



**COLEGIO DE CIENCIAS E INGENIERÍAS**  
**INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN**

Entregable II del Proyecto Integrador

Tutor: Roberto Andrade

Autor: José Contreras

Quito – Ecuador  
2025

## 1. Título del Proyecto:

Orquestador de Agentes de IA para gestión de incidentes de ciberseguridad

## 2. Resumen de actividades realizadas:

## 1. Implementación de prototipo de agente de monitoreo

Para abordar la implementación de este prototipo es importante tomar en cuenta el siguiente flujo de trabajo:

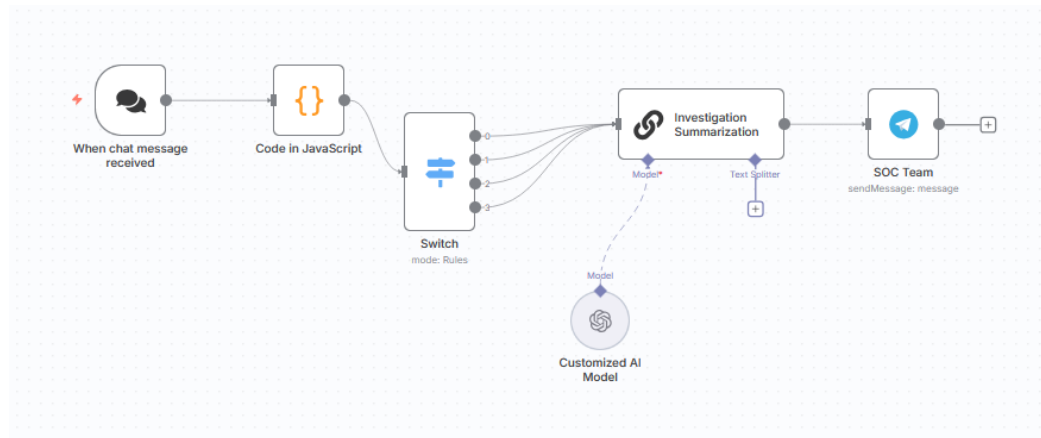


Figura 1. Flujo preliminar

## Descripción 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.

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. Finalmente, el flujo contempla un nodo “**Create Spreadsheet**”, diseñado para registrar de manera opcional los resultados en Google Sheets, permitiendo mantener una trazabilidad histórica de los incidentes procesados.

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:

```
"metrics": {
  "rate_pps": 230000,
  "unique_src_ips": 1200,
  "duration_sec": 1800
},
```

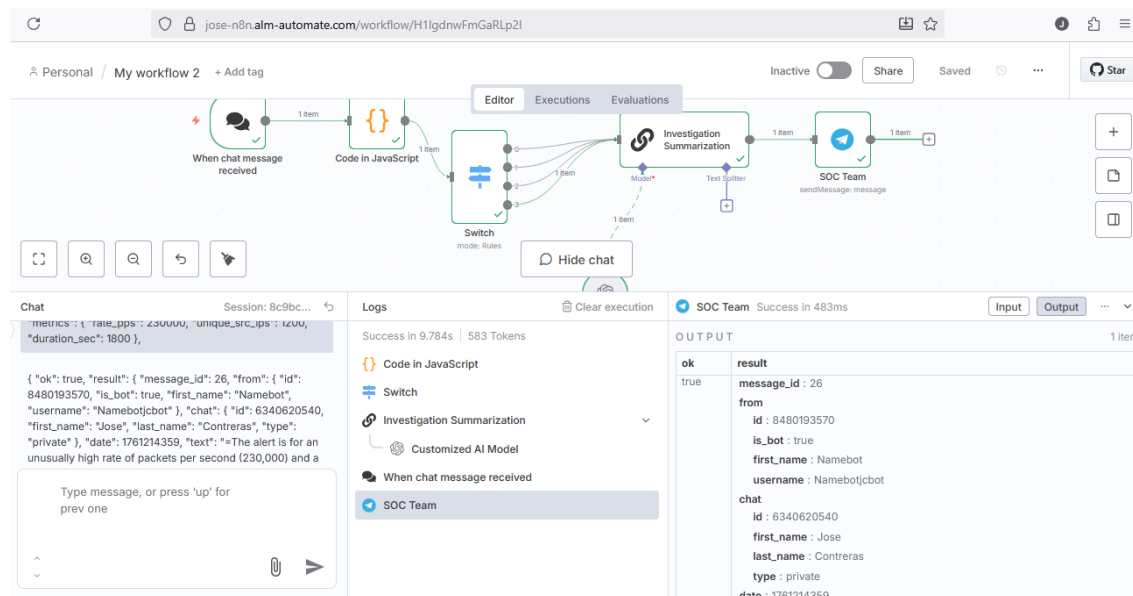


Figura 2. Compilación flujo con mensaje prueba

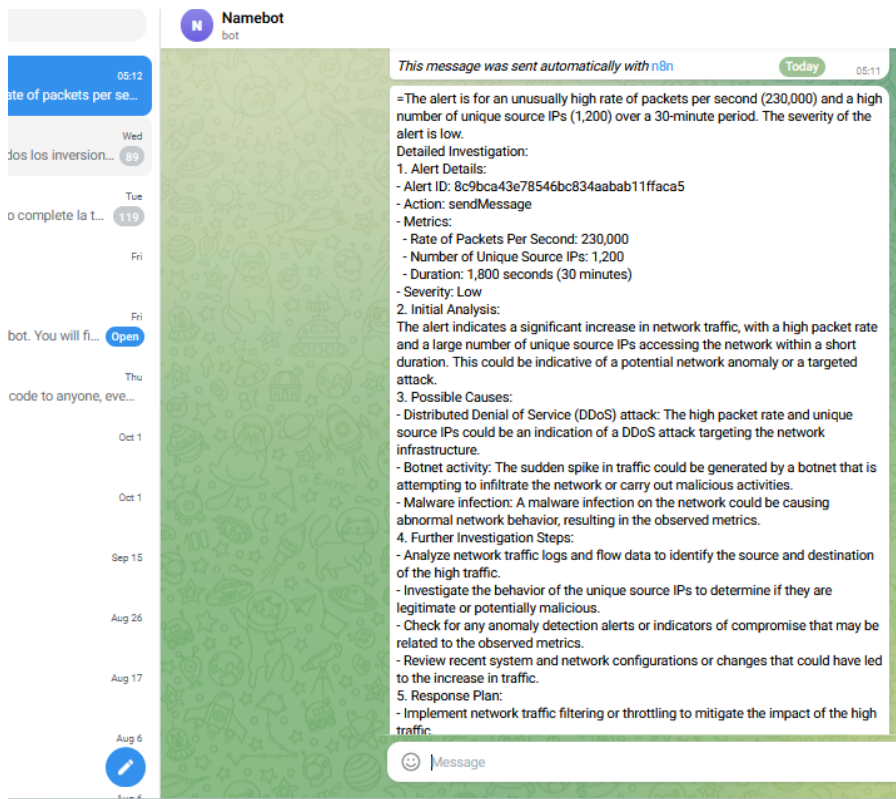


Figura 3. Mensaje de salida Telegram SOC

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

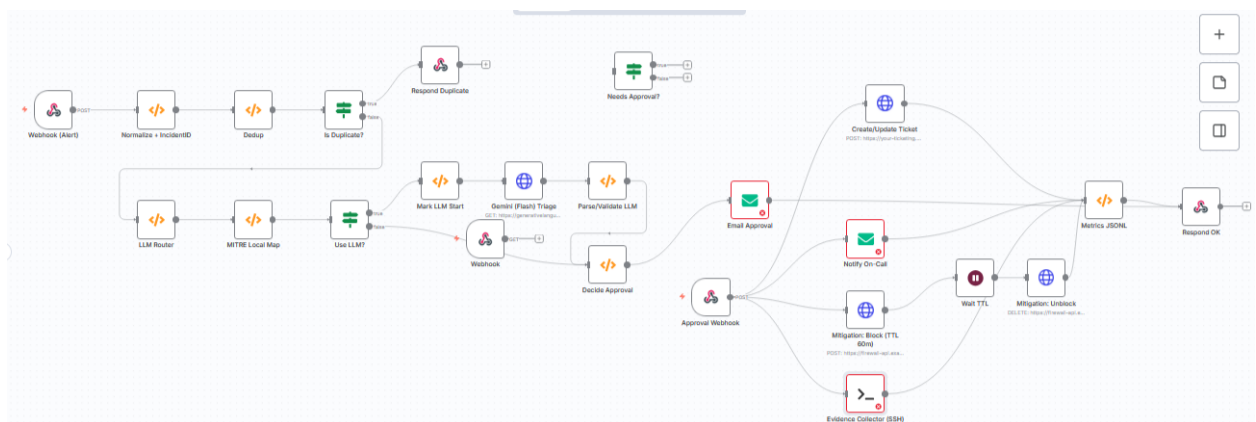


Figura 2. Flujo de monitoreo con Wazuh alert


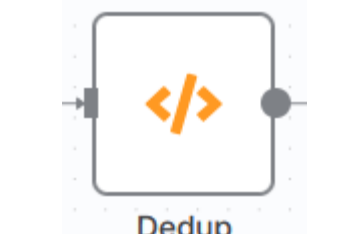
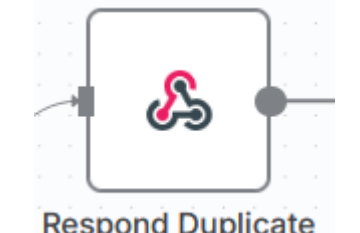

Descripción del flujo por nodo:







El flujo de monitoreo desarrollado tiene como objetivo automatizar la recepción, análisis y priorización de eventos de seguridad provenientes del entorno del SIEM Wazuh. Este flujo se basa en una estructura


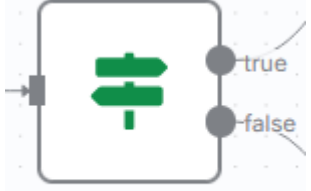

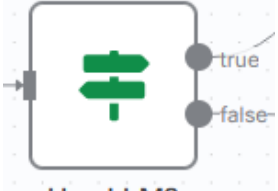


modular de nodos que trabajan de manera secuencial para gestionar de forma eficiente las alertas generadas por el sistema.

El proceso inicia con la recepción de los eventos a través de un webhook que actúa como punto de entrada, seguido de la normalización de los datos y la generación de un identificador único por incidente. Posteriormente, se aplica un mecanismo de deduplicación que evita procesar alertas repetidas y, dependiendo de su complejidad, las envía a análisis mediante un modelo de lenguaje avanzado o a un camino de mitigación directa. Este flujo permite integrar técnicas del marco MITRE ATT&CK, generando reportes de triage y ejecutando acciones automáticas como el bloqueo temporal de direcciones IP o la notificación a equipos de respuesta. En conjunto, este diseño permite una gestión integral del ciclo de monitoreo y respuesta ante incidentes, asegurando trazabilidad, eficiencia y una reducción significativa del tiempo medio de detección y mitigación (MTTD/MTTR).

A continuación, se resume el flujo anterior a nivel de nodo:

Imagen	Nodo	Descripción
 Normalize + IncidentID	<b>Normalize + IncidentID</b>	Este nodo recibe la alerta cruda y la normaliza. Extrae datos clave como dirección IP, aplicación afectada y severidad, y genera un identificador único de incidente ( <i>IncidentID</i> ) calculado a partir de estos campos y una ventana temporal de 5 minutos. Este ID permite asegurar la trazabilidad y evitar duplicidades en los eventos registrados.
 Dedup	<b>Dedup</b>	Implementa un mecanismo de deduplicación temporal basado en memoria estática. Si un <i>IncidentID</i> ya ha sido procesado recientemente (dentro de los últimos cinco minutos), marca la alerta como duplicada y evita su reprocesamiento, optimizando así el rendimiento del flujo.
 Respond Duplicate	<b>Respond Duplicate</b>	Cuando el nodo anterior detecta una alerta repetida, este bloque responde directamente al webhook inicial, deteniendo la ejecución del flujo. De esta manera, se evita el consumo innecesario de recursos computacionales en incidentes ya gestionados.
 MITRE Local Map	<b>MITRE Local Map</b>	Asocia automáticamente la alerta con técnicas y sub-técnicas del marco MITRE ATT&CK. Lo hace a partir de palabras clave detectadas en la descripción o el tráfico (por ejemplo, “HTTP”, “SYN”, “UDP”), enriqueciendo el contexto del incidente con información de tácticas y técnicas reconocidas.

 <p>Parse/Validate LLM</p>	<b>Parse/Validate LLM</b>	Este nodo se encarga de interpretar y validar la respuesta JSON generada por el modelo de lenguaje ( <i>Gemini 1.5 Flash</i> ). Extrae información de triage, plan de acción y mapeo MITRE, y asegura que el resultado cumpla con el esquema esperado antes de continuar con la automatización.
 <p>Decide Approval</p>	<b>Decide Approval</b>	Analiza la información resultante del modelo de lenguaje y determina si el plan de mitigación requiere revisión humana. Si el caso es crítico o ambiguo, marca el proceso como “pendiente de aprobación”, asegurando que las acciones sensibles sean revisadas por un analista.
 <p>Approval Webhook</p>	<b>Approval Webhook</b>	Recibe las aprobaciones humanas mediante una petición HTTP (POST). Este nodo actúa como puerta de enlace entre el analista y el flujo automatizado, permitiendo habilitar o denegar acciones de mitigación en tiempo real.
 <p>Create/Update Ticket POST: https://your-ticketing....</p>	<b>Create/Update Ticket</b>	Crea o actualiza un ticket en el sistema de gestión de incidentes (como Jira, ServiceNow o Zendesk). Registra los detalles del evento, el plan de acción y las referencias MITRE, garantizando la trazabilidad administrativa del proceso.
 <p>Mitigation: Block (TTL 60m) POST: https://firewall-api.ema...</p>	<b>Mitigation: Block (TTL 60m)</b>	Envía una solicitud POST a la API del firewall para bloquear la dirección IP atacante durante un tiempo limitado (TTL de 60 minutos). Este paso representa la ejecución automática de una contramedida técnica frente a ataques de denegación de servicio.
 <p>Respond OK</p>	<b>Respond OK</b>	Es el nodo final del flujo. Envía una respuesta al sistema que generó la alerta, confirmando que el incidente fue procesado correctamente y que las acciones de mitigación o registro fueron completadas sin errores.

 <p>Webhook (Alert)</p>	<b>Webhook (Alert)</b>	<p>Es el punto de entrada del flujo. Recibe las alertas mediante solicitudes HTTP tipo POST provenientes de sistemas externos o del SIEM. Su función es capturar los datos de eventos DoS (Denial of Service) para iniciar el proceso automatizado de análisis y mitigación.</p>
 <p>Is Duplicate?</p>	<b>Is Duplicate?</b>	<p>Evalúa si la alerta actual ya fue procesada recientemente. Si detecta que el incidente es duplicado, redirige la ejecución hacia el nodo <i>Respond Duplicate</i>; si no lo es, continúa hacia el análisis de clasificación.</p>
 <p>LLM Router</p>	<b>LLM Router</b>	<p>Determina si el evento será procesado por la vía rápida (<i>fast path</i>) o si se requiere una intervención más compleja con modelo de lenguaje (<i>use_llm</i>). Para ello, analiza métricas del tráfico y palabras clave, clasificando las alertas obvias o ambiguas.</p>
 <p>Use LLM?</p>	<b>Use LLM?</b>	<p>Nodo condicional que bifurca el flujo dependiendo del valor del parámetro <i>use_llm</i>. Si es verdadero, activa el análisis con el modelo de lenguaje; si es falso, continúa con las acciones predefinidas de mitigación.</p>
 <p>Mark LLM Start</p>	<b>Mark LLM Start</b>	<p>Registra la marca temporal en la que comienza el procesamiento del modelo de lenguaje. Este tiempo se usa posteriormente para calcular métricas de rendimiento del sistema y latencia del análisis.</p>
 <p>Gemini (Flash) Triage GET: <a href="https://generativelangu...">https://generativelangu...</a></p>	<b>Gemini (Flash) Triage</b>	<p>Envía los datos de la alerta al modelo <i>Gemini 1.5 Flash</i> mediante una solicitud HTTP. El modelo analiza el contexto del incidente y genera un informe estructurado con triage, plan de acción, técnicas MITRE y posibles indicadores de falso positivo.</p>

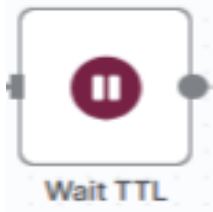

	<b>Wait TTL</b>	Introduce una pausa controlada equivalente al tiempo de vida del bloqueo temporal aplicado (TTL = 60 minutos). Su función es garantizar que la medida de mitigación permanezca activa el tiempo necesario antes de revertirse.
	<b>Metrics JSONL</b>	Registra información operativa del flujo, como el tiempo total de ejecución, uso del modelo LLM, duración de bloqueos, nivel de prioridad y estadísticas de precisión. Esta información alimenta las métricas de rendimiento y auditoría del sistema.

Tabla 1. Descripción Flujo de Monitoreo por nodo

Este flujo demuestra una arquitectura de respuesta automatizada altamente integrada, capaz de combinar reglas lógicas, modelos de lenguaje, comunicación asíncrona y operaciones de red en un solo ciclo operativo. La inclusión del modelo *Gemini* permite mejorar la interpretación contextual de las alertas, mientras que los mecanismos de deduplicación, aprobación y reversión garantizan seguridad y control operacional. En conjunto, este flujo consolida las capacidades de orquestación y respuesta del sistema, alineándose con los principios de un *Security Orchestration, Automation and Response (SOAR)* moderno.

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 procedió a la construcción de un nuevo flujo de trabajo a partir de la estructura preliminar.

## 2. Levantamiento de Laboratorio:

El levantamiento del laboratorio de ciberseguridad se llevó a cabo con el propósito de construir un entorno controlado para la simulación de ataques y la validación del funcionamiento del orquestador. Para ello, se implementó un servidor Wazuh Manager en una máquina virtual preconfigurada con sistema Wazuh v4.13.1 OVA. 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>

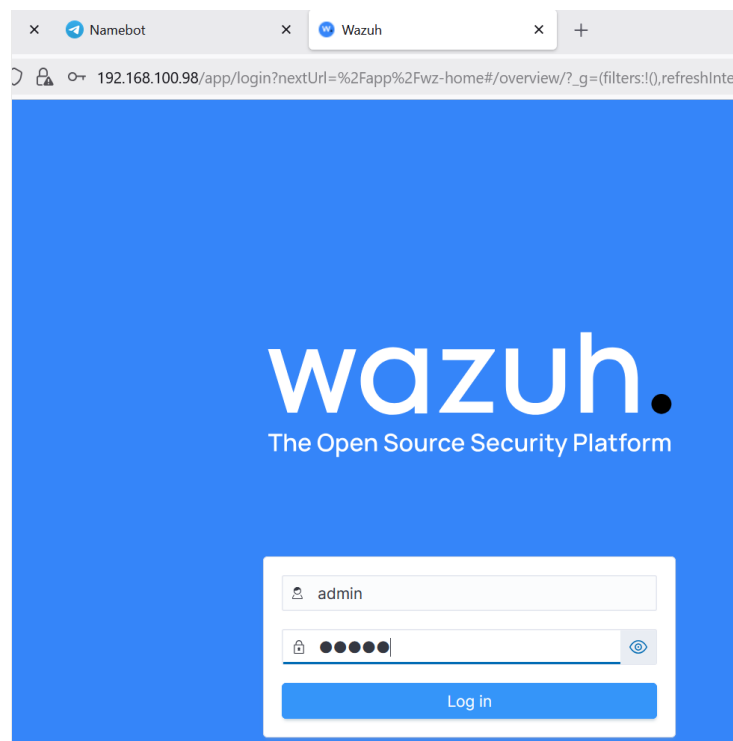


Figura 3. Acceso a Dashboard

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

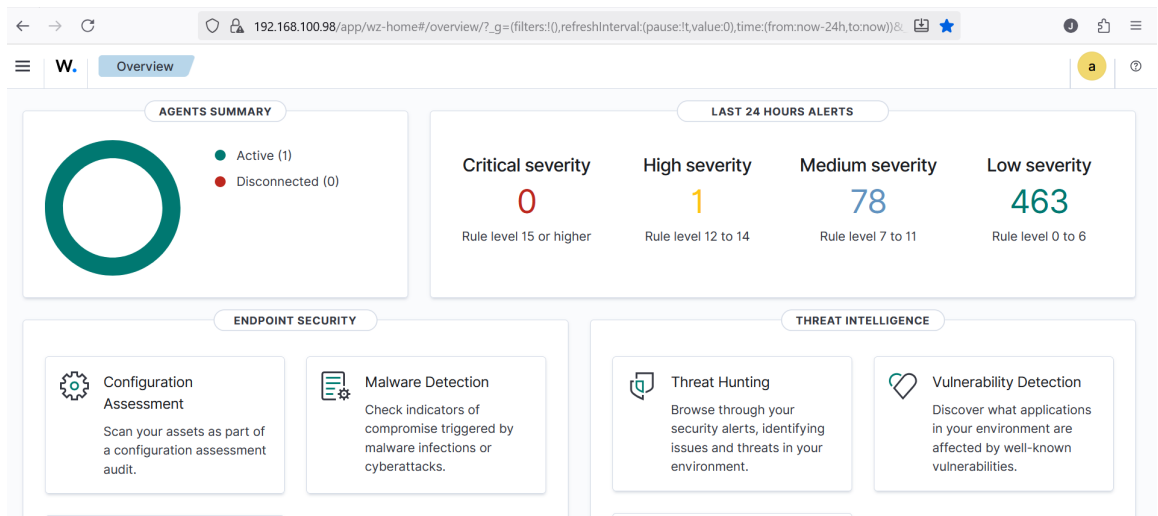


Figura 4. Dashboard Wazuh

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:

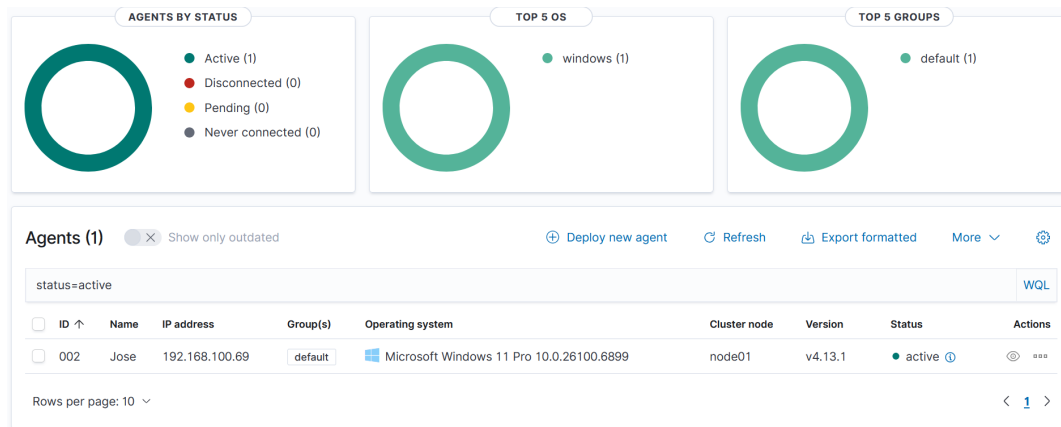


Figura 5. Detalle de conexión Endpoint

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

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::a3d:735a:3615:df78%2
    IPv4 Address. . . . . : 192.168.56.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Wireless LAN adapter Local Area Connection* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    IPv6 Address. . . . . : 2800:bf0:1c2:12ca:1919:5017:e8dc:6e04
    Temporary IPv6 Address. . . . . : 2800:bf0:1c2:12ca:fd9d:c8a0:d649:b862
    Link-local IPv6 Address . . . . . : fe80::7200:5115:c21b:11b5%16
    IPv4 Address. . . . . : 192.168.100.69
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::1%16

```

Figura 6. IP de máquina cliente

#### 4. 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,

```
[wazuh-user@wazuh-server ~]$ sudo systemctl start wazuh-dashboard
[wazuh-user@wazuh-server ~]$ sudo systemctl start wazuh-manager
o as [wazuh-user@wazuh-server ~]$
[wazuh-user@wazuh-server ~]$
[wazuh-user@wazuh-server ~]$
[wazuh-user@wazuh-server ~]$ [ 537.357331] hrtimer: interrupt took 22915703 ns

[wazuh-user@wazuh-server ~]$ sudo /var/ossec/bin/manage_agents

*****
* Wazuh v4.13.1 Agent manager.          *
* The following options are available: *
*****
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
Choose your action: A,E,L,R or Q: A

- Adding a new agent (use '\q' to return to the main menu).
Please provide the following:
  * A name for the new agent: test1
  * The IP Address of the new agent: 192.168.100.99
Confirm adding it?(y/n):
```

Figura 7. 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. A continuación, se muestra un diagrama con la topología a implementarse de forma completa:

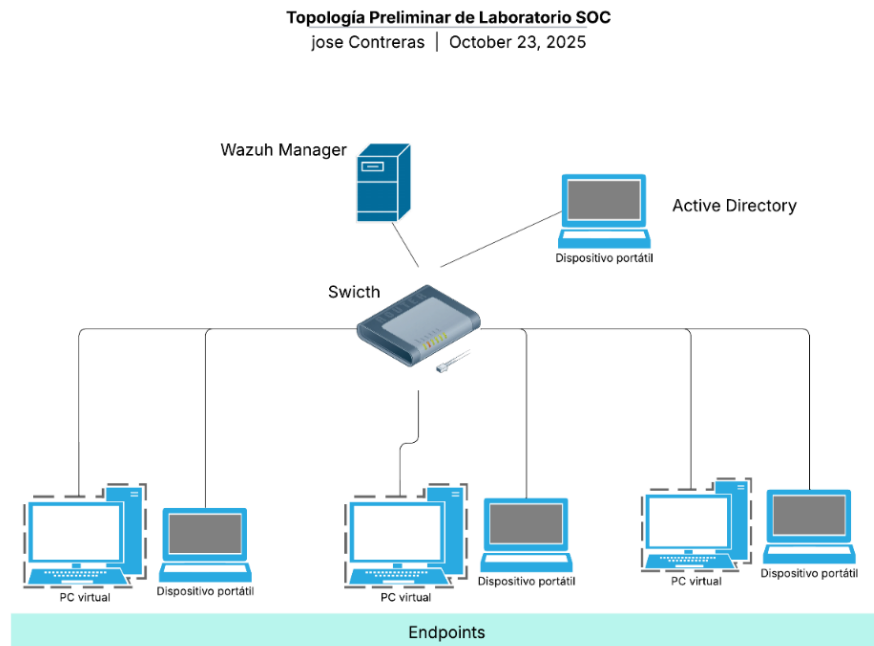


Figura 8. Topología de Laboratorio SOC

#### Descripción de diagrama

La topología representada en la figura muestra la arquitectura preliminar del laboratorio SOC implementado para la validación del proyecto. En ella se puede observar un servidor Wazuh Manager que actúa como núcleo del sistema de monitoreo y gestión de eventos de seguridad (SIEM), encargado de centralizar la recopilación de logs, correlacionar alertas y coordinar la comunicación con los agentes distribuidos en los diferentes equipos finales. Conectado a este se encuentra un servidor de Active Directory, utilizado para administrar usuarios, políticas de autenticación y control de acceso en el entorno experimental. Ambos servidores están interconectados a través de un switch de red, que constituye el punto de enlace principal entre los distintos componentes del laboratorio.

En el nivel inferior se ubican los endpoints, conformados por una combinación de PCs virtuales y dispositivos portátiles donde se instalan los agentes Wazuh, los cuales envían eventos de seguridad al servidor principal. Esta estructura permite recrear escenarios de ataque controlado, como intentos de autenticación fallidos, accesos no autorizados o patrones de tráfico asociados a ataques DDoS, proporcionando un entorno funcional para el análisis de alertas y la evaluación de las capacidades de detección y respuesta del sistema. En conjunto, esta topología reproduce las condiciones de un Centro de Operaciones de Seguridad (SOC), integrando de manera práctica los procesos de monitoreo, análisis y respuesta automatizada en un ambiente controlado de laboratorio.

3. Secciones o capítulos del documento final desarrollados

1. Desarrollo del Prototipo

4. Revisión y firma del tutor del proyecto

Yo, Roberto Andrade, profesor de la carrera de Ingeniería en Ciencias de la Computación, hago constar que he revisado y, por lo tanto, apruebo las actividades realizadas durante este período de trabajo. Por otra parte, considero que el avance del proyecto integrador es adecuado y se corresponde con el cronograma definido en el documento de planificación.

---

Fdo: Roberto Andrade

Quito, 23 de Octubre de 2025