UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Orquestador de Agentes de IA para gestión de incidentes de ciberseguridad

José Luis Contreras Parreño

Ing. Ciencias de la Computación

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en Ciencias de la Computación

Quito, de diciembre de 2025

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Orquestador de Agentes de IA para gestión de incidentes de ciberseguridad

José Luis Contreras Parreño

Nombre del profesor, Título académico **Roberto Andrade,** PhD. in Informatics

Quito, día de diciembre de 2025

© DERECHOS DE AUTOR

Nombres y apellidos: José Luis Contreras Parreño

Código: 00203046

Cédula de identidad: 1718683863

Lugar y fecha: Quito, día de diciembre de 2025

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

Este trabajo presenta el diseño e implementación de un orquestador de agentes de inteligencia artificial (IA) para la gestión de incidentes de ciberseguridad. La investigación parte de las limitaciones de los sistemas tradicionales SIEM y SOAR, los cuales requieren alta intervención humana, integraciones costosas y presentan tiempos elevados de respuesta. El proyecto propone un enfoque multi-agente que combina Modelos de Lenguaje de Gran Escala (LLMs) y técnicas de Recuperación Aumentada por Generación (RAG) cuya implementación está orientada a explorar el potencial de reducir tiempos de respuesta y aminorar la dependencia de intervención humana mediante la automatización de triage de logs, notificación de alertas y acciones de respuesta mediante ejecución de playbooks en un ambiente SOC. La arquitectura diseñada incluye agentes especializados en monitoreo, análisis y respuesta, coordinados mediante n8n y LanGraph. Se implementó un laboratorio para simular un entorno de infraestructura básica de red corportavia con servidores, estaciones de trabajo y herramientas de monitoreo sobre el cual se realizaron ataques controlados y conexión con SIEM de software Libre Wazuh para mediante comparación demostrar el funcionamiento del orquestador, generar una línea base y métricas de rendimiento. Los resultados esperados incluyen una reducción del MTTR de al menos un 20% y una disminución de falsos positivos en un 15% respecto a la línea base. Este trabajo busca demostrar la viabilidad de soluciones flexibles, escalables y de menor costo operativo para la gestión de incidentes en entornos de seguridad corporativa y de servicios.

Palabras clave: ciberseguridad, inteligencia artificial, orquestación multi-agente, SIEM, SOAR, LLM, RAG, LanGraph, n8n, máquina virtual.

ABSTRACT

This work presents the design and implementation of an artificial intelligence (AI) agent orchestrator for cybersecurity incident management. The research is motivated by the limitations of traditional SIEM and SOAR systems, which require extensive human intervention, costly integrations, and often exhibit high response times. The project proposes a multi-agent approach that combines Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG), aimed at exploring the potential to reduce response times and lessen dependence on human operators by automating log triage, alert notification, and response actions through the execution of playbooks in a SOC environment. The proposed architecture includes specialized agents for monitoring, analysis, and response, coordinated through n8n and LangGraph. A laboratory environment was implemented to simulate a basic corporate network infrastructure—with servers, workstations, and monitoring tools—on which controlled attacks were carried out and integrated with the open-source Wazuh SIEM. This setup enables comparison with a baseline configuration and the derivation of performance metrics. The expected results include at least a 20% reduction in Mean Time to Resolution (MTTR) and a 15% decrease in false positives relative to the baseline. This work seeks to demonstrate the feasibility of flexible, scalable, and lower-operational-cost solutions for incident management in corporate and service-oriented security environments.

Key words: cybersecurity, artificial intelligence, multi-agent orchestration, SIEM, SOAR, LLM, RAG, LangGraph, n8n, máquina virtual.

TABLA DE CONTENIDOS

INDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

La gestión de incidentes de ciberseguridad se ha convertido en un desafío crítico en un contexto donde la complejidad de las infraestructuras tecnológicas y la sofisticación de los ataques avanzan a un ritmo sin precedentes. Informes recientes de la Cybersecurity and Infrastructure Security Agency (CISA, 2024) y de la European Union Agency for Cybersecurity (ENISA, 2023) destacan que el volumen de incidentes de seguridad gestionados por centros de operaciones de seguridad (SOC) ha incrementado en alrededor del 30% anual. Este crecimiento genera una sobrecarga para analistas humanos, quienes deben filtrar, clasificar y priorizar cientos o miles de alertas diarias.

Los sistemas tradicionales SIEM (Security Information and Event Management) centralizan registros y correlacionan eventos de distintas fuentes, brindando una primera capa de visibilidad. Sin embargo, sus limitaciones son claras: requieren intervención humana constante, sus correlaciones se basan en reglas estáticas, y presentan dificultades para adaptarse a amenazas emergentes (Islam, Babar & Nepal, 2019). Por su parte, las plataformas SOAR (Security Orchestration, Automation and Response) incorporan flujos de automatización con "playbooks" predefinidos, mejorando los tiempos de reacción, pero su efectividad depende de integraciones costosas y rígidas (Kremer et al., 2023).

En este contexto surge la propuesta de diseñar un orquestador de agentes de inteligencia artificial (IA), donde múltiples agentes especializados en monitoreo, análisis y respuesta colaboran entre sí bajo mecanismos de coordinación estructurados. El presente trabajo busca demostrar que un enfoque multi-agente puede:

- Reducir el MTTR (Mean Time to Resolution) en al menos un 20%.
- Disminuir falsos positivos en un 20% frente a una línea base.

- Ofrecer explicaciones trazables para cada decisión de análisis y respuesta.

Por una parte, la relevancia del proyecto podría significar una asistencia en respuesta a la necesidad de soluciones escalables y adaptables frente a la evolución constante de ciberataques; por otra, plantea una alternativa viable para contextos corporativos y de servicios en países en desarrollo, donde el acceso a licencias propietarias resulta limitado debido al alto costo.

ESTADO DEL ARTE

La gestión de incidentes de ciberseguridad ha evolucionado de manera significativa en las últimas dos décadas. El crecimiento exponencial de los ataques dirigidos y la complejidad de las infraestructuras tecnológicas obligaron a pasar de enfoques manuales a plataformas centralizadas y posteriormente a sistemas automatizados.

La creciente sofisticación y volumen de los incidentes de ciberseguridad ha puesto en evidencia las limitaciones de los sistemas tradicionales de gestión de eventos e información de seguridad (SIEM) y de orquestación, automatización y respuesta de seguridad (SOAR). Estos sistemas, aunque fundamentales, enfrentan retos significativos en la correlación de eventos, reducción de falsos positivos y adaptación a amenazas emergentes. La necesidad de enfoques más adaptativos, escalables y automatizados es cada vez más indispensable.

Mientras que SIEM y SOAR han sido la columna vertebral de los centros de operaciones de seguridad (SOC), la irrupción de modelos de lenguaje de gran escala (LLMs) y arquitecturas de generación aumentada por recuperación (RAG) ha abierto nuevas posibilidades. La orquestación multi-agente basada en LLMs y RAG promete superar las limitaciones de los sistemas tradicionales, integrando razonamiento contextual, aprendizaje continuo y respuesta autónoma.

Un Centro de Operaciones de Seguridad (SOC) es una unidad centralizada responsable de monitorear, analizar y defender la postura de seguridad de una organización. Los SOC desempeñan un papel fundamental en la protección de los activos digitales, como los datos, las aplicaciones y la infraestructura, frente a las ciberamenazas. La cual es realizada a través de una combinación de personas, procesos y tecnologías, incluyendo herramientas sofisticadas como sistemas de gestión de eventos e información de seguridad (SIEM), firewalls, sistemas de detección de intrusos y algoritmos de aprendizaje automático.

Esquema Canónico de Eventos (CEC) y Comparación con Otros Esquemas

El Esquema Canónico de Eventos (CEC) es una representación estructurada de eventos que permite modelar el conocimiento del mundo sobre la progresión típica de eventos. Este esquema se puede inducir utilizando modelos de lenguaje preentrenados (LLMs), lo que simplifica el proceso de inducción y permite manejar relaciones jerárquicas y temporales de manera eficiente. A continuación, se presenta una comparación del CEC con otros esquemas de eventos como CEF de ArcSight, Syslog y STIX.

Comparación de Esquemas de Eventos

Tabla 1. Comparación CEC

ESQUEMA	DESCRIPCIÓN	VENTAJAS	DESVENTAJAS
CEC	Utiliza LLMs para inducir esquemas de eventos a partir de texto no etiquetado.	- Simplificación del proceso de inducción. - Manejo eficiente de relaciones jerárquicas y temporales. - Alta calidad y cobertura de esquemas.	Dependencia de modelos de lenguaje preentrenados.
CEF (ArcSight)	Formato de evento común utilizado para la normalización de eventos de seguridad.	- Estandarización en la industria de seguridad. - Facilita la integración con múltiples sistemas de seguridad.	- Limitado a eventos de seguridad. - Menos flexible para otros tipos de eventos.
Syslog	Protocolo estándar para el envío de	- Amplia adopción y soporte en sistemas operativos y dispositivos	- Formato de mensaje limitado.

	mensajes de registro en una red IP.	de red. - Simplicidad y eficiencia en la transmisión de mensajes.	- Falta de estructura para eventos complejos.
STIX	Estandarización para el intercambio de información sobre amenazas cibernéticas.	- Estructura detallada y rica en contexto. - Facilita el intercambio de información entre organizaciones.	- Complejidad en la implementación. - Enfoque específico en ciberseguridad.

Algunos aspectos a tomar en cuenta en la comparación de estos esquemas de eventos:

- **Calidad y Cobertura**: El CEC, al utilizar LLMs, puede inducir esquemas de alta calidad y cobertura en diversos dominios, superando las limitaciones de esquemas específicos como CEF y STIX que están más enfocados en la seguridad.
- **Flexibilidad**: A diferencia de Syslog y CEF, que tienen formatos más rígidos y específicos, el CEC ofrece una mayor flexibilidad para representar eventos complejos y sus relaciones.
- **Eficiencia**: El uso de LLMs en el CEC permite una inducción más eficiente y precisa de esquemas de eventos, mejorando la calidad de las relaciones temporales y jerárquicas en comparación con métodos tradicionales.

Principios y Arquitecturas de Orquestación Multi-Agente Basadas en LLMs y RAG

Fundamentos de RAG y Orquestación Multi-Agente

- **Agentic RAG:** Extiende RAG tradicional permitiendo ingestión dinámica de datos y razonamiento en tiempo real mediante agentes especializados (Function Calling, ReAct, LLMCompiler).

- **Descentralización:** Arquitecturas RAG descentralizadas mejoran eficiencia y privacidad, permitiendo que entidades distribuidas gestionen recuperación, augmentación y generación.
- **Mitigación de alucinaciones:** Integración de fuentes externas y protocolos de control para reducir errores y mejorar la fiabilidad.

Patrones Arquitectónicos para Respuesta en Tiempo Real

- **Modularidad y escalabilidad:** Frameworks como NetSecGame permiten escenarios multi-agente complejos y toma de decisiones secuencial sin necesidad de reentrenamiento.
- **Integración con herramientas forenses:** Orquestación de agentes LLM con herramientas como PCAP readers y sistemas de recuperación de información para análisis y atribución de incidentes.
- **Seguridad y benchmarking:** Frameworks como Agent Security Bench (ASB) formalizan ataques y defensas, evaluando vulnerabilidades en agentes LLM.

Avances hacia arquitecturas multi-agente

Frente a las limitaciones de SIEM y SOAR, la investigación en ciberseguridad se ha orientado hacia arquitecturas basadas en agentes de IA, capaces de colaborar en tareas de detección, análisis y respuesta. Estos agentes, apoyados en Modelos de Lenguaje de Gran Escala (LLMs) y Recuperación Aumentada por Generación (RAG), han mostrado mejoras notables en precisión y eficiencia.

Un ejemplo destacado es CyberRAG, un sistema que integra RAG y LLMs para la clasificación de ataques. De acuerdo con Blefari et al. (2025), esta arquitectura logra más del 94% de

precisión en entornos simulados y reduce en un 45% los tiempos de triaje en centros de operaciones de seguridad.

De manera similar, Audit-LLM introduce un mecanismo de debate entre agentes especializados para detectar amenazas internas en registros corporativos. Los resultados reportados por Song et al. (2024) muestran una reducción significativa de falsos positivos, alcanzando un rango de 3,7–6,7%, muy inferior al de seis sistemas de referencia evaluados.

Otros proyectos han explorado modelos de coordinación más complejos. El caso de Triangle, desarrollado por Tsinghua University y Microsoft, implementa agentes negociadores para la priorización de incidentes en entornos de nube. Este sistema demostró una reducción sustancial en el tiempo de toma de decisiones frente a métodos manuales (Tsinghua University et al., 2025). En paralelo, CyGATE emplea teoría de juegos para optimizar la planificación de parches y defensa adaptativa, integrando agentes basados en LLMs y RAG para anticipar los movimientos de un adversario (Jiang et al., 2025).

A continuación se presenta una comparación entre implementaciones de orquestación con ML frente a una con nuestra propuesta de n8n :

Tabla 2. Comparación en literatura vs propuesta

ASPECTO	EN EL ARTÍCULO	EN NUESTRA PROPUESTA BÁSICA
Uso de Wazuh para detección	Sí — base del artículo	Sí, la idea es recibir alertas de Wazuh
Filtrado/análisis adicional	Sí — con ML que clasifica / detecta anomalías	Sí — pero con GPT para “análisis narrativo” y criterio humano
Objetivo de disminuir ruido/falsos positivos	Sí, reducir alertas no útiles	Implícito: que GPT ayude a evitar ruido o guiar al SOC

Enfoque automático	Sí, el pipeline se automatiza	Sí, workflow automático n8n
Velocidad / tiempo real	Sí — sugiere latencias de decenas de ms	Nuestra versión básica es menos rigurosa en latencia (depende del plan del LLM etc.)

Diferencias más importantes

1. **ML vs LLM / heurísticas**
    a. El artículo se basa en modelos de ML clásicos (Random Forest, DBSCAN, etc.), que operan en base de datos / características cuantitativas.
    b. Nosotros estamos usando (o proponiendo usar) un modelo de lenguaje como GPT para análisis narrativo, interpretación de contexto, sugerencias.
    c. Los modelos ML pueden ser más consistentes, rápidos y con menor costo (cuando ya entrenados) para tareas específicas de clasificación binaria/anomalía.
2. **Precisión, tasa de falsos positivos y operacionalidad**
    a. El artículo mide métricas como precisión, recall, tasa de falsos positivos, latencia, escalabilidad, etc.
    b. En nuestra propuesta básica, no hay un ciclo formal de entrenamiento / evaluación; la calidad dependerá del diseño del prompt, del modelo de LLM y de ajustes iterativos.
3. **Requisito de datos y entrenamiento**
    a. Para ML necesitas datos históricos etiquetados, características bien definidas, proceso de entrenamiento / validación.

   b. Para la solución GPT / n8n que proponemos, no necesitas (inicialmente) una base etiquetada, sino que se basa más en interpretación “en vivo” del evento.

4. **Detección de anomalías vs explicación / triage**
   a. El artículo va más hacia la detección automática de anomalías / clasificación.
   b. Lo nuestro es más hacia el **triage asistido**, es decir: “¿es esta alerta digna de atención?”, “¿qué sugerencia damos?”, etc.
5. **Latencia / rendimiento**
   a. En producción, un modelo ML bien optimizado puede responder en pocas decenas de ms. El artículo reporta latencias dentro del umbral real-time.
   b. Usar GPT o un modelo de lenguaje puede tener latencias más altas (depende de la infraestructura, el tráfico, la cola, etc.). Para un entorno muy demandante (miles de eventos por segundo) puede no escalar sin ajustes.
6. **Escalabilidad y costos**
   a. ML bien implementado escala bien “linealmente” para muchos eventos.
   b. Usar GPT frecuentemente (por cada alerta) puede resultar en costos (uso de API, tasa de tokens) y límites operativos.
7. **Interpretabilidad vs caja negra**
   a. Los modelos ML clásicos permiten extraer “feature importance”, comprender qué variables influyeron, etc.
   b. LLM entrega explicación narrativa, pero puede ser más “caja negra” en cuanto a por qué decidió algo (aunque podemos pedir explicaciones en el prompt).

Coordinación y orquestación de agentes

Más allá del desempeño individual de cada agente, la literatura reciente enfatiza la importancia de los mecanismos de orquestación y coordinación. Herramientas como LangGraph permiten

integrar múltiples agentes en flujos colaborativos, mientras que enfoques de ChatOps facilitan la interacción en tiempo real entre humanos y sistemas automatizados. En la misma línea, CyberGuardian 2 (Paduraru, Patilea & Stefanescu, 2025) evaluó la integración de agentes LLM en redes distribuidas. Sus resultados resaltan la modularidad como una característica esencial para la escalabilidad y resiliencia, ya que permiten añadir o reemplazar agentes sin comprometer el sistema global.

Un enfoque de orquestación multi-agente basado en LLMs y RAG puede mejorar significativamente la detección, priorización y respuesta a incidentes de ciberseguridad en comparación con los sistemas tradicionales SIEM y SOAR. Esto se debe a la capacidad de estos sistemas para integrar información contextual, reducir falsos positivos mediante técnicas avanzadas de aprendizaje automático y facilitar una respuesta más dinámica y adaptable ante amenazas emergentes. Sin embargo, es importante abordar los retos asociados, como la gestión de latencia, la seguridad frente a vulnerabilidades en la inyección de prompts y la integración con herramientas de seguridad preexistentes.

A continuación se presenta un cuadro de comparación entre SIEM/SOAR con un enfoque multi-agente que implementa RAG.

Análisis Comparativo en Gestión de Incidentes de ciberseguridad entre SIEM y SOAR Tradicionales frente a Multi-Agente LLM+RAG

Tabla 3. Ventajas, Limitaciones Tradicional vs Ansible vs N8N

ASPECTO A EVALUAR	SIEM/SOAR TRADICIONAL	MULTI-AGENTE LLM + RAG
Detección de Incidentes	Basada en reglas y firmas; limitada ante amenazas nuevas; alta tasa de falsos positivos	Razonamiento contextual, integración dinámica de datos, reducción de falsos

		positivos mediante aprendizaje automático
Priorización	Priorización rígida, dependiente de reglas; limitada contextualización	Priorización adaptativa, análisis semántico y contextual, integración de fuentes heterogéneas
Respuesta	Automatización básica, requiere supervisión humana significativa	Respuesta dinámica, orquestación autónoma, integración con herramientas forenses y de seguridad
Reducción de Falsos Positivos	Mejoras con ML, pero limitada por reglas y datos heterogéneos	Aprendizaje profundo, clustering, razonamiento multi-agente, reducción significativa de alertas irrelevantes
Escalabilidad y Adaptabilidad	Escalabilidad limitada, integración compleja, adaptación lenta	Modularidad, escalabilidad nativa, adaptación en tiempo real a nuevas amenazas
Privacidad y Seguridad	Centralización, riesgos de brechas, limitaciones en privacidad	Gobernanza de datos, privacidad reforzada, integración de blockchain y MFA
Latencia y Coste	Latencia variable, costes altos en grandes despliegues	Latencia optimizable, costes ajustables según arquitectura y caching
Transparencia y Confianza	Limitada, dependiente de reporting manual	Evaluación centrada en usuario, métricas de transparencia y confianza

Sin embargo, las soluciones tradicionales aún presentan limitaciones frente al volumen, velocidad y sofisticación de las amenazas actuales, lo que ha motivado el desarrollo de arquitecturas basadas en agentes de inteligencia artificial (IA). Dentro de este enfoque a continuación se presenta una comparación entre SIEM/SOAR, Ansible y N8N.

Tabla 4. Comparacioón entre SIEM/SOAR, Ansible y N8N

ASPECTO	TRADICIONAL	ANSIBLE	N8N
Tecnología Base	Scripts manuales, shell/bash, cron jobs, SIEM/SOAR con reglas estáticas	Motor de automatización declarativo basado en YAML y Python; usa SSH, APIs REST y módulos predefinidos.	Motor de automatización visual basado en flujos de trabajo (Node.js, TypeScript), con nodos conectables por API.
Ventajas	- Control manual total. - No depende de software externo. - Fácil de implementar en entornos pequeños.	- Escalable y repetible. - Gran soporte para infraestructura como código (IaC). - Permite automatizar tareas de seguridad (patching, response, configuración).	- Interfaz visual intuitiva. - Integración rápida con APIs, bases de datos y herramientas SOC (como Wazuh o Slack). - Bajo costo y alto grado de personalización. - Ideal para orquestar agentes LLM o flujos RAG.

Limitaciones	- Requiere intervención humana constante. - Difícil de escalar y auditar. -Alta tasa de falsos positivos.	- Requiere conocimiento técnico avanzado. - Falta de interfaz visual. - Limitado en tareas cognitivas (no procesa texto ni razonamiento contextual).	- Depende de conectividad y APIs estables. - Menor control granular sobre sistemas locales. - No es nativo para automatizaciones de bajo nivel (como configuración de firewalls).

Resultados y métricas de evaluación

Los estudios revisados coinciden en que las arquitecturas multi-agente ofrecen mejoras consistentes frente a los sistemas tradicionales:

- Precisión de detección superior al 94% (Blefari et al., 2025; Alshamrani, 2025).
- Reducción de falsos positivos al rango de 3,7–6,7% (Song et al., 2024).
- Correlaciones exitosas en la priorización de incidentes (MCC=0,998) (Roelofs et al., 2024).
- Reducción del MTTR entre un 35% y 45% (Lin et al., 2025).

Estas métricas evidencian una clara ventaja frente a SIEM y SOAR, que suelen depender de análisis humanos y procesos rígidos.

Limitaciones

A pesar de los resultados prometedores, la literatura también reconoce importantes limitaciones. En primer lugar, gran parte de los estudios se desarrolla en entornos simulados, lo que limita la validez externa y dificulta la extrapolación a entornos empresariales complejos (Nyberg & Johnson, 2024). En segundo lugar, la constante evolución de los LLMs plantea un reto: modelos que hoy ofrecen resultados sobresalientes pueden volverse obsoletos en cuestión de meses, obligando a rediseñar marcos de integración constantemente.

Otro desafío identificado es la necesidad de estudios longitudinales y despliegues reales que evalúen la escalabilidad, resiliencia y costo total de propiedad de estos sistemas en ambientes de producción. Autores como Paduraru et al. (2025) y Roelofs et al. (2024) destacan que, aunque existen pruebas piloto exitosas, aún falta evidencia a gran escala para consolidar su adopción en sectores críticos.

Desafíos:

- **Fatiga de alertas:** Los analistas de los SOC suelen sufrir agotamiento debido al alto volumen de alertas, muchas de las cuales son falsos positivos.
- **Integración y personalización:** Existe dificultad para integrar diversas herramientas de seguridad y adaptarlas a las necesidades específicas de cada organización (9 10).
- **Uso de recursos:** Las herramientas de monitoreo tradicionales pueden consumir muchos recursos y, en algunos casos, no detectar vulnerabilidades de tipo zero-day (7).

Soluciones:

- **Automatización:** Implementar mecanismos automáticos y modelos de aprendizaje automático para reducir los procesos manuales y mejorar la detección de amenazas.
- **Arquitecturas avanzadas:** Utilizar marcos como λ-NF3 y NF3 para un análisis de datos y detección de anomalías más eficientes.

- **Colaboración humano-IA:** Mejorar las operaciones del SOC mediante una colaboración efectiva entre los analistas humanos y las herramientas impulsadas por inteligencia artificial.

Tendencias emergentes

IA y Aprendizaje Automático

Las tendencias emergentes en los Centros de Operaciones de Seguridad (SOC) y en la respuesta a incidentes apuntan hacia una automatización cada vez más profunda, impulsada por inteligencia artificial (IA) y aprendizaje automático. Los SOC están incorporando modelos de IA para mejorar la monitorización en tiempo real, el correlacionado de eventos y el triaje de grandes volúmenes de alertas, reduciendo el ruido y priorizando aquellas con mayor impacto potencial en el negocio (Roelofs et al., 2024; ENISA, 2023). En paralelo, las plataformas de orquestación, automatización y respuesta de seguridad (SOAR) evolucionan hacia soluciones más inteligentes, capaces de integrar múltiples herramientas heterogéneas mediante modelos semánticos y playbooks automatizados, lo que disminuye la complejidad operativa y acorta los tiempos de respuesta (Islam et al., 2019; Kremer et al., 2023).

Implementación de RAG

Esta tendencia se ve reforzada por propuestas recientes que utilizan enfoques RAG y asistentes agentivos para clasificar ataques y generar reportes de forma automatizada, integrándose de forma natural en el ecosistema del SOC (Blefari et al., 2025; Paduraru et al., 2025). Al mismo tiempo, el creciente peso del Internet of Things IoT obliga a los SOC a gestionar un inventario dinámico de activos y a comprender nuevos protocolos y patrones de tráfico, lo que amplía la superficie de exposición y exige capacidades avanzadas de descubrimiento y modelado del contexto (ENISA, 2023; CISA, 2024).

Respuesta a incidentes

En el ámbito específico de la respuesta a incidentes (IR) se observa una transición desde enfoques principalmente reactivos hacia modelos más proactivos, autónomos y socio-técnicos. La literatura reciente propone arquitecturas de defensa soportadas por aprendizaje por refuerzo multi-agente y técnicas de optimización del juego ataque-defensa, orientadas a anticipar movimientos del adversario y a recomendar estrategias de mitigación casi en tiempo real (Alshamrani, 2025; Jiang et al., 2025; Nyberg & Johnson, 2024). Paralelamente, los grandes modelos de lenguaje se exploran como "copilotos" o defensores autónomos capaces de apoyar en el análisis de logs, la correlación de eventos, la clasificación de incidentes y la recomendación de acciones de respuesta, integrándose con herramientas existentes y flujos DevSecOps (Castro et al., 2025; Brahmandam, 2025; Lin et al., 2025; Tsinghua University et al., 2025). Iniciativas como Audit-LLM y soluciones multi-agente para la detección de amenazas internas muestran cómo la colaboración entre agentes especializados puede mejorar la cobertura y la precisión en escenarios complejos (Song et al., 2024). Estas capacidades automatizadas no solo buscan reducir la carga de trabajo en contextos de escasez de talento especializado, sino también habilitar formas de aprendizaje organizacional continuo, alineadas con modelos de aprendizaje de doble bucle y marcos de gestión de incidentes como IRMA, que integran dimensiones técnicas, organizativas y humanas.

Finalmente, la conformación de equipos de respuesta a incidentes tiende a ser cada vez más flexible y adaptativa. Los equipos se forman y reconfiguran ad hoc en función del tipo de incidente, combinando expertos técnicos, responsables de negocio, perfiles legales y de comunicación, lo que permite abordar incidentes complejos desde una perspectiva verdaderamente multidisciplinar (CISA, 2024). En conjunto, estas tendencias — automatización basada en IA, uso extensivo de plataformas SOAR, integración del IoT,

defensa autónoma y enfoques socio-técnicos en IR— configuran un panorama en el que los SOC e IR evolucionan desde centros meramente reactivos hacia capacidades inteligentes, proactivas y fuertemente integradas con la estrategia organizacional de ciberseguridad.

METODOLOGÍA DE TRABAJO

La metodología adoptada en este proyecto se fundamenta en el ciclo de diseño de ingeniería, un enfoque iterativo y secuencial ampliamente utilizado en proyectos tecnológicos, que abarca desde la definición conceptual hasta la validación experimental. El propósito central de este proyecto es garantizar un desarrollo ordenado y validado en cada una de sus fases, permitiendo que los resultados obtenidos sean medibles, reproducibles y relevantes para la gestión de incidentes de ciberseguridad. A continuación, se enumerará los pasos principales que se tomarán en el desenlace del proyecto.

Revisión bibliográfica y definición conceptual

La primera fase consistió en un análisis sistemático de la literatura especializada publicada en los últimos cinco años, con el objetivo de identificar:

- Fortalezas y limitaciones de los sistemas SIEM y SOAR.
- Propuestas recientes de arquitecturas multi-agente basadas en IA.
- Taxonomías de severidad y esquemas de normalización de eventos.

Definición de Esquema Canónico de Eventos (CEC)

Este análisis permitió diseñar un CEC compuesto por los siguientes campos:

Tabla 5. CEC

CAMPO	DESCRIPCIÓN
`agent.ip`	Dirección IP del agente o equipo que genera el evento; permite identificar el origen de la actividad en red.

`agent.name`	Nombre del agente, host o endpoint que reporta el evento, útil para ubicar el activo dentro de la infraestructura.
`rule.description`	Descripción textual de la regla de detección que se ha disparado, indicando el tipo de comportamiento o amenaza detectada.
`rule.groups[]`	Lista de grupos o categorías en las que se clasifica la regla (por ejemplo, *malware*, *lateral_movement*, *credential_access*).
`rule.id`	Identificador único de la regla dentro del motor de correlación o del SIEM, usado para rastreo y mantenimiento.
`rule.level`	Nivel de severidad o criticidad asignado a la regla, empleado para priorizar la atención de alertas e incidentes.
`rule.mitre.tactic[]`	Tácticas MITRE ATT&CK asociadas a la regla, que indican el objetivo o fase de la intrusión que representa la actividad.
`rule.mitre.technique[]`	Técnicas MITRE ATT&CK específicas vinculadas a la regla, que describen cómo se lleva a cabo la acción maliciosa.
`timestamp`	Marca de tiempo en la que se genera o registra el evento, fundamental para la correlación temporal y la línea de tiempo del incidente.

Estas variables se consideran de valor para el proyecto porque permiten describir cada evento de seguridad de forma completa y accionable, combinando información sobre el origen técnico (agent.ip, agent.name), el momento exacto en que ocurrió (timestamp) y el contexto de detección (rule.id, rule.description, rule.groups[]). Además, el nivel de severidad (rule.level) facilita la priorización de incidentes, mientras que el mapeo a MITRE ATT&CK

(rule.mitre.tactic[], rule.mitre.technique[]) aporta un marco estandarizado para entender la intención y las técnicas del atacante, lo que a su vez mejora el análisis, la correlación entre alertas y el diseño de respuestas más efectivas dentro del SOC.

Levantamiento de Laboratorio SOC

El laboratorio sirve una función indispensable dentro del desarrollo de este proyecto y es probar la funcionalidad del orquestador mostrando su capacidad para reconocer ataques controlados, generar alertas y tomar una acción de respuesta bajo la ejecución exitosa de un playbook.

El levantamiento se realizó sobre una topología de red única en la subred 192.168.100.0/24, articulada alrededor de un switch Cisco SG300-28 (192.168.100.2) al que se conectan todos los componentes críticos del laboratorio. Sobre esta infraestructura se desplegó un Wazuh Manager en Windows 11 (192.168.100.103) para la gestión centralizada de eventos, un servidor n8n en Windows 11 (192.168.100.64:5678) como orquestador de flujos, un controlador de dominio Active Directory en Windows Server 2022 (192.168.100.10) y un IDS Suricata en Ubuntu 24 (192.168.100.20). La red se completa con un PC de ataques basado en Kali Linux (192.168.100.200) y tres endpoints Windows 11 con agente Wazuh instalado (192.168.100.80, 192.168.100.64 y 192.168.100.69), lo que permite simular tanto la operación corporativa como distintos escenarios de amenaza de forma controlada y reproducible. A continuación, se resumen las ips en la siguiente tabla:

Tabla 6. Direcciones ip Laboratorio SOC

NODO	IP
Subred	192.168.100.0/24
Gateway Router Wifi	192.168.100.1
SG300 Gestión	192.168.100.2
AD-SOCLAB	192.168.100.10

Suricata	192.168.100.20
Wazuh-Manager	192.168.100.103
n8n-endpoint	192.168.100.64
Kali	192.168.100.200
Endpoint Lenovo	192.168.100.69
Enpoint Acer	192.168.100.64
Endpoint Ideapad	192.168.100.80

Obtención de data

Se planificó la obtención de datos que comprendan escenarios comunes de ciberseguridad en un Laboratorio que simula un entorno de infraestructura de red corporativa básica sobre el cuál se posiciona un SOC. Se realizó primero una conexión desde una instancia n8n con SIEM de software libre Wazuh para obtener alertas y junto con la realización de ataques controlados de distinto tipo permita la conformación de una línea base de métricas orientado a usar para generar estadísticas de rendimiento. A continuación, se presenta algunos de los ataques controlados realizados:

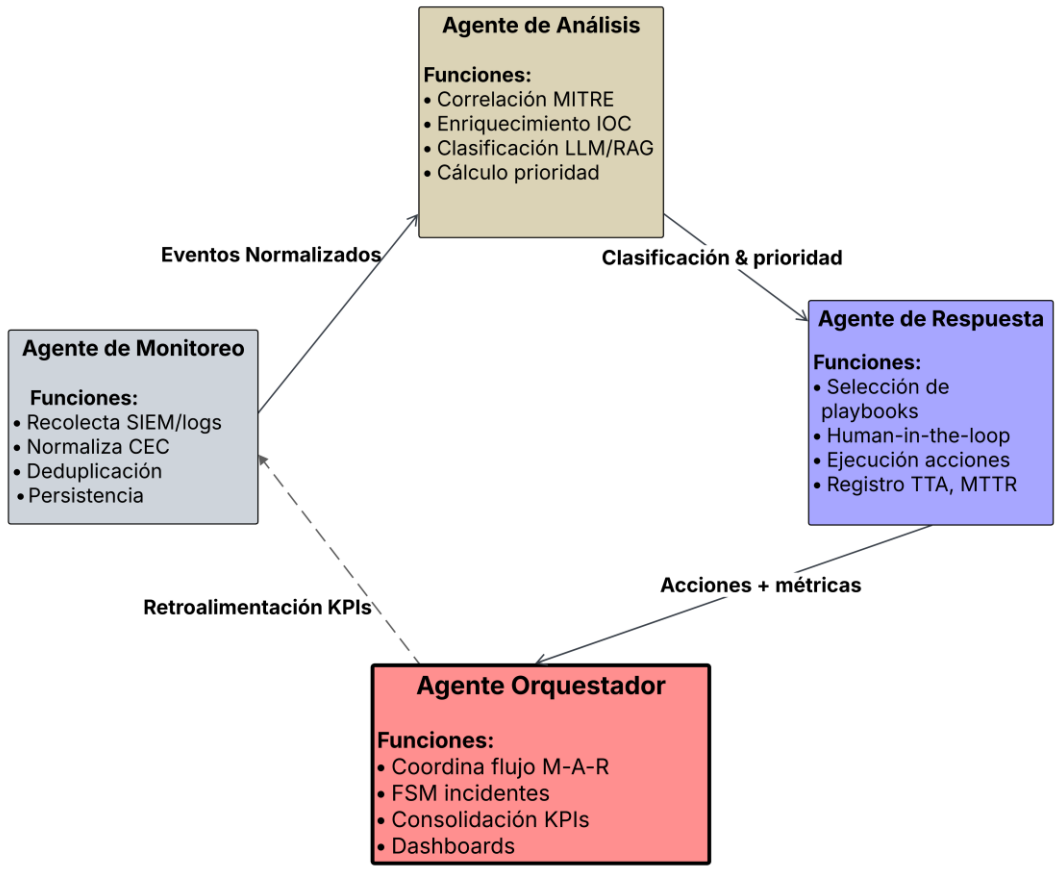
- Intentos de Autenticación SSH Fallida
- Escaneos de Puertos
- DDos
- Navegación sosprechosa (EICAR)

Cada evento fue rotulado con un nivel de severidad en una escala de 1 (informativo) a 15 (crítico), validado como verdadero positivo. Esta estrategia responde a la necesidad de contar con conjuntos de datos confiables y controlados para pruebas reproducibles (Roelofs et al., 2024).

Desarrollo incremental de agentes

A continuación se presenta el esquema de Orquestador de agentes:

Figura 1. Diagrama preliminar de Agentes



Teniendo en cuenta el diagrama de arquitectura presentado en el agente de monitoreo y se procede a describir el mismo:

En ambos flujos se puede observar un enfoque modular y jerárquico en la gestión de incidentes de ciberseguridad. El primer flujo enfatiza la etapa de **detección y análisis**, centrada en la evaluación heurística y la priorización automática, mientras que el segundo aborda la **respuesta y mitigación** mediante la integración de modelos generativos y automatización de infraestructura. La combinación de ambas arquitecturas representa un ciclo completo de

monitoreo inteligente: desde la identificación inicial del riesgo hasta la ejecución de medidas concretas de contención y registro de incidentes.

Este diseño evidencia un uso estratégico de tecnologías de inteligencia artificial y automatización de procesos, donde los modelos de lenguaje natural no reemplazan al analista humano, sino que amplifican su capacidad operativa al sintetizar, clasificar y contextualizar información compleja en tiempo real. En el entorno experimental del laboratorio de ciberseguridad descrito, estos flujos permiten recrear y gestionar ataques controlados (como DDoS o ransomware), midiendo la eficacia de las respuestas y generando datos valiosos para la mejora continua del sistema.

Se construyó de manera modular, siguiendo un esquema de desarrollo incremental, a continuación, se describen los agentes implementados:

- Agente de Monitoreo:

Se encarga principalmente de la ingesta, filtrado y normalización de los eventos provenientes del SIEM y otras fuentes, mapeando cada registro al Esquema Canónico de Eventos (CEC). Este agente garantiza la calidad de los datos al eliminar duplicados, corregir inconsistencias y enriquecer los eventos con metadatos relevantes, de modo que el resto de la plataforma trabaje siempre sobre una vista coherente y estandarizada del estado de la infraestructura.

- Agente de Análisis:

A partir de los eventos ya normalizados, aplica técnicas de Recuperación Aumentada por Generación (RAG) y Modelos de Lenguaje de Gran Escala (LLMs) para priorizar incidentes, reducir falsos positivos y generar explicaciones en lenguaje natural que describen el contexto y el impacto de cada alerta. Además, puede proponer acciones de respuesta y recomendaciones operativas, actuando como un asistente cognitivo para los analistas del SOC y facilitando la toma de decisiones frente a escenarios complejos de ciberseguridad.

- Agente de Respuesta

Es el componente encargado de ejecutar de forma coordinada y automatizada los playbooks de seguridad definidos para cada tipo de incidente, esto puede incluir acciones como el bloqueo de direcciones IP maliciosas, la cuarentena de endpoints comprometidos, el cierre temporal de puertos expuestos y la apertura de tickets de evidencia en las herramientas de gestión correspondientes (por ejemplo, sistemas de ITSM o plataformas de seguimiento de incidentes). A partir de la información priorizada por las capacidades de análisis del sistema, este agente orquesta las medidas de contención y mitigación con el menor impacto posible en la operación, asegurando además el registro detallado de cada acción ejecutada para facilitar la trazabilidad, la auditoría posterior y el aprendizaje continuo del proceso de respuesta. Cada agente será probado y validado en fases sucesivas bajo un enfoque test-driven development (TDD).

- Agente Orquestador:

Actúa como capa de gobierno y observabilidad del sistema multi-agente, consumiendo tanto los eventos de ingesta (metrics_ingest) como los resultados de análisis y respuesta (r_metrics) para consolidar indicadores clave de desempeño. Sobre esta información construye un *data mart* y vistas de KPIs que alimentan dos tipos de paneles: un *dashboard* operacional en tiempo real, orientado al seguimiento continuo de alertas, acciones ejecutadas y estado del SOC, y un *dashboard* ejecutivo enfocado en tendencias de largo plazo, donde se monitorizan métricas como MTTR, FPR, AUC o incluso estimaciones de ROI de la automatización. De este modo, el agente orquestador no interviene directamente en la detección o respuesta, sino que integra y sintetiza la información generada por los demás agentes para apoyar la toma de decisiones tácticas y estratégicas en ciberseguridad.

En primera instancia se procedió a probar el flujo con un trigger de chat para probar procesamiento, priorización y notificación de alerta. Los mensajes se construyeron de tipo json y se estableció un formato estándar para los mensajes.

Una vez compronbado el funcionamiento del flujo se procedió a realizar las configuraciones de n8n para enlazar el mismo con el SIEM Wazuh, para esto se levantó un sitio de n8n con urls https://192.168.100.64:5678/webhook-test/wauh/alert para test y https://192.168.100.64:5678/webhook/wauh/alert para producción. Con esta integración y verificando que el servidor de manager Wazuh mediante ping se comunique con el sitio n8n, el orquestador es capaz de escuchar y procesar todas las alertas reportadas por Wazuh de manera constante y automáticamente genera las alertas por canal Telegram SOC.

Integración y orquestación

Una vez construidos los agentes, estos fueron integrados en un marco de orquestación utilizando n8n como herramienta central. Se implementó un mecanismo de control denominado kill-switch (AUTO_MODE) que garantiza la reproducibilidad de las pruebas y evita la ejecución de acciones destructivas en fases preliminares del prototipo.

Implementación de Ataques

Se implementaron difrerentes escenarios de ataques con el fin de someter a la infraestructura de laboratorio a condiciones lo más similares posibles a la cotidianidad de un SOC y también considerando la severidad qué pueden presentar para generar una alerta en el sistema. Dependiendo el caso se realizó ejecuciones de comando por cmd, o archivos de código .py. Bajo este ambiente simulado y los ataques controlados fue posible demostrar la funcionalidad del orquestador. Es decir, se mostró que el orquestador fue capaz de reconocer, alertar y reaccionar frente a una amenaza. Entre estos ataques se encuentran:

- Intentos de Autenticación SSH Fallida
- Escaneos de Puertos
- DDos

- Navegación sosprechosa (EICAR)

Campaña de Pruebas

Luego de establecer escenarios de ataque se comenzó a trabajar para la generación del ground truth o línea base de conocimiento, bajo la acumulación sistemática de todas las alertas enriquecidas emitidas por el flujo n n8n con el SIEM Wazuh y almacenándolas en un formato estructurado desde un archivo exportado desde telegram (tabla/CSV). Para cada alerta se registraron tanto los campos generados automáticamente (por ejemplo, Alert Name, Timestamp, Wazuh Rule ID, Wazuh Rule Level, AI Risk Level, mapeo MITRE e IPs de origen y destino) como las etiquetas de verdad de terreno asignadas manualmente, incluyendo la clasificación final del evento (malicioso o benigno), el escenario de ataque al que pertenece y la severidad "real" utilizada como referencia. Sobre esta línea base se calcularán las métricas de rendimiento del orquestador multi-agente, comparando el nivel de riesgo estimado por el sistema frente a las etiquetas de referencia para obtener indicadores de precisión (Accuracy), área bajo la curva (AUC), tasa de falsos positivos (FPR), correlación de Kendall en la priorización de incidentes y reducción del MTTR respecto al escenario sin automatización. De esta manera, la evaluación del sistema se apoya en un conjunto de datos controlado, reproducible y explícitamente etiquetado.

El formato para las alertas de notificación es el siguiente:

Tabla 7. Formato estándarizado de alertas

CAMPO	VALOR
Alert Name	Potential Malware Infection
Alert Description	An alert triggered by suspicious activity on the network indicating a potential malware infection.
Timestamp	2021-10-15T08:35:00
Wazuh Rule ID	1001

Wazuh Rule ID Explanation	Malware infection detected
Wazuh Rule Level	10
Wazuh Rule Level Explanation	High severity alert indicating a confirmed malware infection.
AI Risk Level (1–15 + label + justification)	12 – High – Multiple indicators of compromise and high likelihood of malware infection.
Risk Comparison (Wazuh vs AI)	The Wazuh rule level of 10 aligns with the AI risk level of 12 due to the severity and confirmed nature of the malware infection.
Wazuh MITRE Mapping	T1059
Inferred MITRE Mapping (AI)	T1106, T1552
MITRE Mapping Explanation	The Wazuh rule is related to the execution of malware (T1059), while the AI also suggests data exfiltration (T1106) and use of a remote access tool (T1552).
MITRE Mapping Comparison	The Wazuh rule focuses on malware execution, while the AI expands the scope to include data exfiltration and remote access tools.
Agent / Host	Workstation123
Source IP	192.168.1.10
Destination IP	176.31.12.4
Host Machine	Workstation123
Users	UserA
External Artifacts Reputation Check	Analysis: The destination IP 176.31.12.4 has been flagged as malicious by threat intelligence sources.
Security Recommendations	Isolate Workstation123 from the network, conduct a full malware scan, and investigate for any unauthorized remote access activity.
Concise Summary	A high severity alert was triggered on Workstation123 indicating a confirmed malware infection with potential data exfiltration and remote access activity. The destination IP has been identified as malicious, requiring immediate isolation and thorough investigation.

Este mensaje representa una alerta enriquecida por contexto que combina información del motor de reglas de Wazuh y de un componente de IA. Se detallan el momento exacto del incidente, el identificador y nivel de severidad de la regla, junto con una evaluación de riesgo ampliada de la IA que justifica por qué la amenaza se considera de alto impacto. El mapeo a MITRE ATT&CK muestra tanto la táctica original (ejecución de malware) como técnicas adicionales inferidas por la IA (exfiltración de datos y acceso remoto), lo que amplía el contexto del ataque. Además, se identifican claramente el host afectado, las direcciones IP de origen y destino y el usuario involucrado, complementado con un chequeo de reputación de la IP de destino. Finalmente, se incluyen recomendaciones de seguridad concretas y un resumen ejecutivo que facilitan al analista tomar decisiones rápidas sobre aislamiento, análisis forense y seguimiento del incidente.

Obtención de Línea Base y Métricas de Rendimiento

A partir de la campaña de pruebas, se procedió a construir la línea base (*ground truth*) recopilando de manera sistemática todas las alertas enriquecidas emitidas por el flujo y almacenándolas en un formato estructurado (tabla/CSV). Para cada alerta se registraron tanto los campos generados automáticamente (por ejemplo, Alert Name, Timestamp, Wazuh Rule ID, Wazuh Rule Level, AI Risk Level, mapeo MITRE e IPs de origen y destino) como las etiquetas de verdad de terreno asignadas manualmente, incluyendo la clasificación final del evento (malicioso o benigno), el escenario de ataque al que pertenece y la severidad “real” utilizada como referencia. Sobre esta línea base se calcularán las métricas de rendimiento del orquestador multi-agente, comparando el nivel de riesgo estimado por el sistema frente a las etiquetas de referencia para obtener indicadores de precisión (Accuracy), área bajo la curva (AUC), tasa de falsos positivos (FPR), correlación de Kendall en la priorización de incidentes

y reducción del MTTR respecto al escenario sin automatización. De esta manera, la evaluación del sistema se apoya en un conjunto de datos controlado, reproducible y explícitamente etiquetado.

Evaluación del sistema

El desempeño del orquestador multi-agente será evaluado en un entorno simulado de incidentes, con la finalidad de medir indicadores clave de rendimiento (KPIs) frente a una línea base. Los indicadores definidos son:

- Precisión (Accuracy) $\geq 0{,}80$.
- Área bajo la curva (AUC) $\geq 0{,}80$.
- Tasa de falsos positivos (FPR) $\leq 10\%$.
- Correlación de Kendall (τ) $\geq 0{,}6$ en la priorización de incidentes.
- Reducción del MTTR (Mean Time to Resolution) $\geq 30\%$.

La metodología descrita busca asegurar que el desarrollo del orquestador multi-agente no solo cumpla los objetivos de reducción de tiempos y falsos positivos, sino que también establezca un marco de validación alineado con estándares de reproducibilidad académica y práctica profesional en ciberseguridad.

DESARROLLO DEL PROTOTIPO (TIPO 1)

Esta etapa busca demostrar la viabilidad técnica de un orquestador multi-agente de IA aplicado a la gestión de incidentes de ciberseguridad.

Esquema Canónico de Eventos (CEC)

La base del prototipo es el Esquema Canónico de Eventos (CEC), diseñado para garantizar la normalización de registros de seguridad heterogéneos. El CEC está compuesto por los siguientes campos:

Tabla 8. CEC

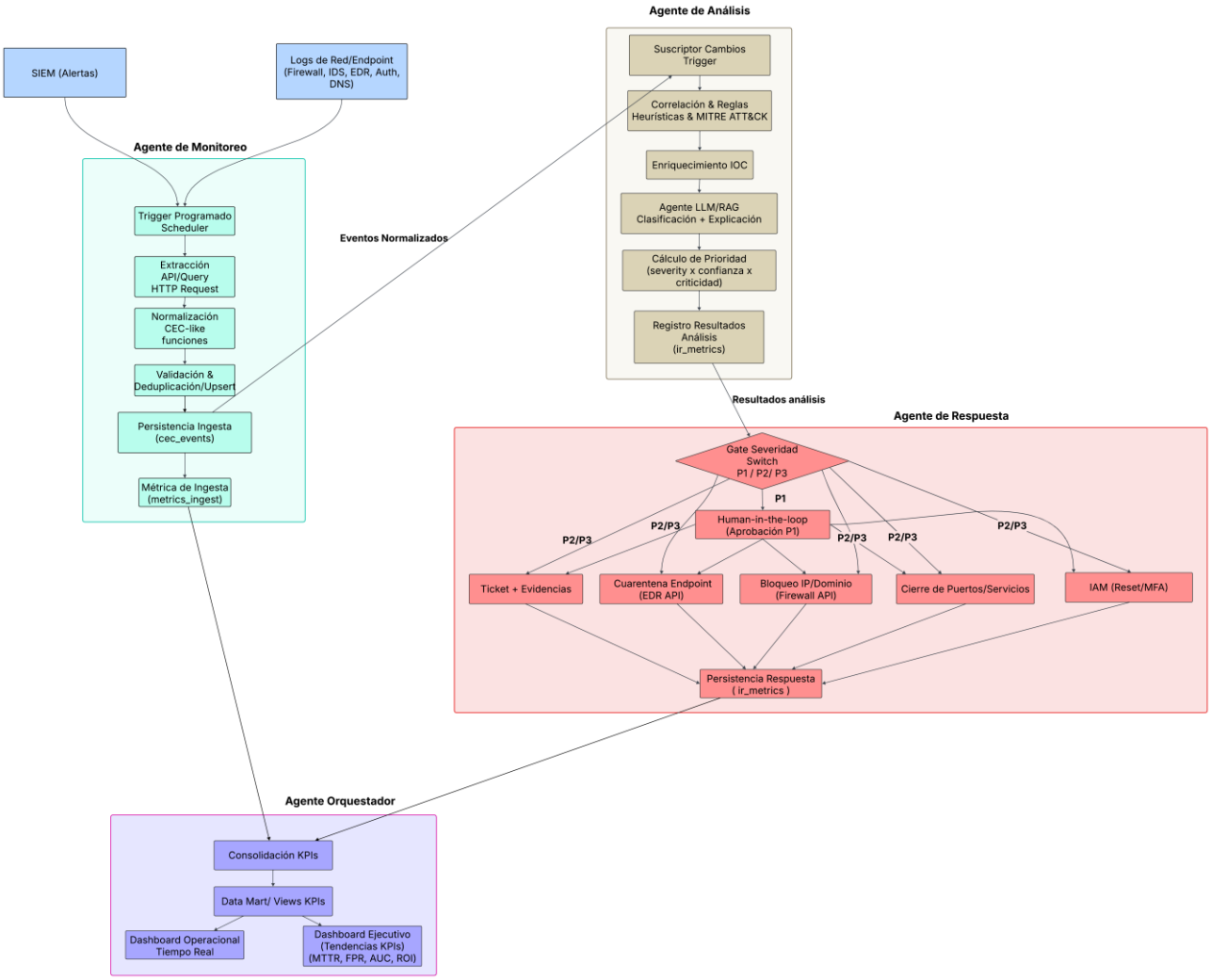
CAMPO	DESCRIPCIÓN
`agent.ip`	Dirección IP del agente o equipo que genera el evento; permite identificar el origen de la actividad en red.
`agent.name`	Nombre del agente, host o endpoint que reporta el evento, útil para ubicar el activo dentro de la infraestructura.
`rule.description`	Descripción textual de la regla de detección que se ha disparado, indicando el tipo de comportamiento o amenaza detectada.
`rule.groups[]`	Lista de grupos o categorías en las que se clasifica la regla (por ejemplo, *malware*, *lateral_movement*, *credential_access*).
`rule.id`	Identificador único de la regla dentro del motor de correlación o del SIEM, usado para rastreo y mantenimiento.
`rule.level`	

	Nivel de severidad o criticidad asignado a la regla, empleado para priorizar la atención de alertas e incidentes.
`rule.mitre.tactic[]`	Tácticas MITRE ATT&CK asociadas a la regla, que indican el objetivo o fase de la intrusión que representa la actividad.
`rule.mitre.technique[]`	Técnicas MITRE ATT&CK específicas vinculadas a la regla, que describen cómo se lleva a cabo la acción maliciosa.
`timestamp`	Marca de tiempo en la que se genera o registra el evento, fundamental para la correlación temporal y la línea de tiempo del incidente.

Estas variables se consideran de valor para el proyecto porque permiten describir cada evento de seguridad de forma completa y accionable, combinando información sobre el origen técnico (agent.ip, agent.name), el momento exacto en que ocurrió (timestamp) y el contexto de detección (rule.id, rule.description, rule.groups[]). Además, el nivel de severidad (rule.level) facilita la priorización de incidentes, mientras que el mapeo a MITRE ATT&CK (rule.mitre.tactic[], rule.mitre.technique[]) aporta un marco estandarizado para entender la intención y las técnicas del atacante, lo que a su vez mejora el análisis, la correlación entre alertas y el diseño de respuestas más efectivas dentro del SOC.

A continuación, se presenta un Diagrama de Orquestador en baja resolución:

Figura *2*. Flujograma Preliminar de Orquestador

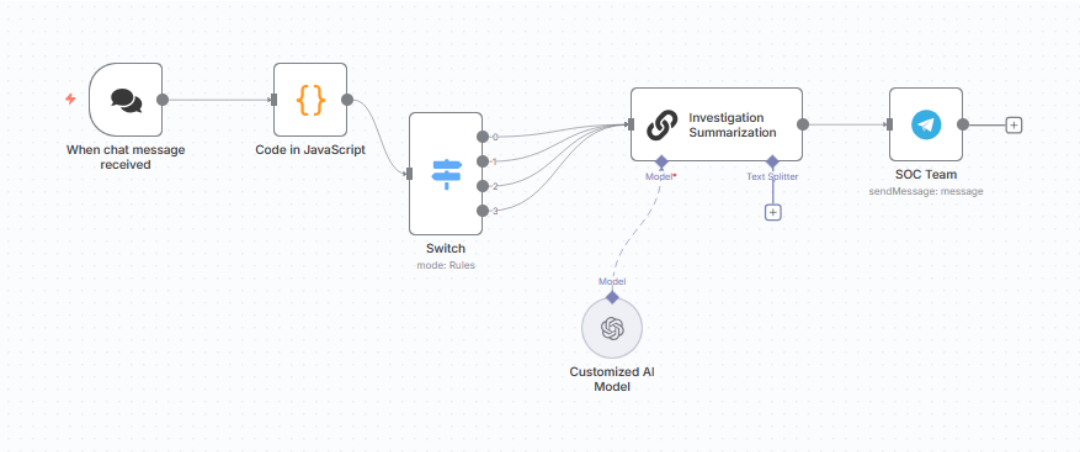


Se procedió a la construcción de un nuevo flujo de trabajo a partir de la estructura:

Agente de Monitoreo en n8n

El primer agente implementado es el de Monitoreo, cuya función principal es la ingesta y normalización de datos en el formato definido por el CEC. Se realizaron los primeros ensamblajes para pruebas de funcionamiento con un trigger de chat, sin SIEM Wazuh con el fin de evaluar la capacidad de notificación de telegram y la configuración correcta de los nodos. A continuación, se presenta el flujo de monitoreo y análisis preliminar que se usó para esta primera fase:

Figura 3. Flujo preliminar con webchat hacia SOC



Decripción del flujo por nodo.

El primer flujo corresponde a la implementación de un agente de monitoreo que recibe datos a través de un nodo de entrada conversacional y los procesa para clasificar y priorizar incidentes de seguridad. Este flujo comienza con el nodo **"When chat message received"**, que actúa como el punto de entrada de datos, permitiendo recibir tanto texto plano como estructuras JSON desde un canal conversacional. Esta flexibilidad es útil para integrar diferentes tipos de entradas, como mensajes humanos o eventos generados por otros sistemas de monitoreo.

Posteriormente, el flujo se dirige al nodo **"Code in JavaScript"**, que contiene la lógica principal de análisis y clasificación. Este bloque ejecuta un script que interpreta los datos entrantes y calcula una serie de métricas relevantes: tasa de paquetes por segundo, número de direcciones IP de origen únicas, duración del evento y nivel de criticidad del activo afectado. A partir de estos valores, el sistema estima una puntuación de riesgo compuesta y determina una prioridad categorizada en cuatro niveles (Low, Medium, High, Critical). Esta etapa incorpora además una bandera denominada *HITL* (Human In The Loop) que marca los casos que requieren revisión humana debido a la baja confianza del modelo o alto impacto potencial.

Nodos Agente de Análisis

El resultado de este procesamiento pasa al nodo **"Switch"**, que actúa como un enrutador condicional en función del nivel de prioridad determinado. En este punto se diferencian los caminos según la criticidad del incidente, permitiendo personalizar las acciones posteriores de acuerdo con la gravedad detectada.

En todos los casos, el flujo continúa hacia el nodo **"Investigation Summarization"**, que corresponde a un proceso de análisis y síntesis automatizada usando técnicas de *chain summarization* de lenguaje natural. Este nodo, apoyado en el modelo definido en **"Customized AI Model"**, el cual genera un reporte estructurado de investigación con base en las variables de entrada, siguiendo un formato predefinido que incluye nombre y descripción de la alerta, táctica y técnica MITRE, alcance, reputación de artefactos externos y recomendaciones de seguridad.

El informe producido se envía mediante el nodo **"SOC Team"**, encargado de la comunicación directa con el equipo de operaciones de seguridad a través de Telegram. Este paso permite asegurar una distribución inmediata del resumen de incidentes hacia el canal operativo, garantizando que la información crítica llegue a los analistas en tiempo real.

En conjunto, este flujo funciona como un agente de monitoreo inteligente capaz de recibir entradas dinámicas, calcular niveles de riesgo de forma autónoma, resumir la información mediante un modelo de lenguaje y distribuir los resultados a un canal de comunicación operativo. La integración de procesamiento numérico, clasificación heurística y lenguaje natural permite automatizar tareas que tradicionalmente requerían intervención manual, optimizando la velocidad de respuesta ante alertas de seguridad.

A continuación, se presenta una captura de la compilación de este flujo con el siguiente mensaje de entrada de prueba:

"metrics": {

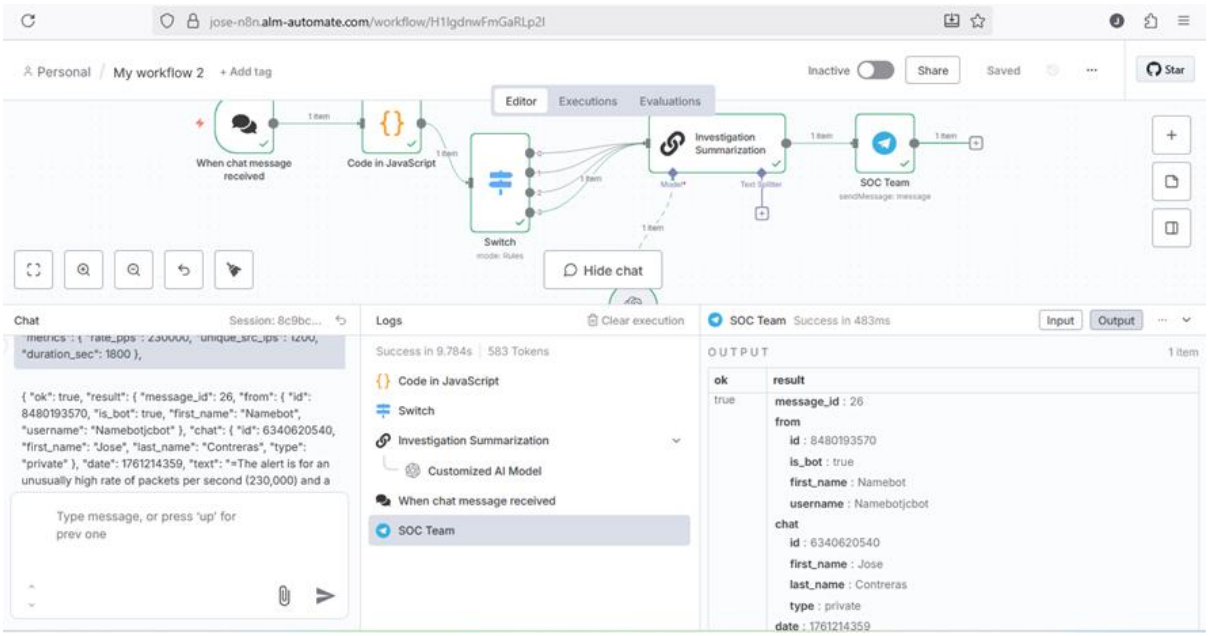
"rate_pps": 230000,

"unique_src_ips": 1200,

"duration_sec": 1800

},

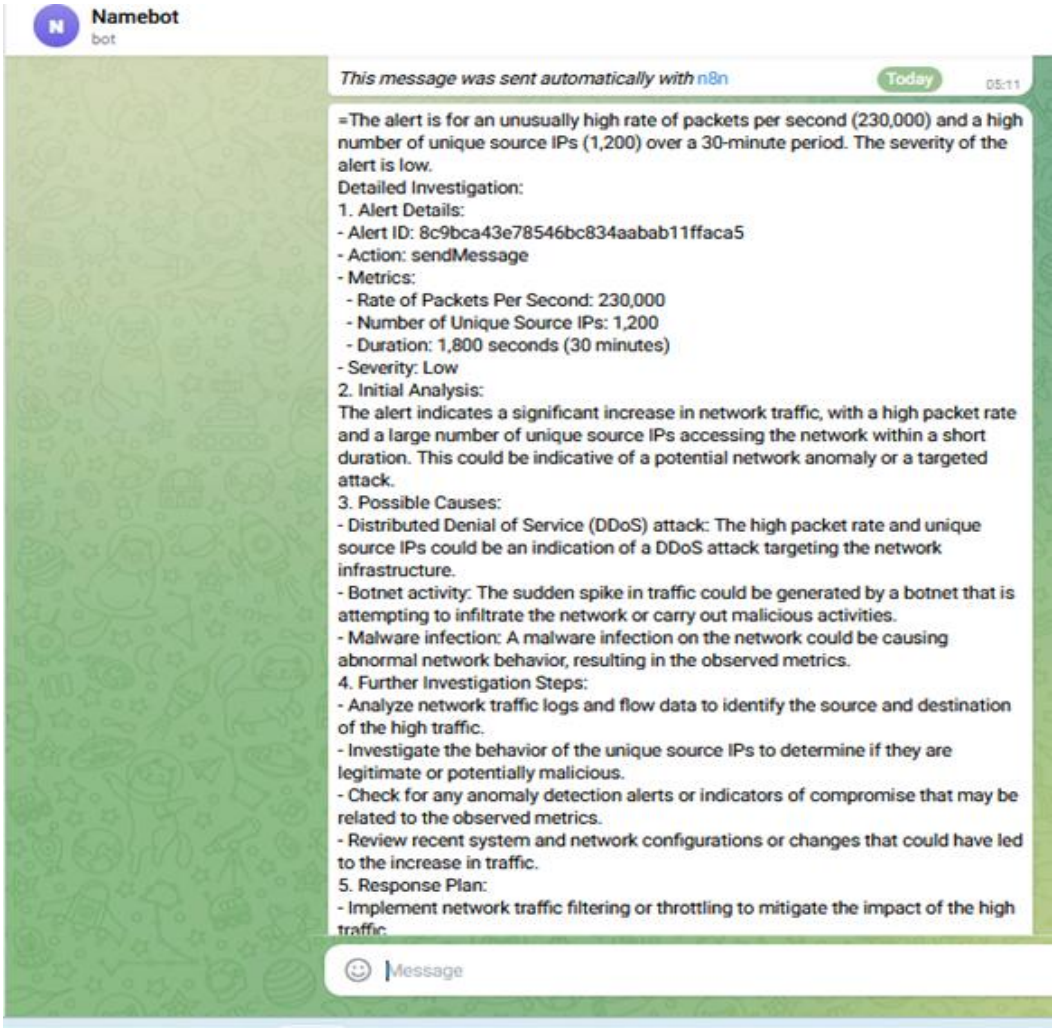
Figura 4. Compilación flujo con mensaje prueba



La imagen muestra una ejecución de prueba del flujo de trabajo en n8n, donde un mensaje de chat recibido dispara una cadena de nodos que incluye código en JavaScript, un nodo *Switch* para aplicar reglas, un componente de análisis con MTTRE de “Investigation Summarization” basado en un modelo de IA personalizado y, finalmente, el envío de una notificación al equipo SOC por Telegram. En la parte inferior se observan el registro del chat —con una alerta sobre una tasa inusual de paquetes por segundo— y el panel de salida del

nodo “SOC Team”, que confirma el envío exitoso del mensaje al bot y al usuario configurado, evidenciando el funcionamiento extremo a extremo del orquestador. A continuación, se presenta el mensaje de telegram.

Figura 5. Mensaje de salida Telegra SOC



La alerta mostrada en el canal de Telegram del SOC corresponde a un incremento inusual de tráfico de red, con una tasa de 230 000 paquetes por segundo y alrededor de 1 200 direcciones IP de origen únicas durante un periodo de 30 minutos, clasificada inicialmente con severidad baja. El mensaje incluye un detalle de la alerta (ID, acción y métricas), seguido de un análisis inicial que interpreta este pico como un posible síntoma de anomalía de red o de un ataque dirigido. A continuación, se enumeran causas posibles, entre ellas un ataque de

denegación de servicio distribuido (DDoS), actividad de botnet o incluso una infección de malware que provoque el comportamiento anómalo observado. La alerta también propone pasos de investigación adicionales —revisión de logs y flow data, análisis del comportamiento de las IPs únicas y verificación de otras alertas relacionadas— y finaliza con un plan de respuesta que sugiere aplicar filtrado o limitación de tráfico para mitigar el impacto mientras se completa la investigación.

Afinación agente de monitoreo y automatización de alertas con Wazuh

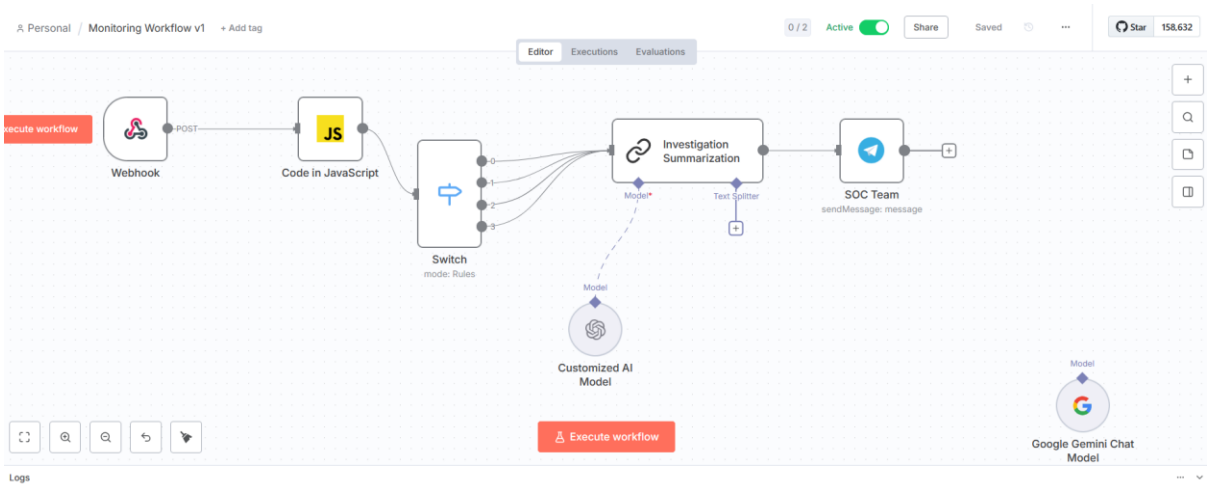
Luego de la fase preliminar de pruebas de funcionamiento se realizó la integración con el SIEM Wazuh mediante un nodo webhook a través de urls de test y producción:

https://192.168.100.64:5678/webhook-test/wazuh/alert

https://192.168.100.64:5678/webhook/wazuh/alert

De esta forma el flujo escucha permanentemente por el puerto 5678 a las alertas generadas por Wazuh. A continuación, se presenta un flujo de trabajo con esta integración:

Figura 6. Flujo de Trabajo con integración SIEM Wazuh



Con esta versión integrada y manteniendo el flujo de trabajo como activo, este es capaz de de escuchar permanentemente por el puerto 5678 al SIEM Wazuh y sin necesidad de ejecutar el workflow.

En el agente de monitoreo se realizó una revisión nodo por nodo con el fin de robustecer el funcionamiento, estandarizar los mensajes de notificación e implementar el paradigma de MITTRE&Attack. En términos del funcionamiento se intervinó especialmente al de investigación & resumen. El script en principio mantiene 2 prompts. El primero que ejecuta el MapReduce y el segundo que implementa un Summarization. Para el MapReduce se modificó el prompt con el fin de que tome en espcial consideración las siguientes variables:

Tabla 9. Variables de interés incluidas en flujo

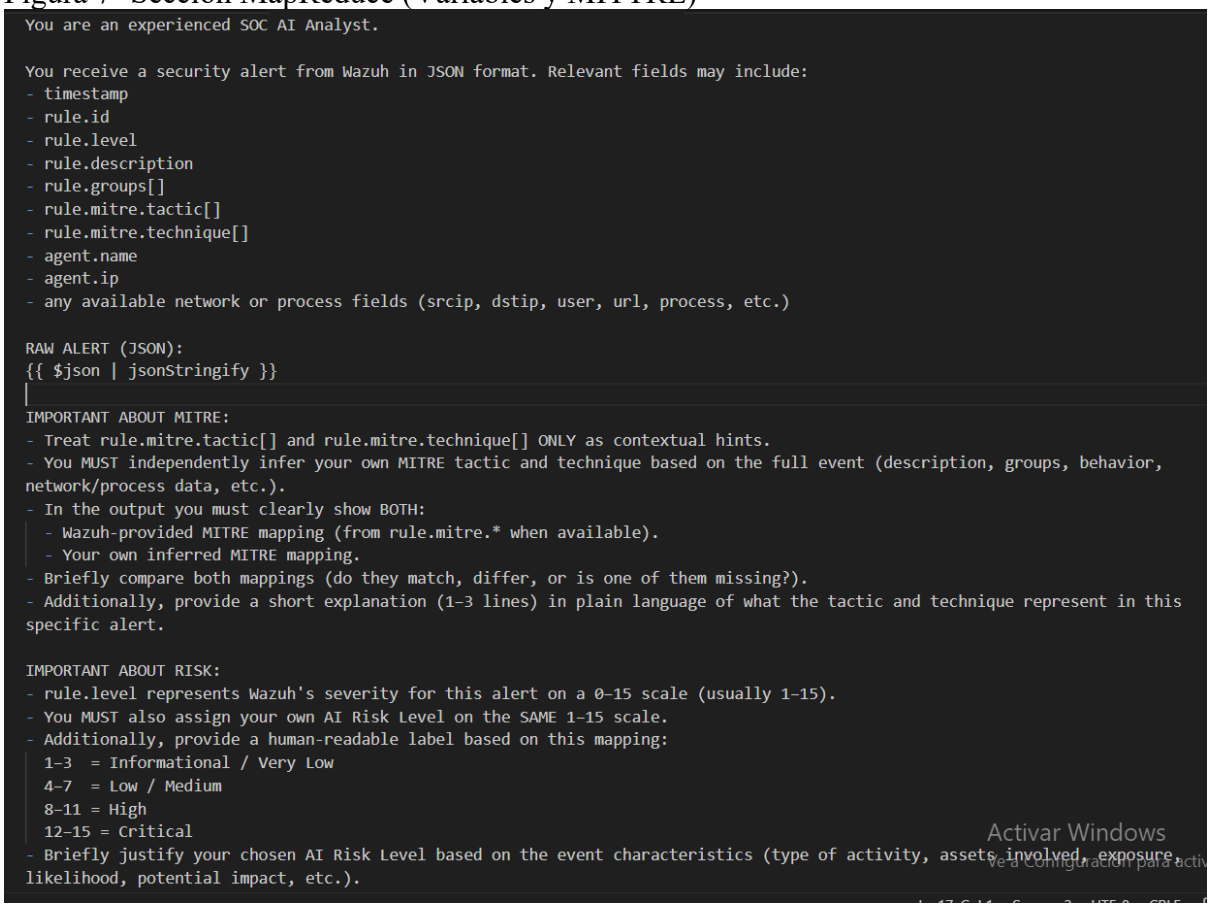
VARIABLE	DESCRIPCIÓN
timestamp	Fecha y hora exacta en la que Wazuh registró el evento/alerta (normalmente en UTC).
rule.id	Identificador numérico único de la regla de Wazuh que se disparó.
rule.level	Nivel de severidad de la alerta (normalmente 0–15 en Wazuh).
rule.description	Texto descriptivo de la regla que resume qué se ha detectado (p. ej. “Multiple failed login attempts”).
rule.groups[]	Lista de “grupos” o categorías a las que pertenece la regla (auth, ssh, malware, windows, etc.).
rule.mitre.tactic[]	Tácticas MITRE ATT&CK asociadas a la regla (p. ej. Credential Access, Discovery).
rule.mitre.technique[]	Técnicas MITRE ATT&CK asociadas (p. ej. T1110 Brute Force, T1046 Network Service Discovery).
agent.name	Nombre del agente Wazuh que generó el evento (normalmente el hostname del endpoint).

El flujo toma en consideración rule.mitre.tactic[] y rule.mitre.technique[], valores entregados por el siem Wazuh para utilizarlos de enriquecimiento contextual. Sin embargo, el

flujo implementa de forma independiente MITTRE, también genera valores de riesgo y los compara para presentar el mensaje de notificación en telegram. Dentro del prompt de summarization se consideró la estandarization de los mensajes para evitar la variablidad de contexto entregado por la IA para diferentes alertas. A continuación se muestran secciones de los prompts en cuestión, el código completo se muestra en anexos:

Prompt MapReduce

Figura 7- Sección MapReduce (Variables y MITTRE)



```
You are an experienced SOC AI Analyst.

You receive a security alert from Wazuh in JSON format. Relevant fields may include:
- timestamp
- rule.id
- rule.level
- rule.description
- rule.groups[]
- rule.mitre.tactic[]
- rule.mitre.technique[]
- agent.name
- agent.ip
- any available network or process fields (srcip, dstip, user, url, process, etc.)

RAW ALERT (JSON):
{{ $json | jsonStringify }}

IMPORTANT ABOUT MITRE:
- Treat rule.mitre.tactic[] and rule.mitre.technique[] ONLY as contextual hints.
- You MUST independently infer your own MITRE tactic and technique based on the full event (description, groups, behavior,
network/process data, etc.).
- In the output you must clearly show BOTH:
  - Wazuh-provided MITRE mapping (from rule.mitre.* when available).
  - Your own inferred MITRE mapping.
- Briefly compare both mappings (do they match, differ, or is one of them missing?).
- Additionally, provide a short explanation (1–3 lines) in plain language of what the tactic and technique represent in this
specific alert.

IMPORTANT ABOUT RISK:
- rule.level represents Wazuh's severity for this alert on a 0–15 scale (usually 1–15).
- You MUST also assign your own AI Risk Level on the SAME 1–15 scale.
- Additionally, provide a human-readable label based on this mapping:
  1–3  = Informational / Very Low
  4–7  = Low / Medium
  8–11 = High
  12–15 = Critical
- Briefly justify your chosen AI Risk Level based on the event characteristics (type of activity, assets involved, exposure,
likelihood, potential impact, etc.).
```


En esta sección del prompt se define el rol del modelo como analista SOC y se especifica el tipo de alerta que recibirá desde Wazuh en formato JSON, indicando los campos relevantes (timestamp, regla, nivel, MITRE, agente, IP, etc.). También se fijan las reglas sobre cómo debe tratar el mapeo MITRE y la evaluación de riesgo: se pide al modelo inferir sus

propias tácticas y técnicas MITRE, además de las provistas por Wazuh, y asignar un nivel de riesgo en escala 1–15 con una justificación clara. De este modo, se establece el marco para que cada "map" produzca un análisis estructurado y comparable de una alerta individual. Luego, se detallan las preguntas cuyo objetivo es brindar contexto y comparación al análisis.

Figura 8. Sección MapReduce (Preguntas de Contexto y Comparación)

```
- In the output you will also explain in 1–2 sentences what it means for this alert to have:
  - Wazuh Rule Level = X/15
  - AI Risk Level = Y/15
  in terms of severity and urgency.
- Finally, compare Wazuh's rule.level against your AI Risk Level.

TASK:
1. Understand what this alert means and why it was triggered.
2. Identify from the event:
   - A clear and human-readable alert name.
   - A natural-language description of what is happening.
   - The timestamp and Wazuh rule ID.
   - Wazuh severity level (rule.level) and what it represents.
   - Your own AI Risk Level (1–15 + label + short justification).
   - The impacted host, IP addresses and any users involved.
   - Wazuh's MITRE mapping (if provided).
   - Your own inferred MITRE mapping (tactic and technique, with IDs if possible).
3. Explain:
   - What the Wazuh Rule ID corresponds to (what type of behavior or detection logic it represents).
   - What the Wazuh Rule Level and your AI Risk Level mean in practice (severity/urgency).
   - What the MITRE tactic and technique mean in the context of this alert.
4. Compare:
   - Wazuh rule.level vs your AI Risk Level.
   - Wazuh MITRE mapping vs your inferred MITRE mapping.
5. Provide a structured investigation report using the exact format below.
6. If some fields are missing in the JSON, explicitly state "Not available".
```

Aquí el prompt detalla los pasos de razonamiento que el modelo debe seguir para entender la alerta en contexto. Se le pide identificar un nombre legible para la alerta, una descripción en lenguaje natural, la severidad según Wazuh y la severidad estimada por la IA, los activos afectados, el mapeo MITRE de Wazuh y el inferido por el propio modelo. Luego debe explicar qué representa la regla, qué implican los niveles de severidad en términos de urgencia y qué significan las tácticas y técnicas MITRE en ese incidente concreto, para finalmente comparar los niveles de riesgo y producir un informe de investigación estructurado, rellenando los campos faltantes con "Not available" si es necesario. Con el análisis realizado

se pasará al siguiente prompt de Investigation & Summarization que trata del appending o de la combinación de estos resultados de análisis y de resumirlos para preparlos para notificación.

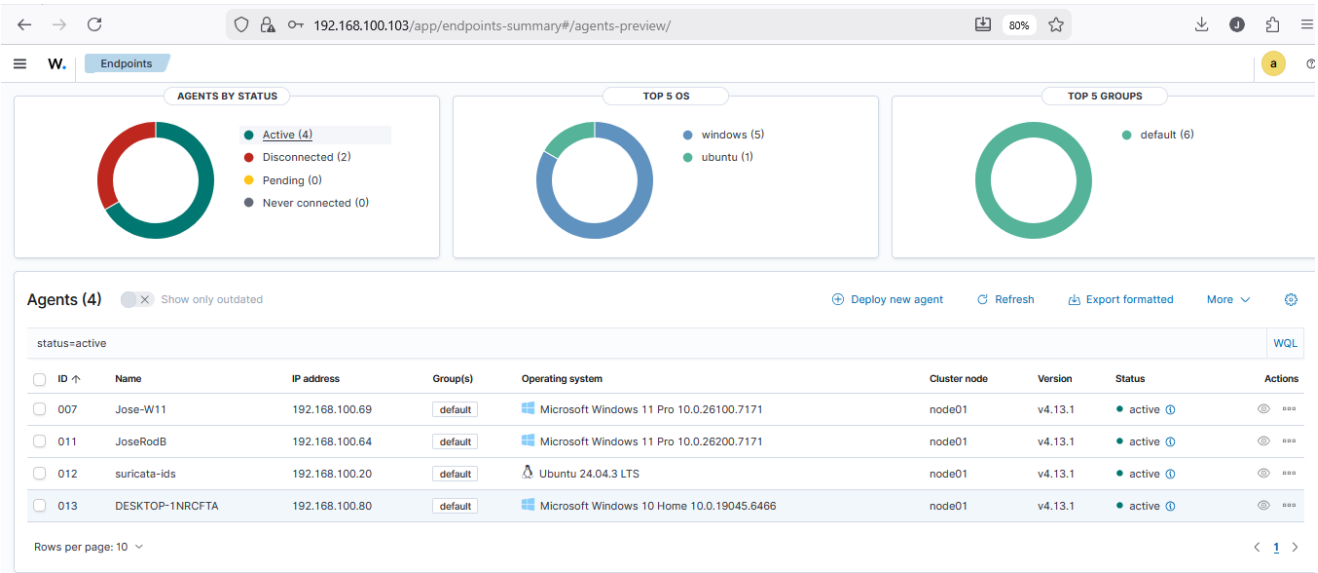
Figura 9. Prompt Combine and Resume

```
You are a senior SOC Analyst.

You will receive one or more partial analyses of the SAME alert, all in plain text.
Your job is to:
- Merge them into a single, consistent investigation report.
- Keep the exact field structure defined below.
- Remove duplicated or contradictory information.
- Resolve conflicts using the most consistent and well-justified explanation.
- Make the language clear, concise and technically accurate.
- Preserve and reconcile both:
  - Wazuh-provided values (rule.level, MITRE mapping, rule.id).
  - AI-inferred values (AI Risk Level, inferred MITRE mapping, explanations).

Below you have the previous analyses:

{{ $json.body }}

Now produce a single, final report using the following format EXACTLY (ENGLISH, one field per line, no markdown bullets, no extra sections):

Final Output:

Alert Name:
Alert Description:
Timestamp:
Wazuh Rule ID:
Wazuh Rule ID Explanation:
Wazuh Rule Level:
Wazuh Rule Level Explanation:
AI Risk Level (1–15 + label + short justification):
Risk Comparison (Wazuh rule.level vs AI Risk Level):

Wazuh MITRE Mapping (if available):
Inferred MITRE Mapping (AI):
MITRE Mapping Explanation:
MITRE Mapping Comparison:

Agent / Host:
Impacted Scope: (Source IP, Destination IP, Host Machine, Users)

External Artifacts Reputation Check:
Analysis:
Security Recommendations:

Concise Summary:
```


En esta última sección se configura un segundo rol de “senior SOC Analyst” cuyo objetivo es combinar varios análisis parciales de la misma alerta en un único informe coherente. El prompt indica que debe unificar la estructura, eliminar duplicados o contradicciones, resolver conflictos con explicaciones bien fundamentadas y conservar tanto los valores originales de Wazuh como los inferidos por la IA. A partir del bloque json.body con los análisis previos, el modelo debe generar un reporte final en un formato fijo de campos (Alert Name, Risk Comparison, MITRE Mapping, recomendaciones, resumen conciso, etc.), funcionando como fase de “reduce” que consolida toda la evidencia en una salida única y lista para el SOC.

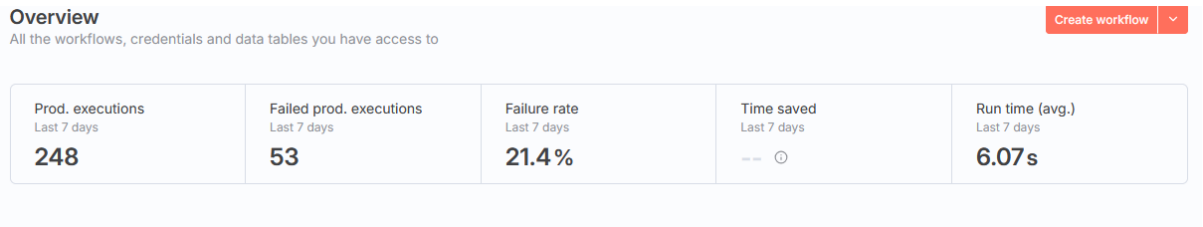
Además de habilitar un url de producción en n8n con el fin de que el flujo se integre con el Siem Wazuh de tal forma que sea capaz de escuchar permanentemente por el puerto 5678, se acopló la escala de riesgo como Wazuh de 1-15 para evitar confunsiones y procesar sus alertas acorde. También se incorporaron nuevos endpoints que ahora se reflejan en el dashboard, entre ellos se encuentra el ids-suricata, ideapad 110s bajo nombre DESKTOP-1NRCFTA y Acer win11. A continuación, se presenta una imagen con el estado actual del dashboard Wazuh.

Figura 10. Estado actual del Dashboard



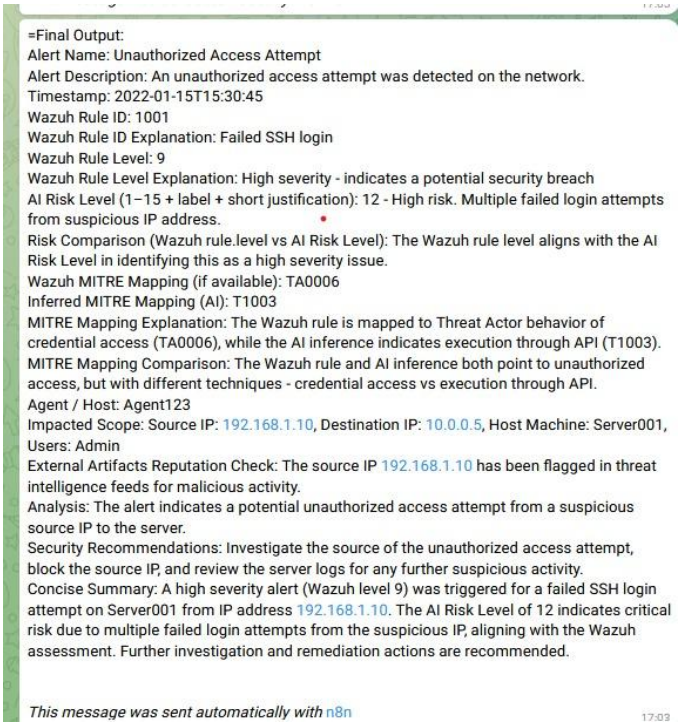
En cuanto al historial de ejecuciones del flujo de trabajo, se presenta un resumen de las mismas en N8N y 1 mensajes de alerta con formato estandarizado:

Figura 11. Resumen de Ejecuciones



Se han realizado 248 ejecuciones, de las cuáles en el chat de mensajería de telegram llamado Namebot se encuentran alrededor de 200 alertas analizadas por el agente. Dentro de este historial se observa en la figura que el tiempo de ejecución es en promedio 6.07s con una tasa de error de 21.4%.

Figura 12. Mensaje de alerta con formato estándarizado



Como se observa en la imagen se ha complementado el formato inicial con algunos cambios como comparaciones en el nivel de riesgo entre el flujo y Wazuh como también en Mapeo MITTRE. También se busca con la configuración del prompt proveer una mayor explicación sobre las características del ataque y el detalle de las sugerencias que el sistema

realiza para mitigar el riesgo. En este ejemplo, se trata de una alerta generada luego de múltiples intentos de loggeo mediante ssh hacia el usuario administrador del manager Wazuh. La explicación provista por el nodo detalla sobre múltiples intentos de loggeo desde una ip desconocida, en este caso 192.168.1.10 siendo externa a su red. El nivel de riesgo se clasifica como considerable alto, designándolo con 12 dentro de la escala de 1-15. En cuánto a las sugerencias el sistema sugiere mayor investigación acerca de la ip de origen realizando una revisión el los log del servidor.

Implementación de Laboratorio de red básica corporativa para SOC

Levantamiento servidores

Servidor Wazuh

Se implementó un servidor Wazuh Manager en una máquina virtual preconfigurada con sistema Wazuh v4.13.1 OVA para levantar el servidor. A continuación, se detalla el procedimiento:

1. Actualización del sistema base:

```
Sudo apt update && sudo apt upgrade -y
```

2. Activación de servicios:

```
sudo systemctl start wazuh-manager
sudo systemctl enable wazuh-manager
sudo systemctl status wazuh-manager
```

3. Acceso a la interfaz web del Dashboard:

Dentro del navegador de preferencia, se direcciona hacia la IP donde se encuentra alojado el servidor. En este caso:

https://<192.168.100.98>

Por otra parte, se realizaron una variedad de configuraciones entre ellas:

AD/Windows Server

El controlador de dominio AD-SOCLAB se implementó sobre Windows Server 2025, configurándolo como servidor de directorio activo, autenticación y DNS interno del laboratorio. En este nodo se definió el dominio corporativo de pruebas soclabs.local, se crearon cuentas de usuario y grupos básicos y se habilitaron políticas de seguridad elementales, simulando el contexto de una red empresarial real. Además, se registraron en el dominio los endpoints Windows monitoreados por Wazuh, de modo que los incidentes generados durante las pruebas pudieran vincularse a identidades de usuario y activos concretos, enriqueciendo así el contexto de las alertas. Para ver ejemplos y la máquina virtual se presentan imágenes en Anexo B.

IDS/Suricata

El sensor IDS se desplegó con Suricata sobre una máquina Ubuntu, conectada a un puerto espejo del switch Cisco para recibir una copia del tráfico de red relevante. Se configuró Suricata con un conjunto de reglas actualizado para la detección de escaneos, intentos de explotación y actividad de malware, habilitando el registro detallado de eventos en formato JSON. Se configuró el puerto 8 del switch como espejo con el fin de tener visibilidad del tráfico. Estos eventos pueden ser enviados al Wazuh Manager para su correlación centralizada, permitiendo contrastar las detecciones basadas en host con la visibilidad a nivel de red y obteniendo así una vista más completa de cada escenario de ataque. Vease anexo C.

Kali Linux

La máquina Kali Linux se utilizó como equipo de ataque controlado dentro del laboratorio SOC. Con las aplicaciones y herramientas del sistema de auditoría y pruebas de penetración, empleadas para generar escaneos de puertos, intentos de autenticación fallida, tráfico DDoS simulado y descargas de malware de prueba, entre otros. Todas estas actividades se ejecutaron de forma planificada y documentada, utilizando la IP fija 192.168.100.200 como origen del tráfico malicioso, lo que facilitó posteriormente la identificación de los eventos asociados y la validación de las reglas de detección en Wazuh y Suricata. Veáse Anexo D

Wazuh Endpoints

Los endpoints Wazuh se implementaron sobre tres estaciones de trabajo Windows 11 que representan equipos de usuario final dentro de la red corporativa simulada. En cada una de ellas se instaló y registró el agente de Wazuh apuntando al Wazuh Manager, habilitando la monitorización de logs del sistema, eventos de seguridad, procesos, integridad de ficheros y actividad de red local. A estos hosts se les asignaron direcciones IP fijas (192.168.100.69, 192.168.100.64 y 192.168.100.80) y se integraron en el dominio AD-SOCLAB, de modo que pudieran servir como blancos de los ataques generados desde Kali y, al mismo tiempo, como fuentes de eventos de host para la construcción del *ground truth* y la evaluación del desempeño del SOC.

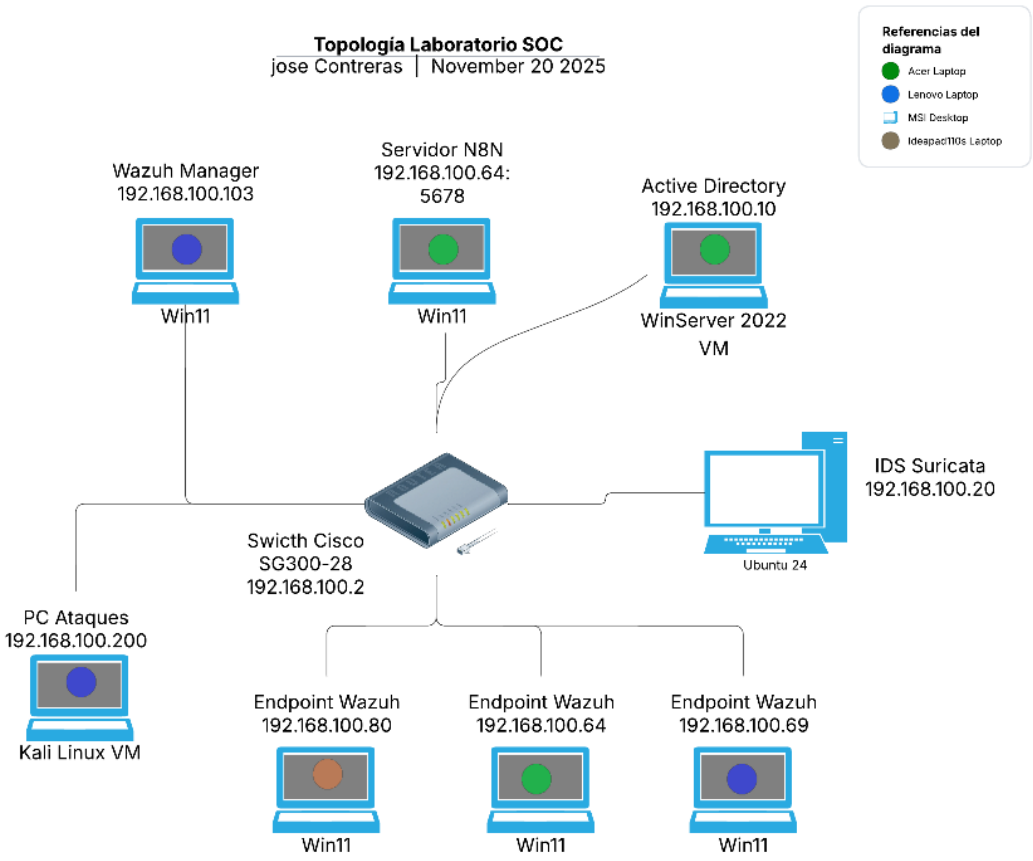
A continuación, se presenta una tabla con las respectivas Ip fijas que se presentan en la topología de laboratorio SOC finalizada:

Tabla 10. Direcciones Ip Laboratorio SOC

NODO	IP
Subred	192.168.100.0/24
Gateway Router Wifi	192.168.100.1
SG300 Gestión	192.168.100.2
AD-SOCLAB	192.168.100.10
Suricata	192.168.100.20
Wazuh-Manager	192.168.100.103
n8n-endpoint	192.168.100.64
Kali	192.168.100.200
Endpoint Lenovo	192.168.100.69
Enpoint Acer	192.168.100.64
Endpoint Ideapad	192.168.100.80

Con esta designación de Ips se procedió a implementar las respectivas configuraciones con el fin de implementar la siguiente topología:

Figura 13. Topología final de Laboratorio SOC



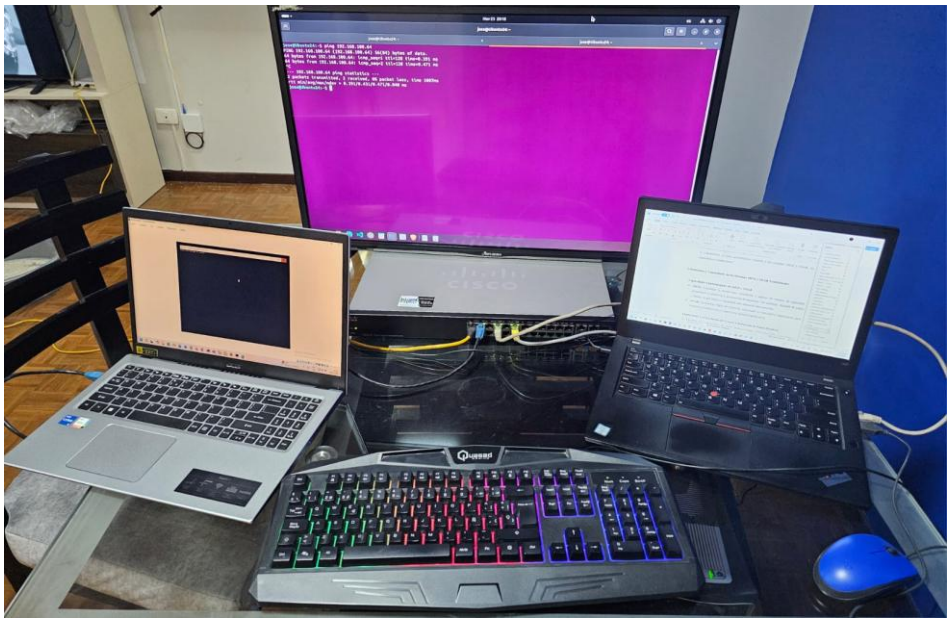
Descripción de la topología y de los ataques que nos permite hacer con la integración que tenemos:

La topología del laboratorio SOC está organizada en torno a un switch Cisco SG300-28 que actúa como núcleo de la red cableada, al que se conectan todos los nodos críticos: el servidor Wazuh Manager en Windows 11, que centraliza los logs y las alertas; el servidor N8N en otro Windows 11, que orquesta las automatizaciones y notificaciones; el controlador de dominio y servidor DNS Active Directory sobre Windows Server 2022; y el sensor IDS Suricata desplegado en Ubuntu 24, encargado de inspeccionar el tráfico en modo espejo desde el switch.

Desde este núcleo, se ramifican tres endpoints Windows 11 con el agente de Wazuh instalado (Acer, Lenovo y el Ideapad), que representan estaciones de trabajo de usuario final y sirven como blanco de pruebas de seguridad, mientras que una máquina Kali Linux funciona como PC de ataques para generar escaneos, intentos de intrusión y malware de prueba. Toda la infraestructura comparte el mismo segmento de red 192.168.100.0/24 con direcciones IP fijas para los servicios principales, lo que facilita la correlación de eventos y la medición de métricas de ataque y respuesta dentro del entorno controlado de laboratorio.

A continuación, se presenta la topología en hardware:

Figura 14. Topología de Laboratorio SOC



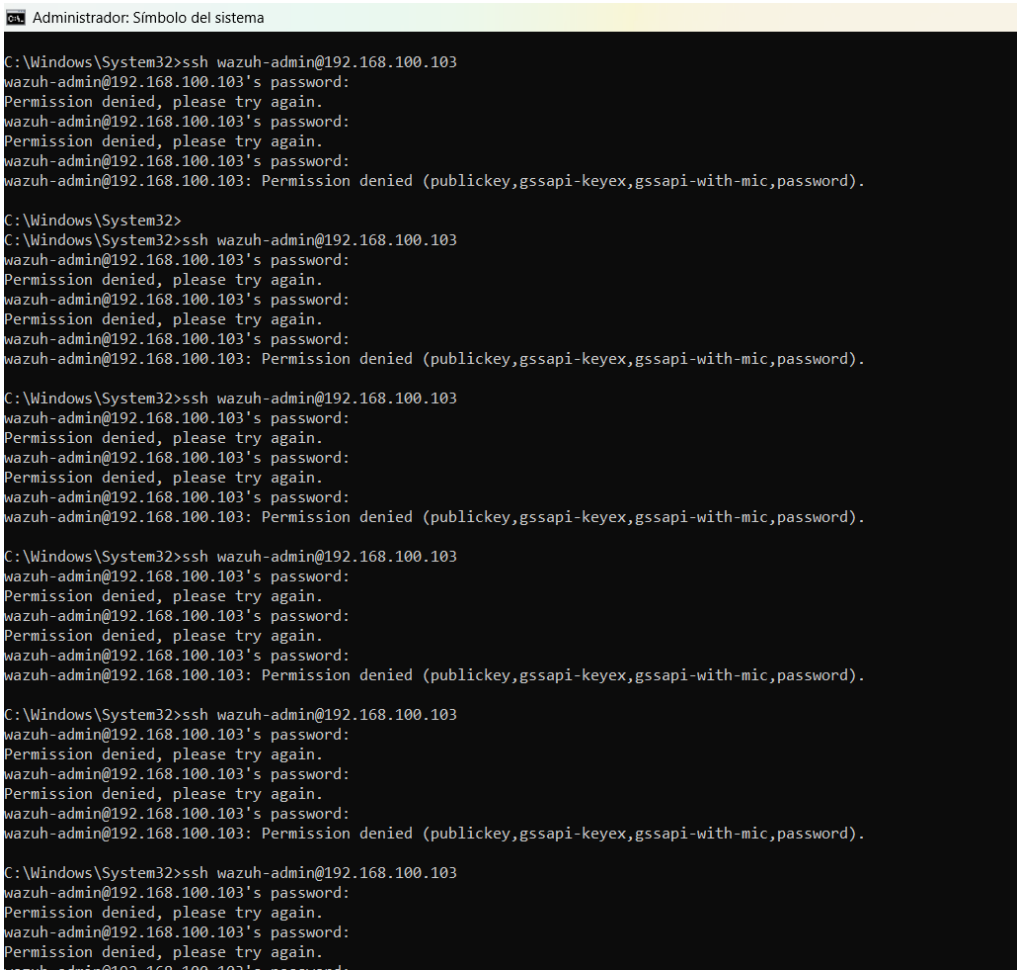
A continuación se procede a tratar la presentación de los tipos de ataques empleados en el laboratorio SOC.

Elaboración de Ataques y campaña de pruebas

Ataque de prueba 1: Intento de autenticación fallido – Wazuh Manager

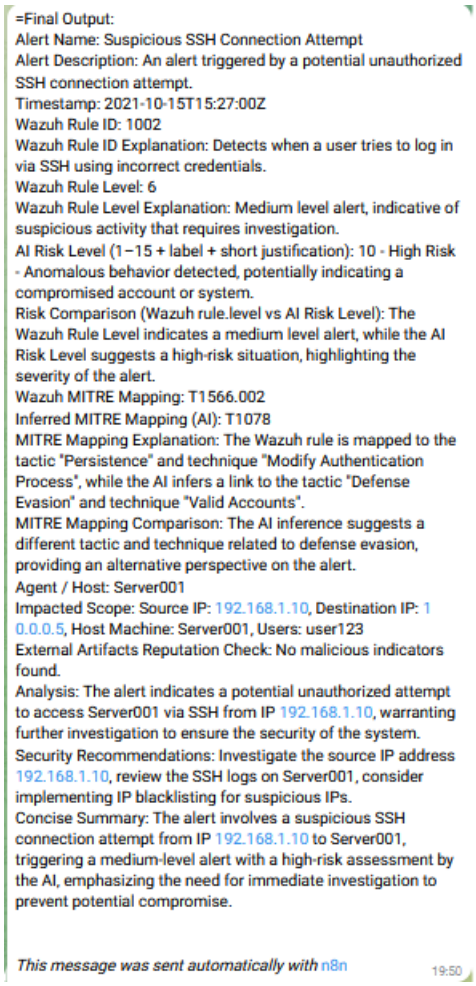
En este caso se utilizará la línea de comandos y el protocolo ssh para realizar un intento de ingreso bajo fuerza bruta con claves incorrectas. Desde la terminal con ip: 192.168.100.64 se corren llamadas ssh hacia el Manager Wazuh con ip 192.168.100.103. A continuación, se presenta una imagen con algunos de los intentos:

Figura 15. Intentos de loggeo SSH a Wazuh Manager



En la figura se observa las llamadas ssh que se realizan

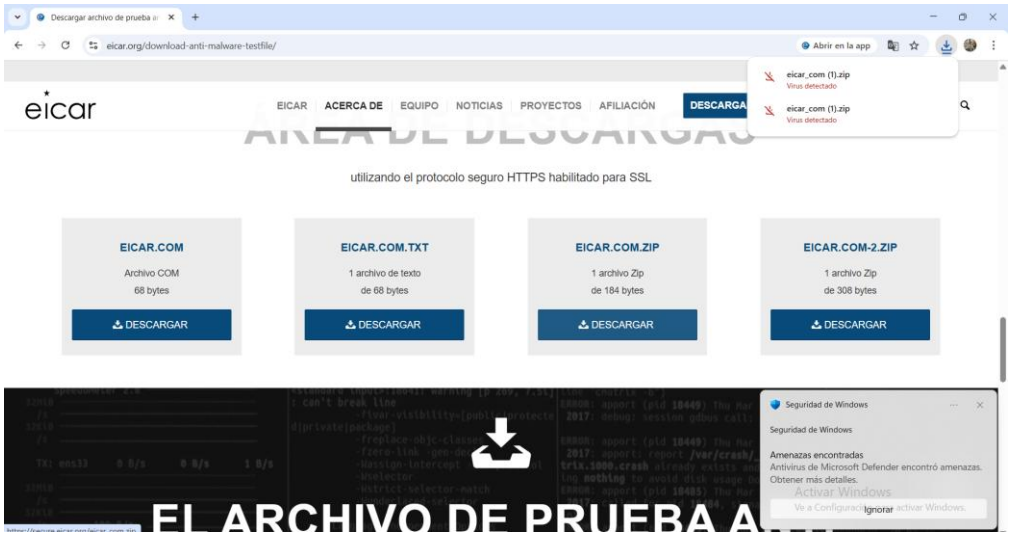
Figura 16. Mensaje alerta SOC telegram



Descripción en detalle del mensaje en narrativa continua

Ataque 2: Descarga y locación de archivo dañino EICAR

Figura 17. Descarga Archivo testfile en win11



Luego de descargar, desde la url: https://www.eicar.org/download-anti-malware-testfile/ el programa EICAR para Windows 11. Al descomprimirlo y guardar estos cambios en el sistema operativo se generó una alerta de un troyano de red detectado además de la alerta de Windows defender. A continuación se muestra el mensaje:

Figura 18. Mensaje de alerta EICAR al SOC

=Final Output:
Alert Name: Network Trojan Detected
Alert Description: An alert was triggered due to suspicious network activity consistent with a Trojan infection.
Timestamp: timestamp when the alert was triggered
Wazuh Rule ID: 1001
Wazuh Rule ID Explanation: Detects a potential Trojan infection based on network traffic analysis.
Wazuh Rule Level: 10
Wazuh Rule Level Explanation: High severity due to the significance of a Trojan infection.
AI Risk Level (1–15 + label + short justification): 12 - High risk of data exfiltration based on the behavior observed.
Risk Comparison (Wazuh rule.level vs AI Risk Level): The Wazuh rule level of 10 aligns with the AI risk level of 12 indicating a high severity alert.
Wazuh MITRE Mapping (if available): T1045 - Software Packing
Inferred MITRE Mapping (AI): T1001 - Data Exfiltration
MITRE Mapping Explanation: The Wazuh rule initially mapped to software packing, but the AI identified data exfiltration as the primary behavior.
MITRE Mapping Comparison: The Wazuh rule's original mapping to software packing differs from the AI's identification of data exfiltration.
Agent / Host: Server01
Impacted Scope: Source IP: 192.168.1.10, Destination IP: 10.0.0.5, Host Machine: Server01, Users: N/A
External Artifacts Reputation Check: No indications of malicious behavior were found in external artifact reputation checks.
Analysis: The alert was triggered by abnormal network traffic patterns consistent with a Trojan infection. Further investigation is recommended to identify the extent of the compromise.
Security Recommendations: Isolate the affected host from the network, conduct a thorough malware scan, and review system logs for any additional indicators of compromise.
Concise Summary: The alert indicates a potential Trojan infection on Server01, with high-risk levels for data exfiltration. Further investigation and remediation actions are necessary to contain the threat and prevent data loss.

Integración con agentes de análisis y respuesta

Si bien el agente de monitoreo constituye la primera fase del prototipo, este fue diseñado para integrarse con módulos adicionales:

- Agente de Análisis: aplicará clasificación mediante técnicas RAG+LLM, reduciendo falsos positivos y priorizando eventos críticos (Blefari et al., 2025; Song et al., 2024).
- Agente de Respuesta: ejecutará playbooks automatizados, incluyendo bloqueo de direcciones IP, cuarentena de dispositivos y apertura de tickets de investigación (Jiang et al., 2025).

La modularidad de este diseño permite añadir o reemplazar agentes sin comprometer el funcionamiento del sistema completo, siguiendo enfoques de escalabilidad y resiliencia ya probados en literatura (Paduraru et al., 2025).

A continuación, se presenta los pasos de implementación de un playbook orientado a gestionar un ataque de denegación de servicio:

DDoS Attack Incident Response Playbook

Step 1: Identification

- Monitor network traffic.
- Run the mitigation script (mitigate_ddos.py).

Step 2: Containment

- Implement rate limiting.
- Block malicious IPs.

Step 3: Eradication

- Update firewall rules.
- Use DDoS protection services.

Step 4: Recovery

- Restore normal traffic flow.
- Monitor for additional attacks.

Step 5: Lessons Learned

- Conduct a post-incident review.
- Update security policies.

Link de Github: https://github.com/Josep94-bot/orchestrator-n8n.git

CONCLUSIONES

Presenta los aportes de este trabajo con base en lo investigado, es importante que como autor puedas analizar el tema y su relevancia para la profesión dentro del contexto nacional e internacional (presenta similitudes, diferencias entre los diferentes enfoques del tema investigado). En el caso de presentaciones artísticas o creativas se debe describir de qué se tratan y justificar sus elementos, obligatoriamente incluir anexos con fotos, evidencias (partituras, enlaces a videos, etc.) del producto elaborado. Realiza un análisis de lo que has aprendido en este trabajo, incluye sugerencias de estudios posibles que se realicen en el futuro para comprender de mejor manera el tema, menciona alguna dificultad que hayas tenido para realizar este trabajo y sus razones.

REFERENCIAS BIBLIOGRÁFICAS

Alshamrani, A. (2025). Federated hierarchical MARL for zero-shot cyber defense. PLoS ONE. https://doi.org/10.1371/journal.pone.0329969

Blefari, F., Cosentino, C., Pironti, F. A., Furfaro, A., & Marozzo, F. (2025). CyberRAG: An agentic RAG cyber attack classification and reporting tool. arXiv. https://doi.org/10.48550/arXiv.2507.02424

Brahmandam, B. A. (2025). AI driven ChatOps for DevSecOps: Automating security incident response. International Journal of Multidisciplinary Research in Science, Engineering and Technology, 8(2), 85–96. https://doi.org/10.15680/ijmrset.2025.0802085

Castro, S. R., Campbell, R., Lau, N., Villalobos, O., Duan, J., & Cardenas, A. A. (2025). Large language models are autonomous cyber defenders. Conference on Algebraic Informatics. https://doi.org/10.1109/CAI64502.2025.00195

European Union Agency for Cybersecurity (ENISA). (2023). ENISA threat landscape 2023. Publications Office of the European Union. https://www.enisa.europa.eu/publications

Islam, C., Babar, M. A., & Nepal, S. (2019). Automated interpretation and integration of security tools using semantic knowledge. In Advanced Information Systems Engineering (pp. 529–544). Springer. https://doi.org/10.1007/978-3-030-21290-2_32

Jiang, Y., Oo, N., Meng, Q., Lin, L., Niyato, D., Xiong, Z., Lim, H. W., & Sikdar, B. (2025). CyGATE: Game-theoretic cyber attack-defense engine for patch strategy optimization. IEEE. https://doi.org/10.1109/CSR61664.2024.10679456

Kremer, R., Wudali, P. N., Momiyama, S., Araki, J., Furukawa, J., Elovici, Y., & Shabtai, A. (2023). IC-SECURE: Intelligent system for assisting security experts in generating playbooks for automated incident response. arXiv. https://doi.org/10.48550/arXiv.2311.03825

Lin, X., Zhang, J., Deng, G., Liu, T., Liu, X., Yang, C., Guo, Q., & Chen, R. (2025). IRCopilot: Automated incident response with large language models. arXiv. https://doi.org/10.48550/arXiv.2505.11901

Nyberg, J., & Johnson, P. (2024). Structural generalization in autonomous cyber incident response with message-passing neural networks and reinforcement learning. Computer Science Symposium in Russia. https://doi.org/10.1109/CSR61664.2024.10679456

Paduraru, C., Patilea, C., & Stefanescu, A. (2025). CyberGuardian 2: Integrating LLMs and agentic AI assistants for securing distributed networks. International Conference on Evaluation of Novel Approaches to Software Engineering. https://doi.org/10.5220/0013406000003928

Roelofs, T.-M., Bárbaro, E., Pekarskikh, S., Orzechowska, K., Kwapien, M., Tyrlik, J., Smadu, D., van Eeten, M., & Zhauniarovich, Y. (2024). Finding harmony in the

noise: Blending security alerts for attack detection. ACM Symposium on Applied Computing. https://doi.org/10.1145/3605098.3635981

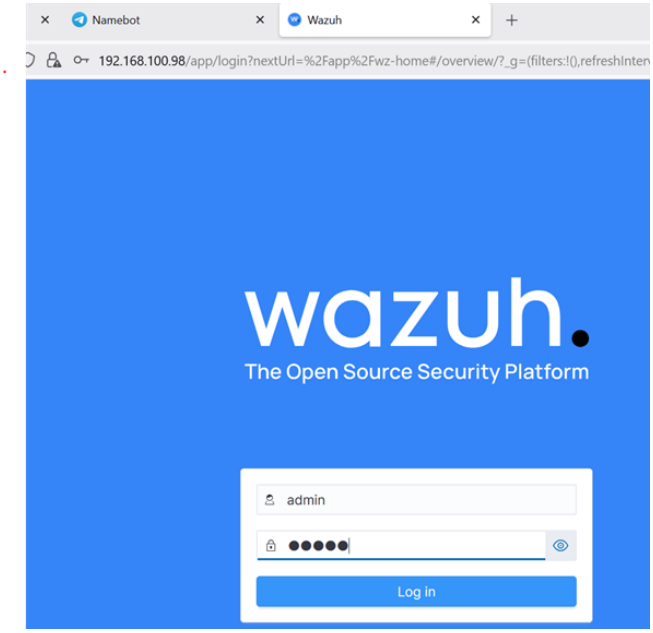
Song, C., Ma, L., Zheng, J., Liao, J., Kuang, H., & Yang, L. (2024). Audit-LLM: Multi-agent collaboration for log-based insider threat detection. arXiv. https://doi.org/10.48550/arXiv.2408.08902

Tsinghua University, Georgia Tech, & Microsoft. (2025). Triangle: Empowering incident triage with Multi-LLM-Agents. arXiv. https://doi.org/10.48550/arXiv.2505.20945

U.S. Cybersecurity and Infrastructure Security Agency (CISA). (2024). CISA cyber incidents report 2024. U.S. Department of Homeland Security. https://www.cisa.gov

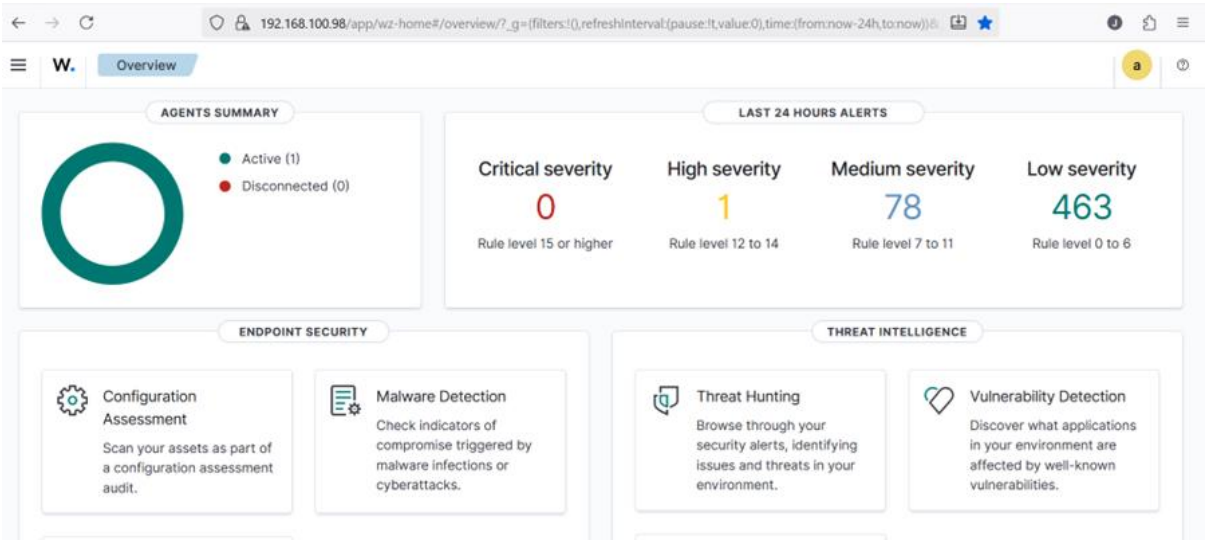
ANEXO A: SERVIDOR WAZUH

A. Acceso a Dashboard



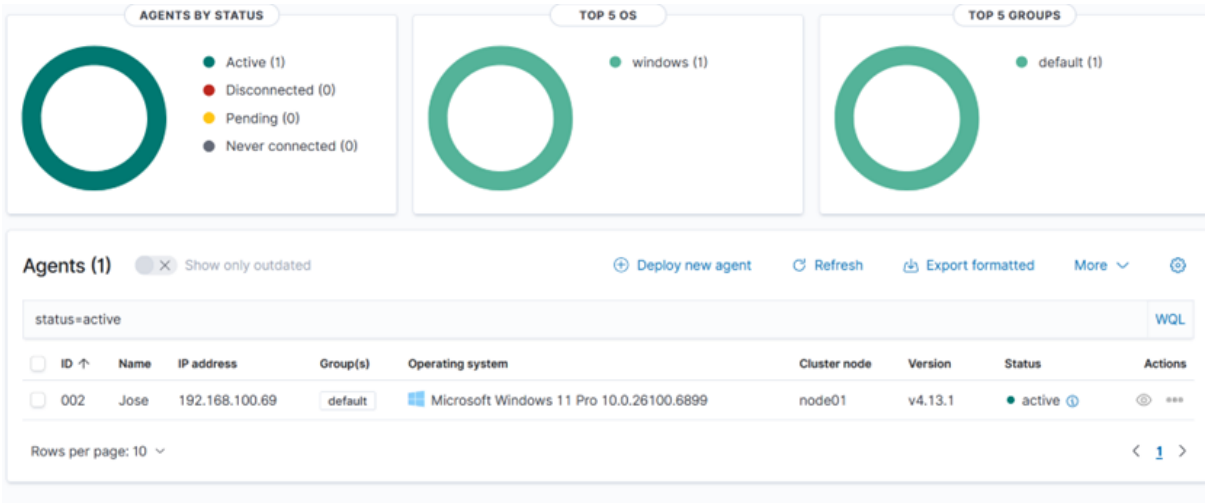
Posteriormente, el dashboard procederá a cargarse presentando los datos actuales al momento:

B. Dashboard Wazuh



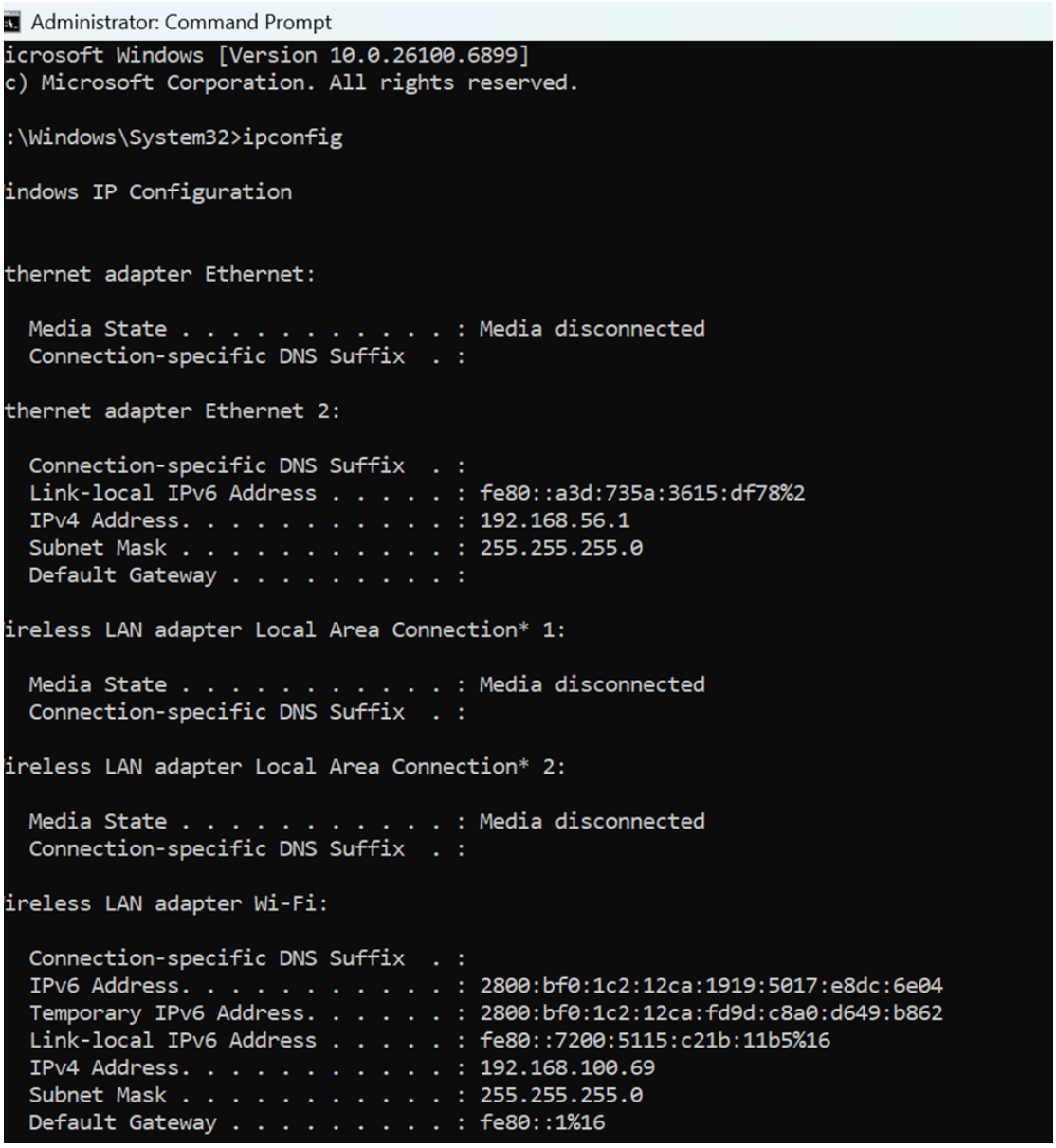
Una vez dentro es importante verificar los detalles de conexión del endpoint, inicialmente se procedió con una máquina con Windows 11, entonces seleccionando en el recuadro de Agent Summary a los equipos activos se tiene:

C. Detalle de conexión Endpoint



Finalmente, se verifica que la IP descrita sea igual a la de la máquina del usuario:

D. IP de máquina cliente



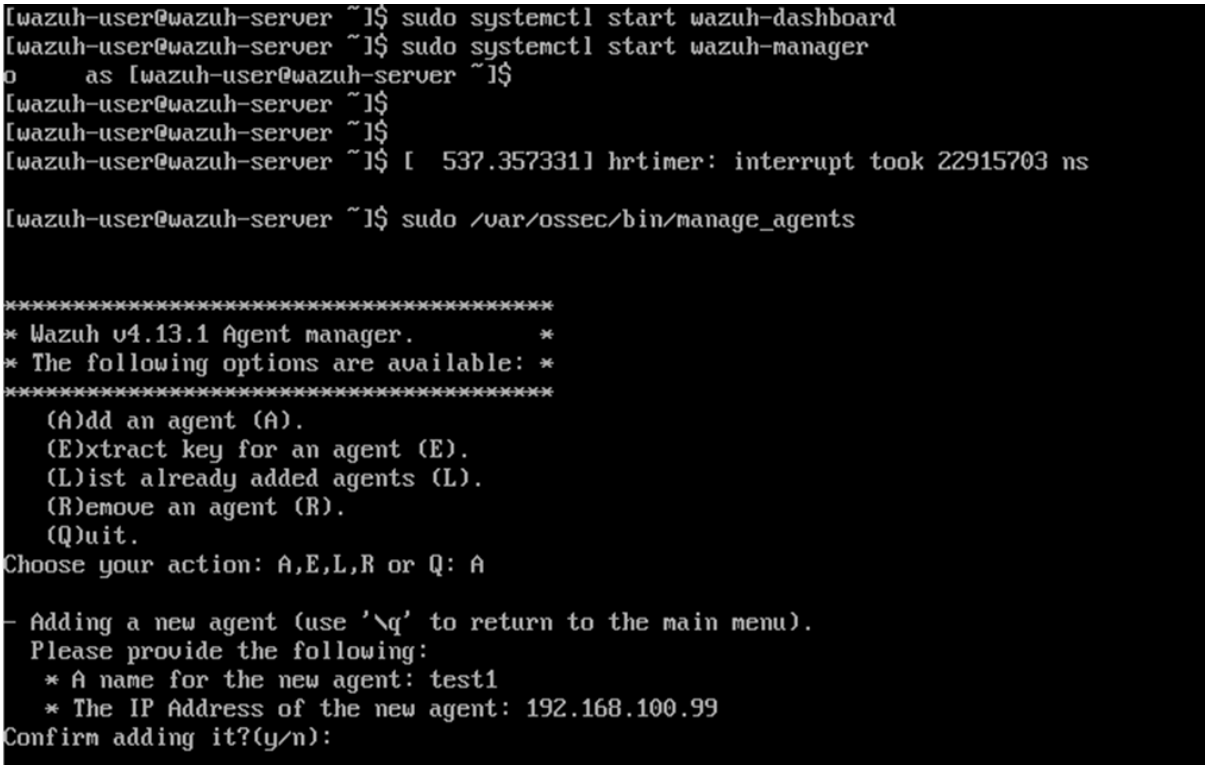
Configuración de agentes simulados:

Una vez iniciado sesión en el servidor, se entra con derechos de administrador a la herramienta para manejar los agentes:

sudo /var/ossec/bin/manage_agents

Se procede a seleccionar la opción A,

E. Listado de agentes



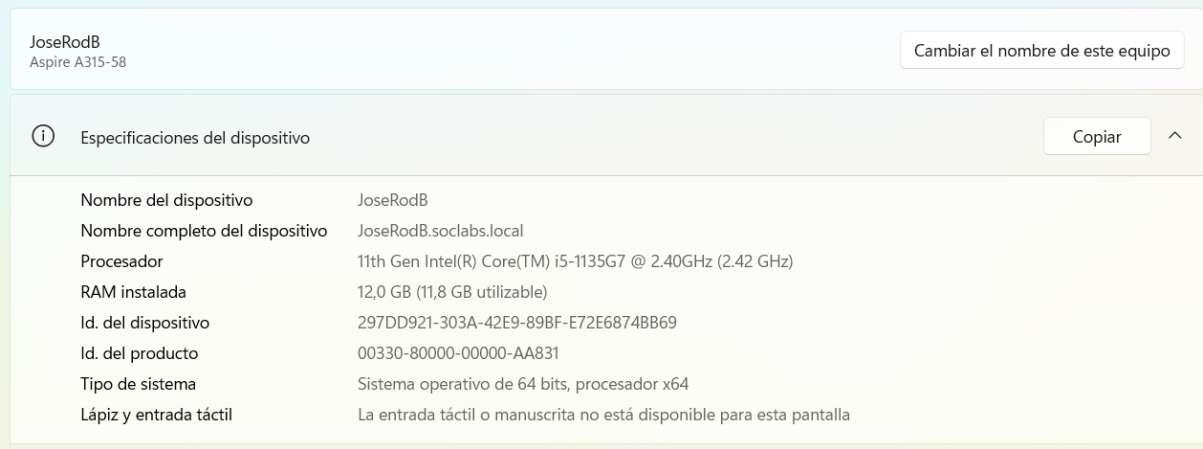
Para luego ingresar un alias, y su respectiva dirección IP con el fin de que pueda ser reconocida por el servidor. Finalmente, se reinicia el servicio wazuh-agent para refrescar los agentes.

sudo systemctl restart wazuh-agent

Con esto, el servidor se encuentra levantado y la máquina cliente enlazada, información que se encontrará reflejada en el dashboard y servidor.

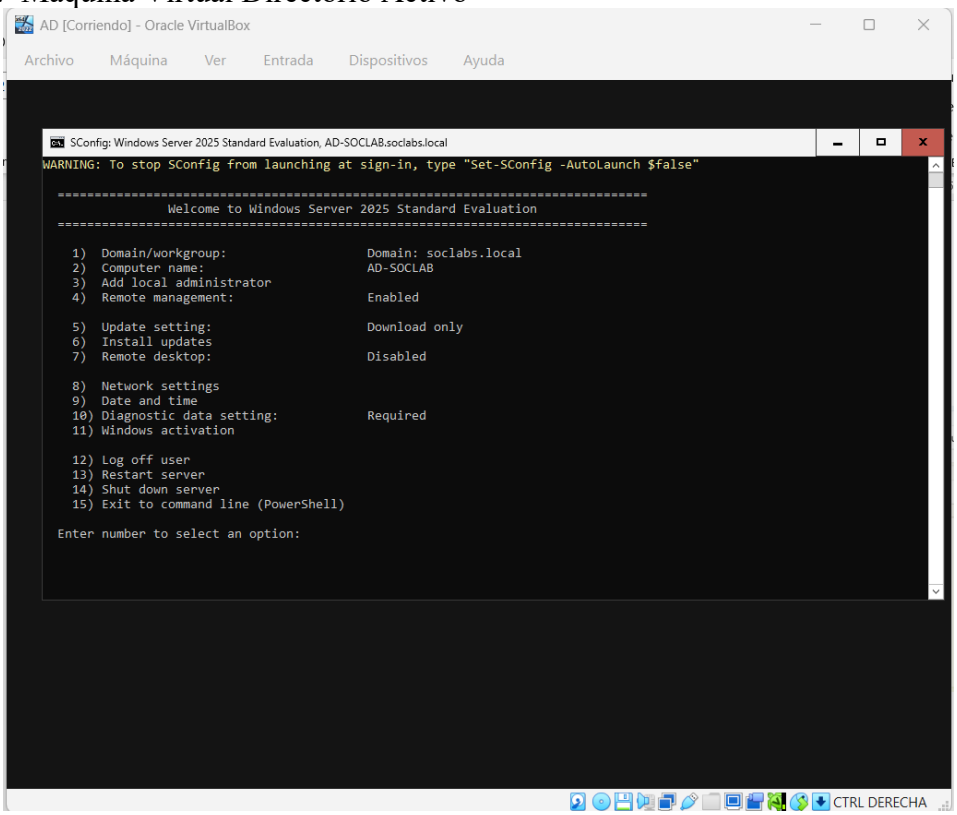
ANEXO B: DIRECTORIO ACTIVO

F. Equipo Hacer Dominio soclabs.local



Por otra parte la máquina virtual dentro de Winserver 2022

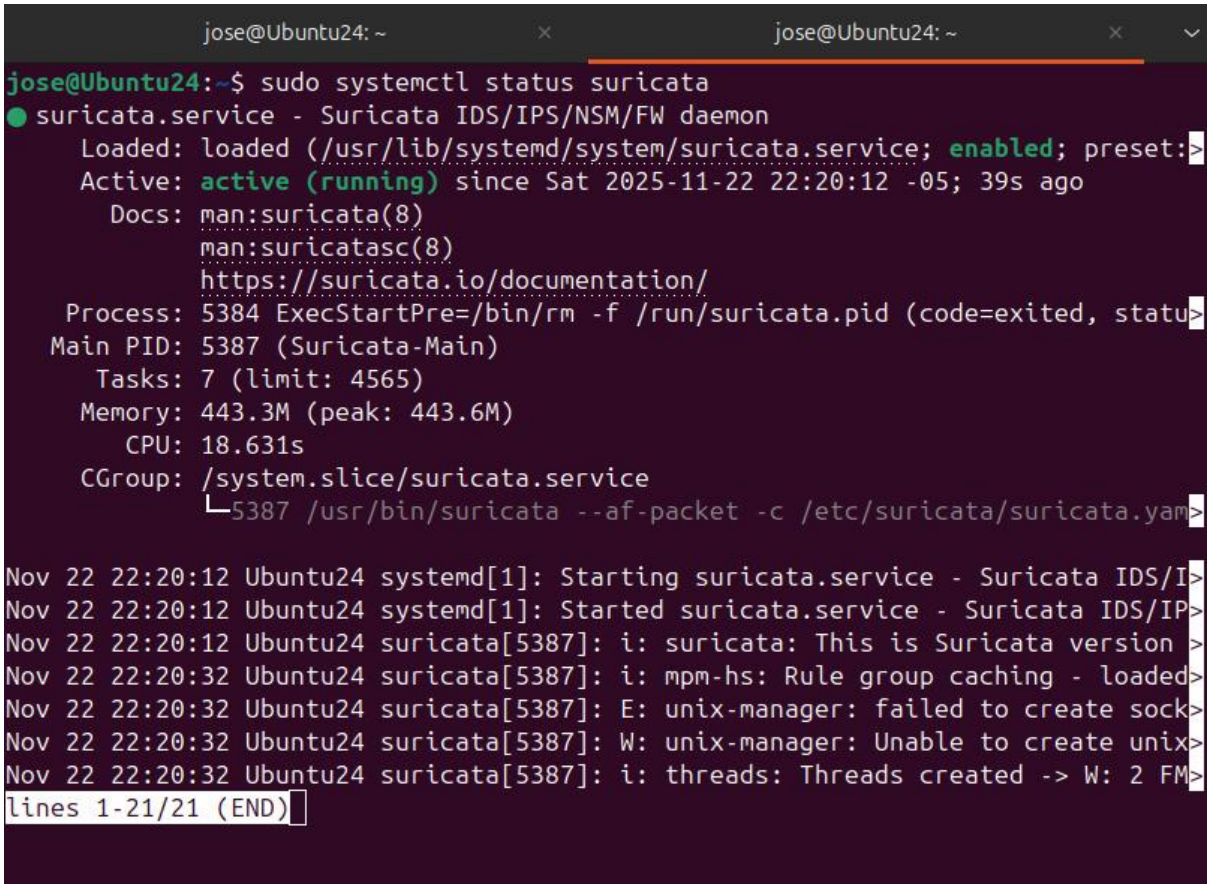
G. Máquina Virtual Directorio Activo



ANEXO C: SURICATA

A continuación se muestra una imagen con la instalación nativa de Suricata sobre Ubuntu 24.04.

H. Estado servicio suricata



Por otra parte, verificando el log de alertas en Suricata:

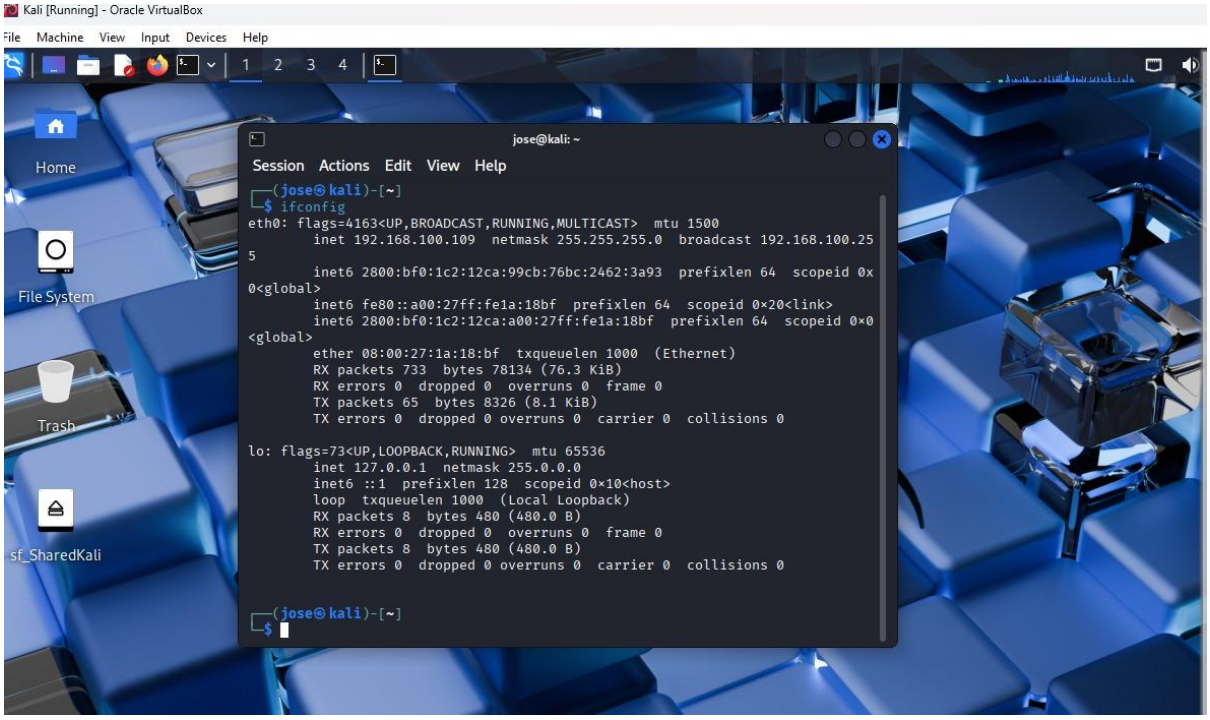
I. Logs de escucha suricata


```
","end":"2025-11-22T22:26:55.421433-0500","age":0,"state":"new","reason":"timeou
t","alerted":false,"exception_policy":[{"target":"app_layer_error","policy":"ign
ore"}]}}
{"timestamp":"2025-11-22T22:27:32.460943-0500","flow_id":2084859673708556,"in_if
ace":"enp3s0","event_type":"flow","src_ip":"fe80:0000:0000:0000:0000:0000:0000:0
001","dest_ip":"2800:0bf0:01c2:12ca:e411:f67e:1f76:71bb","ip_v":6,"proto":"IPv6-
ICMP","icmp_type":135,"icmp_code":0,"flow":{"pkts_toserver":2,"pkts_toclient":0,
"bytes_toserver":172,"bytes_toclient":0,"start":"2025-11-22T22:26:55.419883-0500
","end":"2025-11-22T22:26:55.419985-0500","age":0,"state":"new","reason":"timeou
t","alerted":false,"exception_policy":[{"target":"app_layer_error","policy":"ign
ore"}]}}
{"timestamp":"2025-11-22T22:27:32.460975-0500","flow_id":2093404430928291,"in_if
ace":"enp3s0","event_type":"flow","src_ip":"fe80:0000:0000:0000:0000:0000:0000:0
001","dest_ip":"2800:0bf0:01c2:12ca:3265:60cf:c535:d759","ip_v":6,"proto":"IPv6-
ICMP","icmp_type":135,"icmp_code":0,"flow":{"pkts_toserver":2,"pkts_toclient":0,
"bytes_toserver":172,"bytes_toclient":0,"start":"2025-11-22T22:26:55.421872-0500
","end":"2025-11-22T22:26:55.421900-0500","age":0,"state":"new","reason":"timeou
t","alerted":false,"exception_policy":[{"target":"app_layer_error","policy":"ign
ore"}]}}
{"timestamp":"2025-11-22T22:27:32.461007-0500","flow_id":2083992601366074,"in_if
ace":"enp3s0","event_type":"flow","src_ip":"fe80:0000:0000:0000:0000:0000:0000:0
001","dest_ip":"2800:0bf0:01c2:12ca:d51e:46dc:5805:e597","ip_v":6,"proto":"IPv6-
ICMP","icmp_type":135,"icmp_code":0,"flow":{"pkts_toserver":2,"pkts_toclient":0,
"bytes_toserver":172,"bytes_toclient":0,"start":"2025-11-22T22:26:55.419681-0500
","end":"2025-11-22T22:26:55.419781-0500","age":0,"state":"new","reason":"timeou
t","alerted":false,"exception_policy":[{"target":"app_layer_error","policy":"ign
ore"}]}}
{"timestamp":"2025-11-22T22:27:32.461063-0500","flow_id":2094118238295728,"in_if
ace":"enp3s0","event_type":"flow","src_ip":"fe80:0000:0000:0000:0000:0000:0000:0
001","dest_ip":"2800:0bf0:01c2:12ca:5ffa:eb9c:f06f:bc42","ip_v":6,"proto":"IPv6-
ICMP","icmp_type":135,"icmp_code":0,"flow":{"pkts_toserver":2,"pkts_toclient":0,
"bytes_toserver":172,"bytes_toclient":0,"start":"2025-11-22T22:26:55.422038-0500
","end":"2025-11-22T22:26:55.422067-0500","age":0,"state":"new","reason":"timeou
t","alerted":false}}
{"timestamp":"2025-11-22T22:27:32.842144-0500","in_iface":"enp3s0","event_type":
"alert","pkt_src":"wire/pcap","alert":{"action":"allowed","gid":1,"signature_id"
:2200121,"rev":1,"signature":"SURICATA Ethertype unknown","category":"Generic Pr
```

ANEXO D: KALI LINUX

A continuación se muestra la máquina virtual de Kali levantada en virtualbox.

J.



ANEXO E: INTERFAZ SWITCH CISCO SGB300

A continuación se presenta la interfaz de conexiones en el switch:

K.