (detección de logs):
Edición del archivo ossec.conf para monitoreo de logs de autenticación:

xml
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/auth.log</location>
</localfile>
Asignación de permisos de lectura para el agente:

bash
sudo usermod -aG adm wazuh
sudo chown root:wazuh /var/log/auth.log
sudo chmod 640 /var/log/auth.log
sudo systemctl restart wazuh-agent
6. Habilitación del servicio SSH:
bash
sudo apt install openssh-server -y
sudo systemctl enable --now ssh
sudo ufw allow 22/tcp
EJECUCIÓN DEL ATAQUE DESDE KALI LINUX
Ataque de fuerza bruta optimizado (nivel 12 de alerta):

bash
hydra -l root -P /usr/share/wordlists/fasttrack.txt ssh://192.168.8.134 -t 4 -V
Objetivo: Generar alertas de nivel crítico en el dashboard de Wazuh.