Implementación en contenedores de la plataforma de código abierto Wazuh

Protección SIEM y XDR

Primitivo Calvo roo

Ingeniería Informática
Seguridad informática

Nombre Tutor/a de TF
Jorge Miguel Moneo
Profesor/a responsable de la asignatura
Andreu Pere Isern Deyà

Fecha Entrega
01/2024

Universitat Oberta de Catalunya

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Implementación en contenedores de la plataforma de código abierto Wazuh*
Nombre del autor:	*Primitivo Calvo Roo*
Nombre del consultor/a:	*Jorge Miguel Moneo*
Nombre del PRA:	*Andreu Pere Isern Deyà*
Fecha de entrega (mm/aaaa):	*01/2024*
Titulación o programa:	Ingeniería Informática
Área del Trabajo Final:	*Seguridad informática*
Idioma del trabajo:	*Castellano*
Palabras clave	*SIEM, XDR, Docker*

Resumen del Trabajo

La mayor complejidad y sofisticación de los ciberataques hace necesario que las organizaciones dispongan de los medios organizativos, técnicos y de personal para hacerle frente. Sin embargo, las pequeñas y medianas empresas sin los recursos necesarios son un objetivo preferente de los ciberdelincuentes, con el riesgo que ello implica.

En este trabajo se despliega la plataforma de seguridad Wazuh (SIEM con XDR) para evaluar el estado de seguridad y proteger proactivamente los dispositivos finales, dado que aparecen relacionados en una parte importante de los vectores de ataque.

Primero, se realiza una investigación de soluciones SIEM de código abierto o comercial de uso gratuito, para concluir con una comparativa con un enfoque cuantitativo y cualitativo que permita contrastar la plataforma Wazuh con otras soluciones y verificar si cumple los requisitos necesarios para su implementación.

A continuación, el análisis de documentación sirve de base para adaptar el despliegue de la plataforma en contenedores con los principios básicos de bastionado de los sistemas y con los mínimos componentes indispensables.

Finalmente, se analizan y evalúan cualitativamente posibles casos de uso de la plataforma Wazuh que ayuden a proteger los dispositivos finales, teniendo en cuenta tácticas, técnicas y procedimientos utilizadas por los atacantes.

Las conclusiones del trabajo pretenden determinar que la solución implementada proporciona medios de protección eficaces a distintos ciberataques que se observan en la actualidad.

Abstract

The greater complexity and sophistication of cyber-attacks makes it necessary for organisations to have the organisational, technical and staff resources to deal with them. However, small and medium-sized enterprises without the necessary resources are a prime target for cybercriminals, with the risk that this entails.

In this work, the Wazuh security platform (SIEM with XDR) is deployed to assess the security status and proactively protect endpoints, due to the fact that they are related in an important part of the attack vectors.

First, an investigation of open source or free-to-use commercial SIEM solutions is carried out, to conclude with a comparison with a quantitative and qualitative approach that allows contrasting the Wazuh platform with other solutions and verifying whether it meets the necessary requirements for its implementation.

Next, the documentation analysis is used as a starting point for adapting the deployment of the containerised platform with the bastioning basic principles of the systems with the minimum necessary components.

Finally, possible use cases of the Wazuh platform to help protect endpoints are analysed and qualitatively evaluated, taking into consideration tactics, techniques and procedures used by attackers.

The conclusions of the work aim to determine that the implemented solution provides protection effective measures against different cyberattacks that are currently observed.

Índice

Lista de figuras

Lista de tablas

Lista de ficheros

1. Introducción

1.1. Contexto y Justificación del Trabajo

Actualmente se observa una mayor sofisticación en los incidentes de seguridad y la existencia de amenazas en la modalidad de servicio, como son Malware-as-a-Service, Hacker-as-a-Service o Access-as-a-Service, tal y como se indica en diferentes informes [1–3].

Además, Deloitte destaca, en su informe de tendencias de ciberamenazas de marzo de 2023 [1], que los cibercriminales ya están empezando a utilizar la inteligencia artificial (IA), concretamente ChatGPT, para actividades tan variadas como ataques de fuerza bruta, el escaneo de vulnerabilidades o la creación de exploits. Este hecho simplifica la realización de ciberataques sofisticados por parte de actores maliciosos, con el consiguiente incremento de riesgo que supone.

Otro dato preocupante que se extrae del balance de ciberseguridad de 2022 del INCIBE [4] es que el 87% de los incidentes de seguridad de las empresas están relacionados con vulnerabilidades en sus productos tecnológicos.

Todo el contexto indicado hace necesaria la concienciación en materia de ciberseguridad en las organizaciones, para que se apliquen buenas prácticas y se utilicen productos de seguridad que permitan una detección y una respuesta proactiva frente a distintas variedades de malware e intentos de intrusión.

La situación se complica aún más cuando una empresa no dispone de los recursos económicos, técnicos y organizativos suficientes para adquirir soluciones TIC que les proporcionen un grado de protección adecuado.

La finalidad de este Trabajo es estudiar y realizar una implementación de la plataforma de seguridad de código abierto Wazuh (SIEM con XDR) [5], que permita recopilar de manera centralizada distintos eventos de seguridad de dispositivos finales o endpoints para proporcionar una detección y una respuesta ante distintas amenazas. Se busca suministrar una opción válida en las organizaciones que no suponga el pago de licencias por su utilización, especialmente para aquellas en las que la única alternativa es optar por una solución de uso libre, ya sea de código abierto o propietario.

Empresas de consultoría de gran reputación como Gartner y Forrester centran sus análisis del mercado de las TIC en productos comerciales, pero eso no quiere decir que no existan otras alternativas viables. El modelo de negocio de Wazuh se basa únicamente en los ingresos por sus servicios profesionales y por su producto SaaS en la nube, con clientes de renombre como Salesforce, Walgreens, Verifone, la NASA y PWC sin suponer ningún coste por las licencias de uso [6].

La implementación que se propone se efectuará mediante la tecnología de contenedores en un único servidor/equipo físico para empresas con unos 100 dispositivos finales, que deberá desplegarse de manera sencilla y permitir su escalado horizontal o en la nube en caso de ser necesario.

Las conclusiones de este Trabajo permitirán determinar el estado actual de la solución y verificar que se cumplen los requisitos necesarios para monitorizar y proteger los

activos finales de la red de las organizaciones, especialmente en el caso de que tengan carencias de recursos TIC.

1.2. Objetivos del Trabajo

El objetivo general (OG) de este Trabajo es analizar, implementar en contenedores y evaluar la plataforma de seguridad Wazuh (SIEM con XDR integrado). Se trata de determinar si esta solución de código abierto proporciona funcionalidades que ayuden a proteger de manera efectiva los dispositivos finales frente a diferentes ataques empleados hoy en día.

Los objetivos específicos que ayudan al cumplir el objetivo general son los siguientes:

- OE.1. Análisis de soluciones actuales de tipo SIEM de código abierto y comerciales de uso gratuito, que ofrezca una visión global de las alternativas disponibles con sus funcionalidades más destacables.
- OE.2. Estudio de la plataforma de seguridad Wazuh para comprender su arquitectura y cómo se interrelacionan sus componentes entre sí. Este conocimiento facilita la detección de posibles puntos de fallo y ayuda a comprender diseños similares de otros productos o las opciones de escalabilidad de la plataforma en función del software de cada componente.
- OE.3. Despliegue de servicios complejos mediante la tecnología de contenedores, especialmente cuando se trata de aplicaciones multicontenedor. Se busca configurar y adaptar el producto Wazuh para que cada uno de sus componentes principales se despliegue en su propio contenedor y con sus recursos necesarios, para así aprovechar sus beneficios: portabilidad, aislamiento e independencia del sistema.
- OE.4. Protección de las comunicaciones de los servicios mediante herramientas de código abierto, implementando los componentes necesarios que permitan que el tráfico intercambiado entre los servicios principales se realice de manera cifrada.
- OE.5. Evaluación y configuración de módulos que ayuden en la detección de amenazas en los endpoints y su mitigación, teniendo en cuenta tanto las funcionalidades de Wazuh como las tácticas y técnicas utilizadas por los ciberatacantes.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

Este Trabajo pretende alinearse con distintos Objetivos de Desarrollo Sostenible (ODS) que la ONU aprobó en su Agenda 2030 [7], de tal forma que se fomente la sostenibilidad, el comportamiento ético, la responsabilidad social, la sostenibilidad y el respecto, tanto de los derechos humanos como de la diversidad.

El estudio de la implementación de una herramienta de seguridad de código abierto Wazuh (SIEM con XDR) pretende ofrecer una solución al alcance de cualquiera, independientemente de su género y de su situación social, económica o política. Se encuadra, por tanto, con los objetivos ODS 5 (Igualdad de género) y ODS 10 (Reducción de las desigualdades).

La disponibilidad de este tipo de herramientas de código abierto favorece el crecimiento económico y nuevos modelos de negocio. Por una parte, las organizaciones (públicas y privadas) tendrán a su alcance una solución que protege los activos tecnológicos con los que desempeña su actividad y, por otra, podrán contratar el servicio de personal cualificado en el caso de que no dispongan de los recursos necesarios. Se considera, por tanto, que hay un alineamiento completo con ODS 8 (Trabajo decente y crecimiento económico) y parcial con ODS 16 (Paz, justicia e instituciones sólidas). Una institución no podrá ser sólida si no dispone de todos sus activos disponibles para desempeñar su misión y responsabilidad social.

La solución que se busca fomenta la sostenibilidad gracias a su adaptabilidad y portabilidad a distintos componentes tecnológicos, cumpliéndose los objetivos ODS 9 (Industria, innovación e infraestructura), ODS 11 (Ciudades y comunidades sostenibles) y ODS 12 (Producción y consumo responsables). El uso de herramientas innovadoras hace posible alcanzar un consumo responsable y la reutilización de los dispositivos informáticos hasta el final de su vida útil.

1.4. Enfoque y método seguido

Los enfoques y métodos seguidos en este Trabajo junto a las herramientas seleccionadas se pueden agrupar en los siguientes grupos de tareas principales.

1.4.1. Gestión y planificación del Trabajo

La gestión del Trabajo y la planificación de sus tareas se basa en la metodología Agile, de tal forma que se permita un refinamiento continuo de los productos que se deben de entregar al final. Se pretende ir proporcionando información más precisa y conveniente en cada iteración a medida que se va ampliando el conocimiento sobre la temática del Trabajo.

Complementariamente, se ha optado por un método visual que permita un seguimiento sencillo de las tareas proyectadas, que consiste en el funcionamiento sincronizado del cronograma (diagrama de Gantt) con flujos de trabajo Kanban, es decir, un tablero con las listas *Listas de tareas*, *En proceso* y *Hecho* en el que las tareas, representadas por tarjetas, van evolucionando hasta que queden completadas en la lista correspondiente.

Las opciones gratuitas de las herramientas web TeamGant [8], planificación temporal con las distintas tareas e hitos en un diagrama de Gantt, y Trello[9], gestión de tareas mediante flujos de trabajo Kanban, permiten su sincronización mediante la opción Power-Ups de esta última.

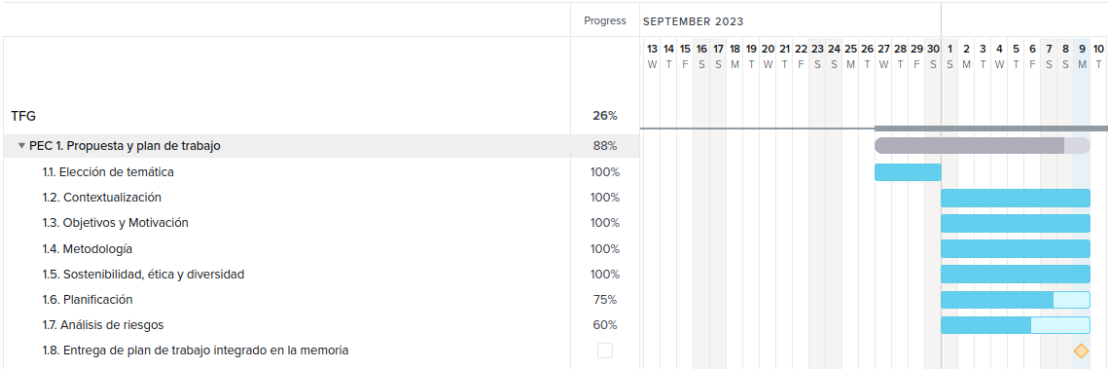


Figura 1: Ejemplo de progreso de las tareas

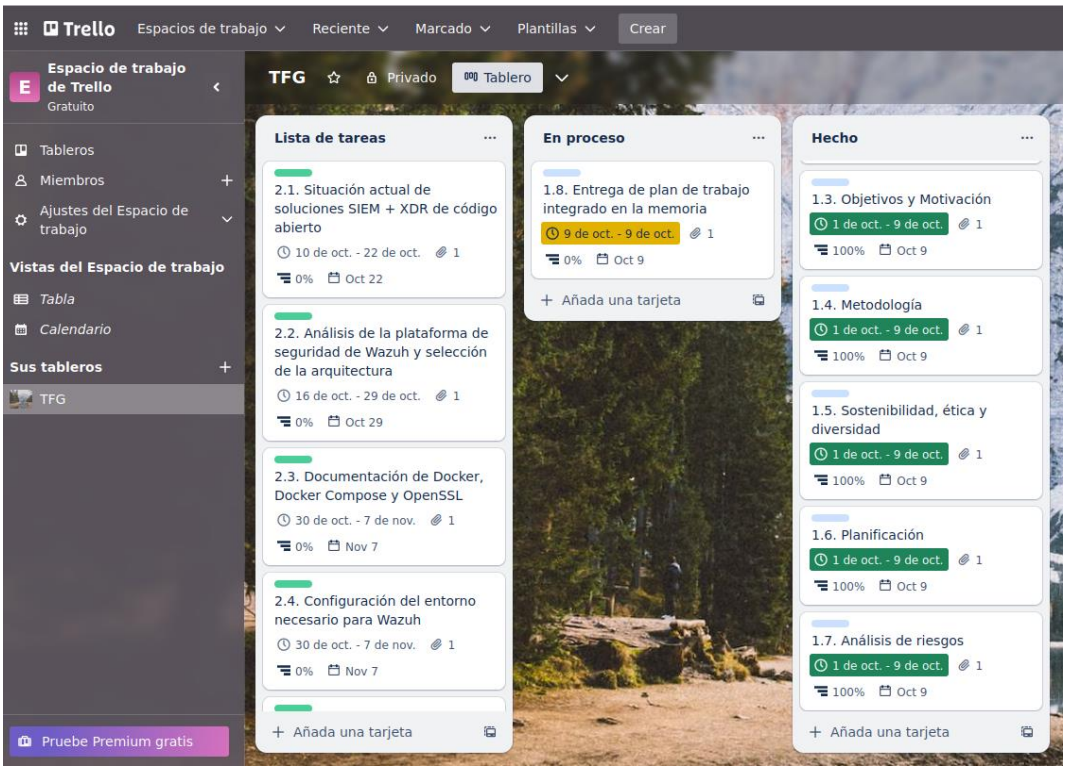


Figura 2: Tablero de Trello

1.4.2. Estado del arte y análisis de soluciones SIEM de uso gratuito

Este apartado consiste en la revisión bibliográfica y de fuentes de Internet contrastadas, para contextualizar el trabajo y conocer distintas soluciones SIEM de software libre o comercial de uso gratuito que dispongan de XDR para proteger a los dispositivos finales. Se pretende realizar un trabajo de investigación para concluir con una comparativa con un enfoque cuantitativo y cualitativo de funcionalidades relevantes para el caso, de tal forma que se pueda conocer el estado del producto Wazuh con respecto sus posibles alternativas en la actualidad.

1.4.3. Implantación de la plataforma de seguridad Wazuh

La implantación de la plataforma de seguridad Wazuh se fundamenta en el análisis de documentación y la adaptación de la solución para que cada uno de sus componentes principales funcionen dentro de su propio contenedor, teniendo en cuenta buenas prácticas y recomendaciones de organismos acreditados en lo relacionado con la seguridad informática. Esto facilitará la portabilidad y permitirá que la arquitectura pueda dimensionarse adecuadamente en el caso de que los requisitos cambien.

1.4.4. Casos de uso para proteger los dispositivos finales

La selección de casos de uso que protejan los dispositivos finales supone el análisis de las Tácticas, Técnicas y Procedimientos (TTPs) del MITRE [10] utilizadas por los atacantes según informes recientes [2,11], para, finalmente, activar las medidas de protección correspondientes en la plataforma de seguridad Wazuh y evaluar su efectividad cualitativamente.

1.4.5. Gestión bibliográfica

Este Trabajo tiene un alto componente de revisión de documentación e investigación de soluciones, por lo que se escoge la aplicación Zotero [12] para recopilar las fuentes de información relevantes y para gestionar adecuadamente la bibliográfica que se utilizará.

Siguiendo las recomendaciones de la UOC, se utilizará una modificación del estilo de citación Vancouver proporcionado por Zotero para que se muestre el nombre completo de los autores o autoras y, de este modo, eliminar estereotipos y dar visibilidad a la aportación que realizan las mujeres. Además, las referencias se realizarán mediante números entre corchetes y la bibliografía se ordenará por orden de aparición.

1.5. Planificación del Trabajo

Los recursos necesarios para el Trabajo es un ordenador personal que permita ejecutar al menos dos máquinas virtuales simultáneamente: una para la plataforma de seguridad Wazuh con al menos 4 vCPU y 8 GB de RAM y otra para simular a los dispositivos finales (*endpoints*) con al menos 2 vCPU y 3 GB de RAM (SO GNU/Linux y SO Microsoft Windows).

Se priorizará la utilización de *software* libre para tareas relacionadas con la implementación de la plataforma Wazuh.

Las tareas, hitos y planificación temporal aparecen reflejados en la tabla y en el diagrama de Gantt que aparecen a continuación.

Tareas e hitos	Inicio	Fin
PEC 1. Propuesta y plan de trabajo	**27/09/2023**	**09/10/2023**
1.1. Elección de temática	27/09/2023	30/09/2023
1.2. Contextualización	01/10/2023	09/10/2023
1.3. Objetivos y Motivación	01/10/2023	09/10/2023
1.4. Metodología	01/10/2023	09/10/2023
1.5. Sostenibilidad, ética y diversidad	01/10/2023	09/10/2023
1.6. Planificación	01/10/2023	09/10/2023
1.7. Análisis de riesgos	01/10/2023	09/10/2023
1.8. Entrega de plan de trabajo integrado en la memoria (hito)	*09/10/2023*	*09/10/2023*
PEC 2. Documentación previa y configuración del entorno	**10/10/2023**	**07/11/2023**
2.1. Situación actual de soluciones SIEM + XDR de código abierto	10/10/2023	22/10/2023
2.2. Análisis de la plataforma de seguridad de Wazuh y selección de la arquitectura	16/10/2023	29/10/2023
2.3. Documentación de Docker, Docker Compose y OpenSSL	30/10/2023	07/11/2023
2.4. Configuración del entorno necesario para Wazuh	30/10/2023	07/11/2023
2.5. Entrega parcial de la memoria del TFG (hito)	*07/11/2023*	*07/11/2023*
PEC 3. Puesta en marcha de la plataforma de seguridad Wazuh	**08/11/2023**	**03/12/2023**
3.1. Generación de certificados necesarios para la plataforma mediante un script	08/11/2023	12/11/2023
3.2. Instalación y configuración de Wazuh Indexer, Wazuh Manager y Wazuh Dashboard en contenedores	13/11/2023	24/11/2023
3.3. Instalación y configuración de los agentes Wazuh (Linux + Windows)	25/11/2023	03/12/2023
3.4. Entrega parcial de la memoria del TFG (hito)	*03/12/2023*	*03/12/2023*
3.5. Entrega de archivos utilizados en la implementación (hito)	*03/12/2023*	*03/12/2023*
PEC 4. Casos de uso y entrega de la memoria del TFG	**04/12/2023**	**09/01/2024**
4.1. Elección de casos de uso que ayuden a proteger los dispositivos finales	04/12/2023	09/12/2023

Tareas e hitos		Inicio	Fin
	4.2. Configuración y documentación de los módulos necesarios para los casos de uso	10/12/2023	31/12/2023
	4.3. Redacción final de la memoria del TFG: Conclusiones y trabajos futuros + Revisión final	23/12/2023	09/01/2024
	4.4. Entrega de la versión definitiva de la memoria del TFG (hito)	*09/01/2024*	*09/01/2024*
	4.5. Entrega de archivos utilizados en la implementación (hito)	*09/01/2024*	*09/01/2024*
PEC 5. Presentación en vídeo		**06/01/2024**	**15/01/2024**
	5.1. Preparación de la presentación de diapositivas	06/01/2024	13/01/2024
	5.2. Preparación de la demostración de la implementación	06/01/2024	13/01/2024
	5.3. Realización del vídeo-presentación	14/01/2024	15/01/2024
	5.4. Entrega del vídeo-presentación	*15/01/2024*	*15/01/2024*

Tabla 1: Tareas e hitos

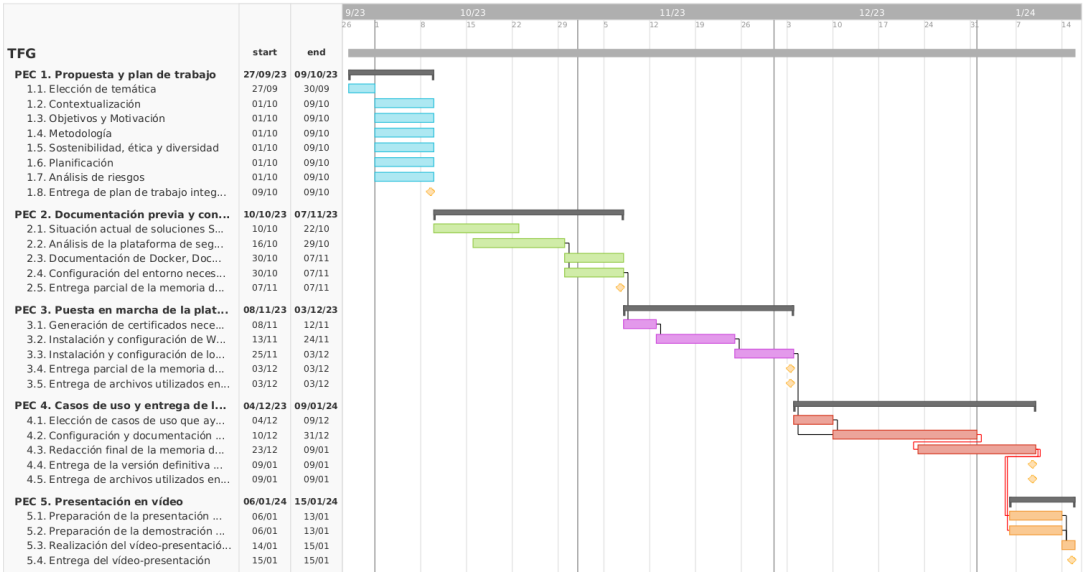


Figura 3: Diagrama de Gantt completo

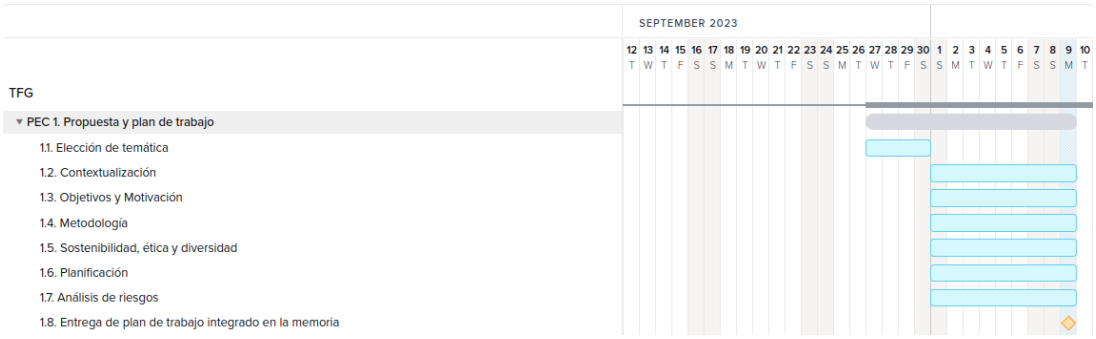


Figura 4: Diagrama de Gantt de la fase PEC 1

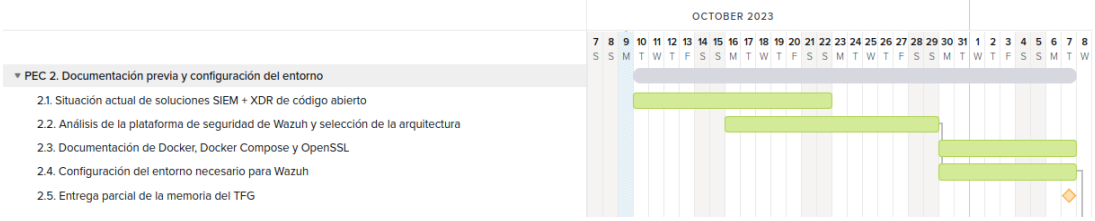


Figura 5: Diagrama de Gantt de la fase PEC 2

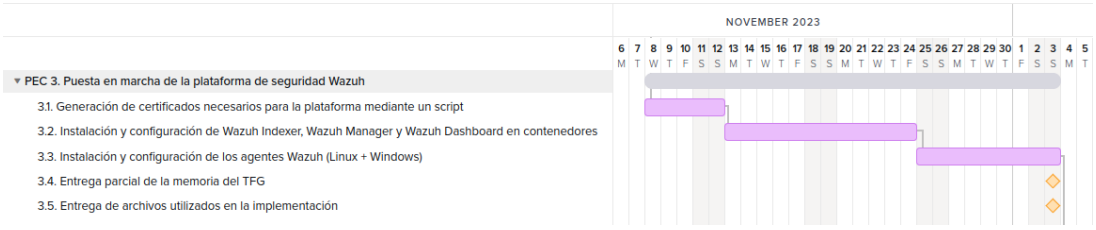


Figura 6: Diagrama de Gantt de la fase PEC 3

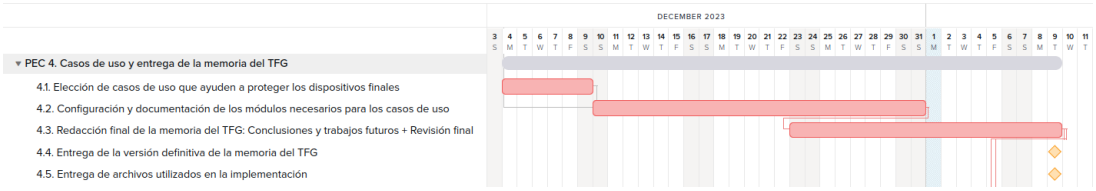


Figura 7: Diagrama de Gantt de la fase PEC 4

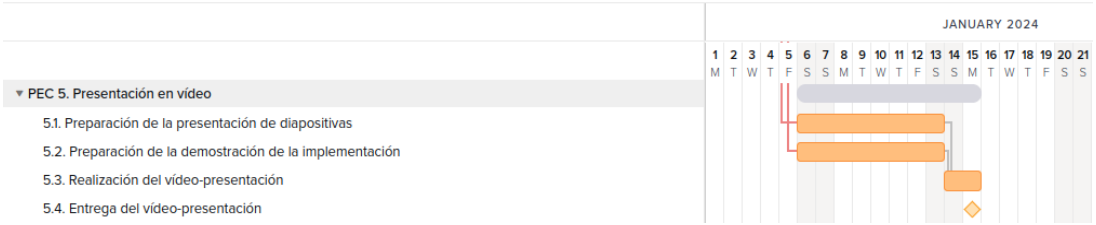


Figura 8: Diagrama de Gantt de la fase PEC 5

1.6. Análisis de riesgos

El siguiente análisis permitirá evaluar y gestionar los riesgos o amenazas relevantes a los que está expuesto este Trabajo para poder mitigarlos.

1.6.1 Matriz de riesgo

Se define una matriz simplificada de clasificación del riesgo junto con la definición de sus distintos niveles de probabilidad y de impacto. Mediante ella se valorarán los riesgos identificados.

		Impacto			
		Bajo (1)	Medio (2)	Alto (3)	Crítico (5)
Probabilidad	Baja (1)	2	3	4	6
	Media (2)	3	4	5	7
	Alta (3)	4	5	6	8

Riesgo bajo (2-3) Riesgo medio (4-5) Riesgo alto (6-8)

Tabla 2: Matriz de riesgos

Impacto	
Impacto Bajo	Puede afectar a aspectos formales del Trabajo, pero no a sus objetivos.
Impacto Medio	No afecta a ninguno de los objetivos del Trabajo, pero es necesario tomar algunas medidas para mitigar la materialización del riesgo.
Impacto Alto	Puede afectar a alguno de los objetivos del Trabajo si no se toman medidas para mitigar la materialización del riesgo.
Impacto Crítico	Uno o más objetivos del Trabajo no se pueden conseguir sin buscar soluciones alternativas y sin aplicar medidas para mitigar la materialización del riesgo.

Tabla 3: Niveles de impacto del riesgo

Probabilidad	
Probabilidad Baja	La materialización del riesgo ocurre pocas veces (0% - 35%)
Probabilidad Media	La materialización del riesgo ocurre ocasionalmente (36% - 60%)
Probabilidad Alta	La materialización del riesgo ocurre regularmente o muchas veces (61% -100%).

Tabla 4: Niveles de probabilidad del riesgo

1.6.2 Identificación, valoración de los riesgos

Esta sección pretende identificar y valorar los riesgos más relevantes que pueden afectar a la consecución de los objetivos del Trabajo.

ID	Riesgo	Probabilidad	Impacto	Valoración
R1	La plataforma de Wazuh pasa a ser un producto descontinuado.	Baja (1)	Crítico (5)	Alto (6)
R2	Cambios en el Trabajo propuestos por el tutor.	Media (2)	Medio (3)	Medio (5)
R3	Retrasos en las tareas.	Baja (1)	Alto (3)	Medio (4)
R4	Fallo hardware/software en los recursos necesarios.	Baja (1)	Alto (3)	Medio (4)
R5	Desconocimiento de las tecnologías y de la plataforma a implementar.	Media (2)	Medio (2)	Medio (4)
R6	Documentación incompleta de las soluciones tecnológicas necesarias.	Media (2)	Medio (2)	Medio (4)

Tabla 5: Valoración de los riesgos identificados

1.6.3 Tratamiento de los riesgos

Riesgo R1: *La plataforma de Wazuh pasa a ser un producto descontinuado.*

Es poco probable que la plataforma Wazuh pase a ser un producto descontinuado a corto o medio plazo debido a que dispone de una sólida cartera de socios a nivel mundial y de clientes reconocidos como la NASA o eBay. Además, se observan publicaciones frecuentes de nuevas versiones del producto con nuevas funcionalidades y correcciones de errores [13].

Su tratamiento, en caso de materializarse el riesgo, consistiría en seleccionar una plataforma alternativa durante la fase de estudio de soluciones de SIEM con EDR/XDR o de utilizar el software disponible si el proyecto está ya en una fase más avanzada.

Riesgo R2: *Cambios en el Trabajo propuestos por el tutor.*

Se deben seguir las indicaciones del tutor para que el TFG sea viable, por lo que se ha de ser lo suficientemente flexible para poder adaptarse a las modificaciones que se propongan. Las entregas se refinarán de manera continua durante el proyecto y la planificación temporal actual se adaptará a los requisitos indicados siempre que sea posible. En caso contrario, se optará por medidas correctoras como son la dedicación de más horas a tareas con retrasos o la replanificación de estas en el caso de ser necesario.

Riesgo R3: *Retrasos en las tareas.*

Ciertos imprevistos pueden provocar retraso en las tareas planificadas, por lo que se ha establecido una planificación temporal que pueda contemplar esta casuística y mitigarlo al menos en parte.

La medida correctora es asignar más horas a las tareas con retrasos para intentar mantener la planificación establecida y, si no es posible, proceder a la replanificación de las tareas.

Riesgo R4: *Fallo hardware/software en los recursos necesarios.*

Esta circunstancia depende de la severidad del fallo y se mitigará mediante medidas preventivas de copias de seguridad de la información, que es el activo más valioso.

Las medidas correctoras de fallo de software pueden consistir desde, consultar fuentes de información para reconfigurarlo, hasta reinstalar el sistema operativo o contenedor. Por su parte, los errores de hardware implican la reparación o sustitución del ordenador por otro similar.

Riesgo R5: *Desconocimiento de las tecnologías y de la plataforma a implementar.*

El desconocimiento de las tecnologías y de la plataforma a implementar puede suponer retrasos en las tareas planificadas.

Se establecieron medidas preventivas contemplando fases de documentación en la planificación temporal de las tareas complejas e incluso añadiendo otras específicas para el aprendizaje de las tecnologías necesarias.

La media correctora es la misma que para el riesgo R3.

Riesgo R6: *Documentación incompleta de las soluciones tecnológicas necesarias.*

Es posible que la documentación de ciertas funcionalidades o tecnologías sea incompleta o escueta. En esta situación, se tratará de mitigar el riesgo mediante la consulta de fuentes de información proporcionadas por la UOC y a través de Internet en foros de soporte o discusión, blogs, agregadores de contenido, etc. Se priorizarán fuentes de información reconocida y se contrastará la información que no proceda de fuentes oficiales.

1.7. Breve sumario de productos obtenidos

Los productos obtenidos de este Trabajo son:

- Memoria del Trabajo.
- Scripts y archivos empleados para la implementación y configuración.
- Presentación en vídeo.

1.8. Breve descripción de los capítulos de la memoria

Se proporciona una breve descripción de los capítulos de la memoria en los que se destacan los temas más relevantes que se tratan.

Introducción: Define el contexto del Trabajo junto a su justificación, sus objetivos, la metodología empleada, su planificación, su análisis de riesgos, los productos que se obtendrán y el compromiso con la ética, la sociedad, la diversidad y la sostenibilidad.

Estado del arte: Estudio del conocimiento reciente acerca de la ciberseguridad, soluciones tecnológicas frente a las ciberamenazas y tecnologías de contenedores que permiten contextualizar la temática que se va a tratar.

Análisis de soluciones SIEM de uso gratuito: Análisis de la situación actual de las plataformas de seguridad de uso gratuito con capacidades de SIEM y EDR/XDR, para concluir con una comparativa cualitativa y cuantitativa con características relevantes de los productos.

Plan de implantación: Estudio y documentación de todo el proceso de la puesta en marcha de la plataforma de seguridad Wazuh, desde el entorno que necesita para su funcionamiento hasta la instalación y configuración de los distintos componentes en el servidor y en los endpoints, teniendo siempre en mente el uso de buenas prácticas en materia de seguridad informática.

Casos de uso: Análisis de distintos módulos de la plataforma de seguridad Wazuh para hacer una selección de los que resulten relevantes para proteger los dispositivos finales frente a amenazas. Este apartado no pretende ser una guía de usuario, sino estudiar posibles casos de uso que ayuden a una organización a proteger sus dispositivos finales.

Resultados: Resumen de los resultados obtenidos en los tres apartados anteriores.

Conclusiones y trabajos futuros: Análisis y reflexión crítica de la planificación, metodología y de los resultados obtenidos en este Trabajo. Se tendrán en cuenta los imprevistos, las lecciones aprendidas y posibles nuevas líneas de trabajo.

Glosario: Recopilación en orden alfabético de definiciones de palabras y términos utilizadas en el Trabajo.

Bibliografía: Referencias a fuentes de información consultadas y a software utilizado en este Trabajo.

Anexos: Apartados de temas específicos que no se incluyen en la memoria del Trabajo por su extensión.

2. Estado del arte

2.1. Ciberseguridad

La complejidad y sofisticación de los ataques actuales hace necesario articular un conjunto de políticas de seguridad en las organizaciones que permitan una coordinación eficaz de procesos, tecnología y personal, para permitir de esta manera la mitigación y neutralización de dichas amenazas. De hecho, hoy en día la ciberseguridad debe concebirse como una función transversal dentro de los procesos de las instituciones y abarcar aspectos tan relevantes como el bastionado de los activos tecnológicos, la formación del personal en nuevas tecnologías y la implementación de un plan director de seguridad de la información que incluya buenas prácticas.

Una parte importante de los vectores de ataque están relacionados con actividades en los dispositivos de punto final o terminales según el INCIBE [14], por lo que es preciso disponer de herramientas tecnológicas que ofrezcan una protección adecuada frente a las amenazas y seguir los siguientes principios del bastionado o *hardening* de los sistemas informáticos [15]:

- Mínimo punto de exposición utilizando únicamente los componentes necesarios.
- Mínimo privilegio estrictamente necesario para que los usuarios o servicios puedan desempeñar su tarea.
- Defensa en profundidad de los distintos componentes del sistema.
- Confianza cero mientras no se ha verificado la identidad de los usuarios, procesos o dispositivos.

Artículos y estudios evidencian que las pequeñas y medianas empresas (PYME) son el objetivo principal de los ciberatacantes debido a la falta de recursos TIC y de concienciación en materia de ciberseguridad, llegando a suponer el 43% de los ciberataques totales [16]. Organismos oficiales, como la Agencia de la Unión Europea para la Ciberseguridad (ENISA) [17], la Agencia de Seguridad de Infraestructura y Ciberseguridad (CISA) de Estados Unidos [18] o el Instituto Nacional de Ciberseguridad (INCIBE) en España [19], dedican recursos para asistirlas porque son conscientes de su importancia en la economía y de los desafíos a los que se enfrentan.

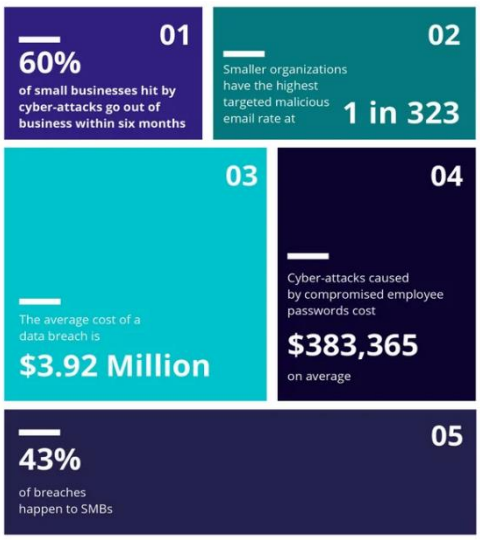


Figura 9: Estadísticas de ciberataques de las PYME [20]

2.2. Soluciones tecnológicas frente a las ciberamenazas

En la Figura 10 se observa una demanda creciente a nivel mundial en productos de seguridad para dispositivos de punto final, tal y como se desprende de los beneficios obtenidos por las empresas de dicho sector, y las previsiones son que la tendencia se mantendrá durante los próximos años.

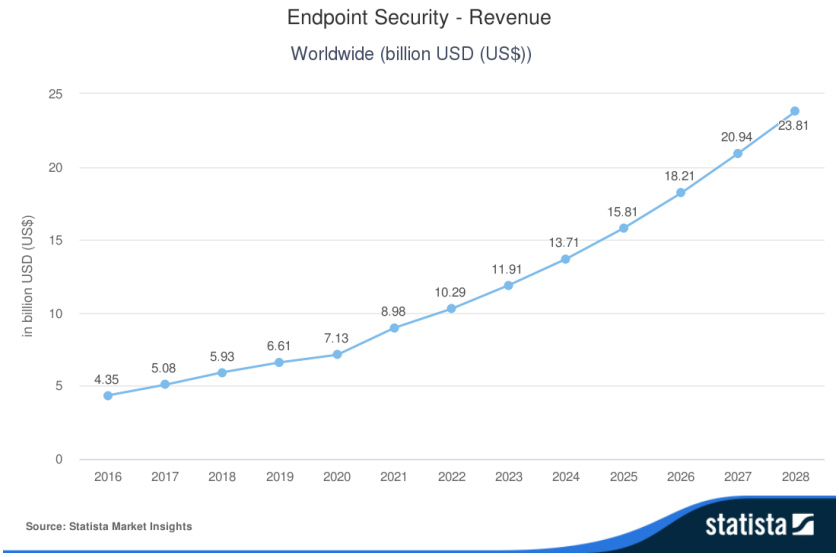


Figura 10: Ingresos por productos de la seguridad de enpoints [21]

Los endpoints se protegen habitualmente con antivirus tradicionales, conocidos como EPP (Plataforma de Protección de Endpoint), pero actualmente se combinan con otras soluciones que permiten supervisarlos para poder actuar en tiempo real frente a amenazas más avanzadas: los sistemas EDR (Detección y Respuesta de Endpoint), que recopilan información exclusivamente de los dispositivos finales, y los sistemas XDR (Detección y Respuesta Extendida), que son una evolución de los EDR al manejar además eventos de otras fuentes de la infraestructura tecnológica para poder ofrecer una visión holística de la amenaza.

Las herramientas de gestión de eventos e información de seguridad, conocidas como SIEM, permiten la recopilación de eventos de seguridad de distintos orígenes para su análisis en tiempo real. Aunque presentan funcionalidades similares con las soluciones XDR, se diferencian principalmente en que las primeras están pensadas para detectar amenazas a través del análisis de los eventos, mientras que las segundas están más enfocadas en ofrecer un tiempo de respuesta reducido con las amenazas descubiertas. Ambos productos deben verse como complementarias, más que como excluyentes [22–24].

Actualmente no existe una gran variedad de soluciones SIEM con licencias de código abierto que cuenten con la aprobación de la OSI (Iniciativa de Código Abierto) [25], entre las cuales se pueden destacar Wazuh [5] o AlienVault OSSIM [26]. Wazuh destaca por integrar capacidades de SIEM y XDR además de por utilizar las licencias GNU General Public License, version 2 y Apache License, Version 2.0 (ALv2) en sus componentes, ambas aprobadas por la OSI.

La combinación de las herramientas SIEM con las soluciones SOAR, Coordinación, automatización y respuesta de seguridad, permiten automatizar flujos complejos de

respuesta frente a las amenazas a partir de la integración de las distintas herramientas y sistemas de la plataforma de seguridad, lo que proporciona un tiempo de detección y respuesta más eficiente. Son de gran utilidad para la realización de tareas frecuentes y repetitivas de manera automática, como son la apertura de un ticket, el enriquecimiento de información a partir de la inteligencia de amenazas o la respuesta automática contra la amenaza, todo ello integrado en procedimientos de respuesta o *playbooks*.

Los directores de seguridad de las empresas (CISO) tienen en cuenta soluciones XDR, SOAR y de contenedores, entre otras, tal y como se puede ver en los resultados de una encuesta del año de 2022 (Figura 11) en la que se valora su estado actual y su implantación para los siguientes 12 meses [27].

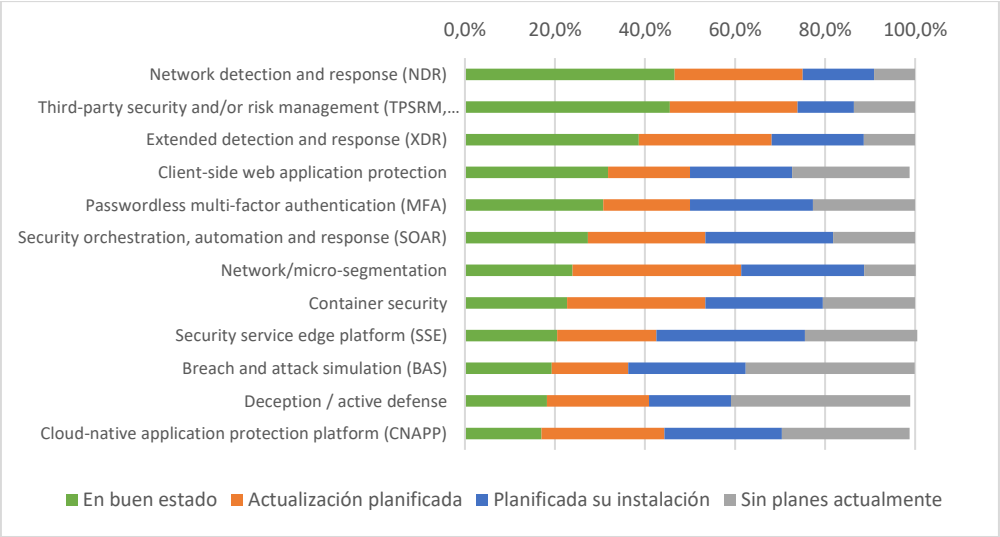


Figura 11: Tecnologías para la seguridad de la información (año 2022) [27]

2.3. Tecnología de contenedores

Hoy en día, multitud de software se ofrece mediante imágenes Docker [23] para poder ejecutarlas en contenedores, los cuales proporcionan múltiples ventajas, entre las que destacan las siguientes: son ligeros, al utilizar el kernel del sistema anfitrión y no necesitar de un sistema operativo completo, son portables e independientes de la plataforma, al disponer de todas las dependencias necesarias, y son seguros, al proporcionar aislamiento de las aplicaciones que ejecutan. Existen, además, la herramienta Docker Compose, para el despliegue de aplicaciones multicontenedor, y el motor de orquestación Kubernetes [28], que es apto para entornos complejos donde se debe manejar un número elevado de contenedores.

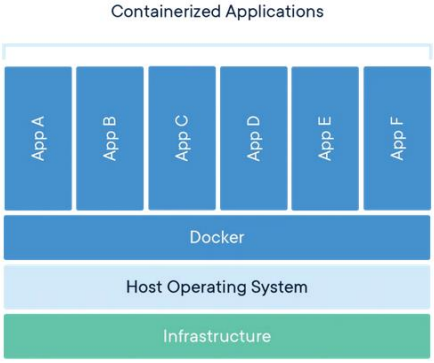


Figura 12: Ejecución de aplicaciones en contenedores [29]

El uso de contenedores en las organizaciones es elevado y casi la mitad se despliegan y gestionan a través de Kubernetes según estudios recientes [30].

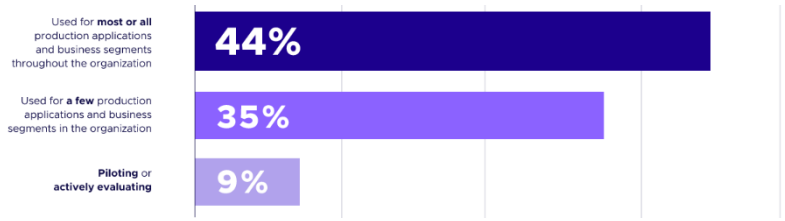


Figura 13: Uso de contenedores dentro de las organizaciones [30]

Docker, que se caracteriza por gestionar todo el ciclo de vida de los contenedores con containerd y por ejecutarlos por defecto a través del motor de ejecución (runtime) runC, es la opción predominante como plataforma y como motor de ejecución de contenedores [31–33].

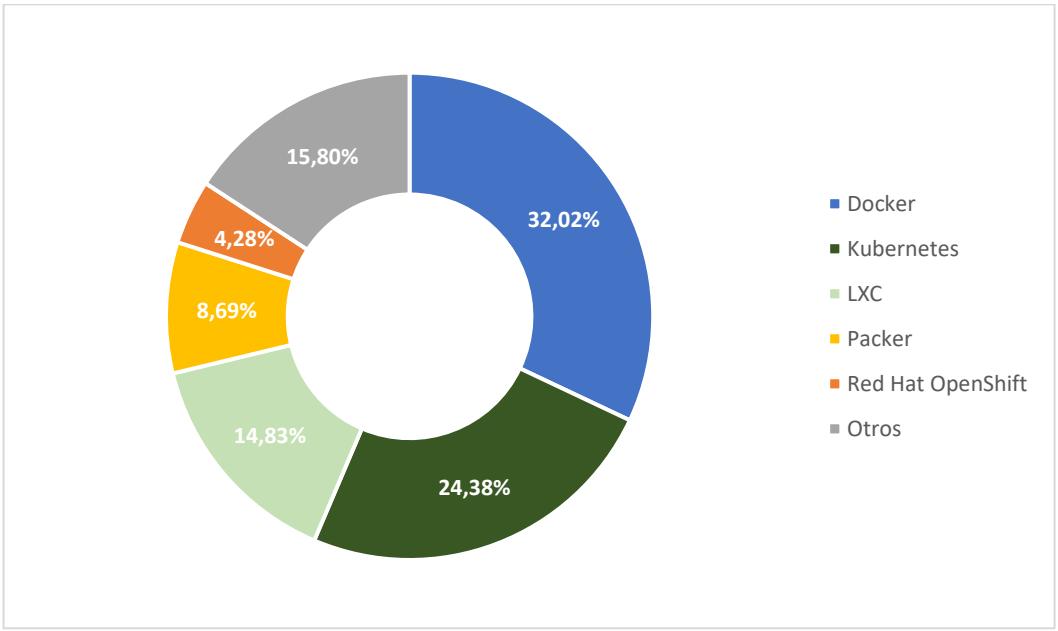


Figura 14: Cuota de mercado de productos de contenedores [33]

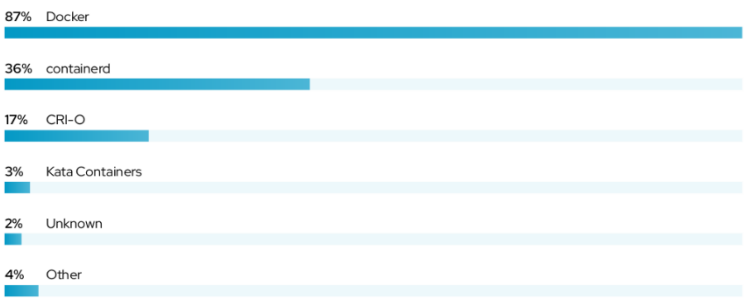


Figura 15: Tecnologías de motor de ejecución de contenedores en Kubernetes [31]

3. Análisis de soluciones SIEM de uso gratuito

La revisión bibliográfica de artículos y la consulta de información en Internet [34,35] sirve como punto de partida para analizar la situación actual de productos SIEM de código abierto o comercial de uso gratuito, donde se descartan los que se han dejado de desarrollar (deprecado) o han pasado a ser productos exclusivamente comerciales de pago. Se usa la denominación gratuita para englobar tanto al software de código abierto aprobado por la OSI, que podría ser de pago, aunque no sea habitual, y al privativo, que no cobra por su licencia de uso. Algunos de los productos privativos gratuitos analizados presentan funcionalidades limitadas, tal y como se observa en los resultados, y acostumbran a incluir un periodo de prueba limitado en su versión de pago para poder comprobar sus funcionalidades.

Se procede, por tanto, a crear una comparativa cualitativa y cuantitativa (Tabla 7) a partir de los siguientes criterios relevantes de una solución SIEM, para la que se tendrá en cuenta la información disponible de los productos gratuitos en su sitio web oficial. Se les ha dado más peso a los criterios que valoran las distintas opciones de despliegue, las funcionalidades básicas de un SIEM, la existencia de una buena documentación y la disponibilidad de distintas opciones de soporte, al considerarse fundamentales para el despliegue en distintos entornos corporativos.

La selección de los criterios se basa en los siguientes aspectos fundamentales:

- El tipo de licencia, que restringe el estudio de soluciones SIEM al ámbito definido.
- Las opciones de despliegue y escalabilidad disponibles, que determinan la flexibilidad de la solución para adaptarse a distintos entornos TIC y a necesidades cambiantes.
- El conjunto de funcionalidades y herramientas, teniendo en cuenta el estado del arte e información de soluciones SIEM actuales para hacer frente a las tácticas, técnicas y procedimientos utilizados por los ciberatacantes.
- El soporte y documentación disponible, que se considera imprescindible en la implantación, manejo y en la resolución de errores en la solución SIEM.

Criterios
Licencia/Licencias: Nombres de las licencias gratuitas utilizadas en el producto. Obtienen 1 punto las de tipo de código abierto aprobadas por la OSI (Iniciativa para el código abierto) y 0,5 puntos las restantes.
Opciones despliegue: Se recogen las opciones de instalación on-premise que hay disponibles. Se obtiene la máxima puntuación de 2 puntos si se disponen de al menos de tres métodos diferentes: binarios o imagen ISO, imágenes Docker o despliegue mediante Kubernetes, y alguna herramienta de automatización. Para dos métodos distintos se obtienen 1,5 puntos y para uno solo 0,5 puntos.
Servicio en la nube: Se indica si se ofrece el producto como servicio en la nube, obteniendo en este caso 1 punto.
Escalabilidad: Alcanzan 1 punto las soluciones que pueden escalar horizontalmente para adaptarse a nuevas necesidades o requisitos.

Criterios
Gestión, normalización, correlación y consulta de logs: Este conjunto de funcionalidades se consideran imprescindibles para un producto SIEM, por lo que se obtienen 2 puntos si se cumplen todas y 0 en el resto de los casos.
Dashboard (Panel de control): Los productos SIEM con Dashboard propio o preconfigurado para su manejo, gestión y visualización de resultados obtienen 1 punto.
Alertas: Las soluciones que proporcionan alertas frente a eventos de interés se puntúan con 1 punto.
Informes: La capacidad de generar informes a partir de los datos recopilados en el SIEM se puntúan con 1 punto.
XDR: La detección y respuesta extendida se valora con 1 punto si viene incluida de serie en el producto.
SOAR: Wazuh incluye de fábrica la integración con la solución externa Shuffle SOAR de código abierto, que presenta la ventaja de poder usarse de manera gratuita con instalación propia o como servicio en la nube. Se puntúa en este caso con 0,5 puntos al depender la funcionalidad SOAR de un producto externo.
Machine Learning: Sólo las soluciones SIEM que incluyen complementos o utilidades de aprendizaje automático para detectar amenazas alcanzan la puntuación de 1 punto.
Caza de amenazas: Se puntúa con 1 punto a los productos que incluyen funcionalidades que permitan la búsqueda proactiva de amenazas en el entorno TIC que se monitoriza.
Detección de vulnerabilidades / Análisis de seguridad: Las soluciones que incluyen módulos que permiten determinar debilidades en los nodos de la red alcanzan la puntuación de un punto.
Soporte y Documentación: Se considera relevante para un entorno corporativo que un producto gratuito combine una buena documentación junto con un soporte de la comunidad y profesional adecuado. Por ello, se ha establecido la siguiente graduación de puntuación: ▪ Soporte: Puede alcanzar la puntuación máxima de 3 puntos sin incluye el soporte de la comunidad, el soporte profesional y los servicios de formación. ▪ Documentación: documentación escasa (0,5 puntos), documentación básica (1 punto), documentación completa (1,5 puntos) y documentación extensa (2 puntos).
Comentarios: En este apartado, que no puntúa, se pretende enfatizar las características o limitaciones relevantes del producto SIEM que no se han analizado en criterios anteriores y que pueden provocar que se descarte al no ser válido para entornos TIC concretos.

Tabla 6: Criterios de comparación

	AlienVault OSSIM	Elastic Stack	Graylog Open	Security Onion	Splunk Free	Wazuh
Licencia/Licencias *Máximo 1 punto.*	GNU GPL v2.	Elastic License 2.0 (ELv2) y SSPL (uso libre).	SSPL (uso libre).	Elastic License 2.0 (ELv2).	Free Splunk License (uso libre)	GNU GPL v2 y Apache License, Version 2.0 (ALv2).
	1 punto.	0,5 puntos.	0,5 puntos.	0,5 puntos.	0,5 puntos.	1 punto.
Opciones despliegue *Máximo 2 puntos.*	ISO.	Binarios. Docker. Ansible, Puppet, Chef (comunidad).	Binarios. Docker. Ansible, Puppet, Chef (oficiales).	ISO, repositorio Git. Imágenes Docker en ISO.	Binarios. Docker. Kubernetes	Binarios, OVA, AMI. Docker, Kubernetes. Ansible, Puppet.
	0,5 puntos.	2 puntos.	2 puntos.	1,5 puntos.	1,5 puntos.	2 puntos.
Servicio en la nube *Máximo 1 punto.*	Si, AlienVault USM Anywhere	Si, Elastic Cloud.	Si, Graylog Cloud.	Si.	Si, Splunk Cloud Platform	Si, Wazuh Cloud.
	1 puntos.	1 punto.	1 punto.	1 punto.	1 punto.	1 punto.
Escalabilidad *Máximo 1 punto.*	No.	Si.	Si.	Si.	No.	Si.
	0 puntos.	1 punto.	1 punto.	1 punto.	0 puntos.	1 punto.
Gestión, normalización, correlación y consulta de logs *Máximo 2 puntos.*	No incluye gestión de logs.	Si	No dispone de motor de correlación de eventos.	Si	Si	Si.
	0 puntos.	2 puntos.	0 puntos	2 puntos.	2 puntos.	2 puntos.
Dashboard (Panel de control) *Máximo 1 punto.*	Si.	Si.	Si.	Si.	Si.	Si.
	1 punto.	1 punto.	1 punto.	1 punto.	1 punto.	1 punto.
Alertas *Máximo 1 punto.*	Si.	Si.	Si.	Si.	No	Si.
	1 punto.	1 punto.	1 punto.	1 punto.	0 puntos.	1 punto.
Informes *Máximo 1 punto.*	Si.	Si.	No en la versión gratuita. Disponible en Graylog Operations.	No. Son necesarios productos externos de pago (Kibana +Skedler).	Si.	Si.
	1 punto.	1 punto.	0 puntos.	0 puntos.	1 puntos	1 punto.
XDR *Máximo 1 punto.*	No en la versión Open Source.	Si, con Elastic Security.	No.	No.	No.	Si.
	0 puntos.	1 punto.	0 puntos.	0 puntos.	0 puntos.	1 punto.
SOAR *Máximo 1 punto.*	No en la versión Open Source.	No en versión gratuita.	No.	No.	No.	Si, incluye integración con Suffle SOAR.

	AlienVault OSSIM	Elastic Stack	Graylog Open	Security Onion	Splunk Free	Wazuh
	0 puntos.	0 puntos.	0 puntos.	0 puntos.	0 puntos.	0,5 puntos.
Machine Learning *Máximo 1 punto.*	No en la versión Open Source.	No en versión gratuita.	No en versión gratuita.	Si, con la utilidad logscan.	A través de aplicaciones y complementos desarrollados para Splunk.	No, permite integración con productos externos.
	0 puntos.	0 puntos.	0 puntos.	1 punto.	1 punto.	0 puntos.
Caza de amenazas (Threat Hunting) *Máximo 1 punto.*	Si.	Si.	No en versión gratuita. Disponible en Graylog Security.	Si.	A través de aplicaciones y complementos desarrollados para Splunk.	Si.
	1 punto.	1 punto.	0 puntos.	1 punto.	1 punto.	1 punto.
Detección de vulnerabilidades / Análisis de seguridad *Máximo 1 punto.*	Si.	Si.	No.	No.	No.	Si.
	1 punto.	1 punto.	0 puntos.	0 puntos.	0 puntos.	1 punto.
Soporte *Máximo 3 puntos.*	Comunidad. Profesional.	Comunidad, Profesional. Formación.	Comunidad. Profesional.	Comunidad. Profesional. Formación.	Comunidad. Profesional. Formación.	Comunidad. Profesional. Formación.
	2 puntos.	3 puntos.	2 puntos.	3 puntos.	3 puntos.	3 puntos.
Documentación *Máximo 2 puntos.*	Escasa	Muy buena.	Muy buena.	Buena.	Muy buena	Muy buena.
	0,5 punto.	2 puntos.	2 puntos.	1,5 puntos.	2 puntos.	2 puntos.
Comentarios *No puntúa.*	Limitaciones relevantes para aplicar en un entorno corporativo.	Incluye XDR en Elastic Security con la licencia gratuita.	Licencia gratuita de Graylog Operations a pequeñas empresas (máximo 2GB/día)	No es un SIEM, pero incluye parte de sus funcionalidades. Dispone de Appliances (Hardware + software).	Permite indexar 500 MB por día. No permite roles ni cuentas de usuario. Limitaciones relevantes para aplicar en un entorno corporativo.	Es un SIEM con XDR.
Puntuación Total	**10**	**17,5**	**10,5**	**14,5**	**14**	**18,5**

Tabla 7: Comparativa de soluciones SIEM de uso gratuito

Como resumen final se concluye lo siguiente:

- Las soluciones AlienVault OSSIM [26] y Graylog Open [36] no incluyen todas las funciones imprescindibles de un SIEM: recopilación y gestión de logs de distintas fuentes, su normalización y su posterior visualización.
- El producto Security Onion [37] no es catalogado como un producto SIEM, pero incluye parte de sus funcionalidades que pueden ser suficientes para obtener visibilidad de red y de hosts en ciertos entornos TIC.
- AlienVault OSSIM, Graylog Open y Splunk Free [38] presentan limitaciones relevantes para funcionar en un entorno corporativo.
- Wazuh [5] y Elastic Stack [39] son los productos más completos de los evaluados y presentan funcionalidades muy similares como SIEM con XDR integrado de uso gratuito.

Finalmente, se incluye en la Figura 16 un diagrama de columnas con la puntuación total obtenida por cada una de las soluciones analizadas según los criterios indicados.

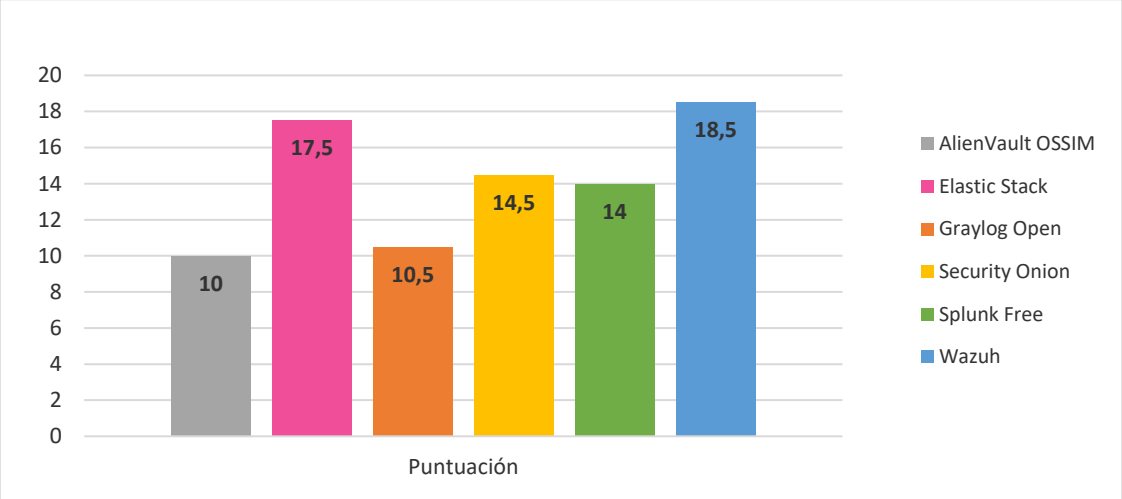


Figura 16: Puntuación final de las soluciones SIEM analizadas

4. Plan de implantación

La implantación de la solución se estructura en las siguientes seis fases que se irán detallando en los apartados correspondientes.

Figura 17: Fases de implantación de la plataforma de seguridad Wazuh

4.1. Descripción de los componentes principales

La plataforma de seguridad Wazuh se fundamenta en el despliegue de agentes en los dispositivos finales y en tres componentes centrales: el indexador, el servidor o gestor y el dashboard o panel de control [40].

El indexador (Wazuh indexer) utiliza el motor de búsqueda y análisis distribuido OpenSearch [41] para almacenar información de eventos, alertas, monitorización y estadísticas procedentes del gestor Wazuh. Destaca por ofrecer sus capacidades en casi tiempo real y por soportar configuraciones tanto en mononodo como en multinodo, obteniéndose en este último caso alta disponibilidad y escalabilidad.

El gestor de Wazuh (Wazuh manager) se encarga de la administración de los agentes de los dispositivos finales y de recibir sus eventos para examinarlos. Su motor de análisis, basado en el HIDS de código abierto OSSEC, se encarga de originar alertas cuando se detectan amenazas o anomalías, apoyándose en fuentes de inteligencia de amenazas, en el comportamiento de los ciberatacantes según MITRE ATT&CK, en la evaluación de la configuración de seguridad, en los requerimientos de cumplimiento normativo aplicables y en el análisis de vulnerabilidades. Por otra parte, el componente Filebeat de Wazuh manager permite enviar sus eventos y alertas a Wazuh Indexer y, si es necesario, Wazuh manager puede escalarse horizontalmente mediante su despliegue en un clúster combinado con un balanceador de carga.

El panel de control de Wazuh (Wazuh dashboard) es una interfaz web que, conectándose a la interfaz API de Wazuh manager, permite varios conjuntos de tareas principales:

- Extracción, análisis y visualización de datos.
- Configuración y administración de los agentes.
- Gestión y monitorización del estado de la plataforma.
- Herramientas de desarrollo para crear reglas y decodificadores que ayudan a detectar amenazas o vulnerabilidades.

Los agentes de Wazuh son multiplataforma para poder monitorizar una amplia variedad de endpoints y necesitan de unos 35 MB de RAM de media para poder funcionar. Presentan un diseño modular, cada uno con una función específica:

- Recolector de logs con capacidad de añadirle metadatos.
- Ejecución de comandos de manera periódica para obtener sus resultados.
- Monitorización de la integridad de los ficheros a través del módulo FIM.
- Evaluación de la configuración de seguridad (SCA) basado en perfiles de configuración CIS (Centro para la seguridad en Internet).
- Inventariado de los sistemas monitorizados.

- Detección de malware mediante firmas o la detección de anomalías.
- Respuesta activa frente a amenazas detectadas.
- Monitorización de la seguridad de los contenedores Docker.
- Monitorización de la seguridad en la nube.

4.2. Diseño de la solución

La arquitectura que se elige es la mínima indispensable para poner en marcha la plataforma de seguridad Wazuh y que, según su guía de inicio rápido, es suficiente para monitorizar hasta 100 *endpoints* con los siguientes requisitos de hardware:

vCPU	RAM	Storage	# Agentes
4	8 GiB	50 GB	1-25
8	8 GiB	100 GB	25-50
8	8 GiB	200 GB	50-10

Tabla 8: Requisitos de hardware

La Figura 18 contiene la arquitectura y el diseño de red propuesto que se simulará con el uso de tres máquinas virtuales del software de virtualización Oracle VM VirtualBox en una red NAT con acceso a Internet.

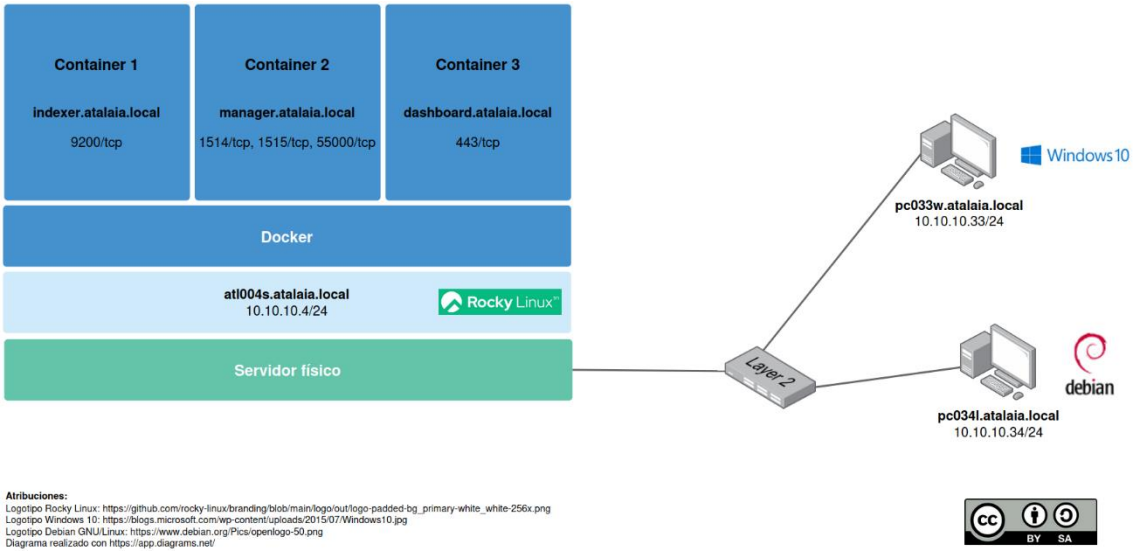


Figura 18: Diseño de red de la plataforma de seguridad Wazuh

# VM	Nombre VM	Comentario
VM1	atalaia-atl004s	Representa a un servidor que contiene los componentes centrales de la plataforma de seguridad Wazuh
VM2	atalaia-pc033w	Representa a un endpoint con SO Windows 10
VM3	atalaia-pc034l	Representa a un endpoint con SO Debian GNU/Linux 12

Tabla 9: Máquinas virtuales utilizadas en el diseño de red propuesto

El diseño propuesto supone una serie de peticiones de establecimiento de conexión TCP para cada uno de sus servicios, que se representan en la Figura 19 y en donde se aprecia qué componentes centrales de la plataforma Wazuh presentan dependencias entre sí para que puedan funcionar correctamente:

- El gestor de Wazuh hace uso del indexador.
- El panel de control de Wazuh hace uso tanto del indexador como del gestor.

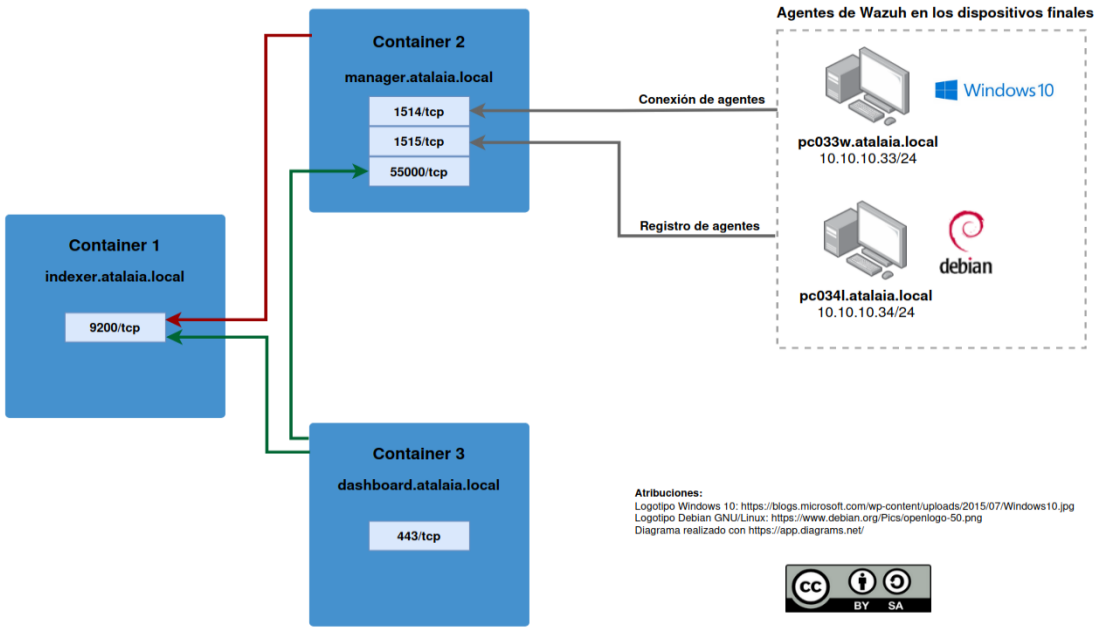


Figura 19: Peticiones de tráfico previstas en el diseño propuesto

4.3. Entorno para la plataforma de seguridad Wazuh

La instalación de la plataforma de seguridad de Wazuh según el diseño propuesto tiene una serie de requisitos previos que se enumeran a continuación y cuyo despliegue se detalla en el Anexo I:

- Creación de las máquinas virtuales VM1, VM2 y VM3.
- Instalación y configuración del sistema operativo seleccionado para cada una de las máquinas virtuales.
- Instalación de Docker Engine, Docker Compose y OpenSSL en VM1.

Como sistema operativo de la máquina virtual atalaia-atl004s, se ha optado por seleccionar una distribución de código abierto de uso gratuito que sea compatible con Red Hat Enterprise Linux (RHEL), con el objetivo de facilitar la migración a una solución empresarial en caso de ser necesario. Red Hat dispone de documentación para convertir las distribuciones Alma Linux, CentOS Linux, Oracle Linux y Rocky Linux a la suya propia [42], aunque CentOS Linux ya no es una opción válida debido a que finaliza su soporte el 30 de junio de 2024 [43].

Tal y como se puede ver en la Figura 20 [44], el repositorio EPEL (Paquetes adicionales para Linux empresarial) de Fedora Project sirve de indicador para hacer una estimación de la adopción de distribuciones Linux compatibles con RHEL, donde destaca actualmente Rocky Linux. Además de lo expuesto, se opta por la versión 9 de dicha distribución Rocky Linux en el diseño planteado porque incluye actualizaciones hasta el 31 de mayo de 2032 [45], soporte comercial y las aplicaciones necesarias para el despliegue de la solución.

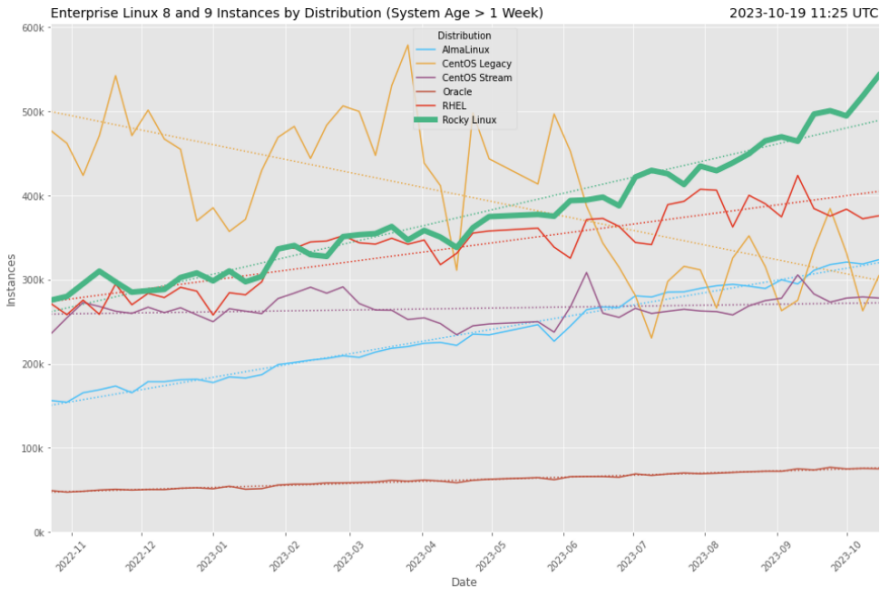


Figura 20: Nº de instancias por distribución según el uso de repositorios EPEL [44]

Siguiendo los principios de bastionado y defensa en profundidad, se tienen en cuenta algunos aspectos de protección adicionales a la hora de la implementación de la máquina atalaia-atl004s.

- Se ha habilitado el arranque seguro (Secure Boot) en la máquina virtual para que sólo las aplicaciones de arranque autorizadas puedan iniciar el sistema.
- Se ha protegido el gestor de arranque GRUB para que sólo los usuarios validados puedan alterar o manipular el arranque definido.
- Se ha utilizado un perfil de seguridad de nivel 2 para servidores proporcionado por el CIS, Centro para la Seguridad en Internet, con el fin de aplicar una serie de configuraciones en el sistema que garanticen una protección elevada basada en el concepto de defensa en profundidad [46].
- Se ha configurado el cortafuegos Firewalld con políticas de cero confianza (zero trust) y se ha permitido el tráfico estrictamente necesario para la red de trabajo.
- Se han aplicado restricciones en determinadas particiones del sistema.

Además, el sistema operativo Rocky Linux dispondrá de la tecnología de contenedores multiplataforma Docker Engine y de la utilidad Docker Compose, con la finalidad de desplegar cada uno de los componentes centrales de la plataforma Wazuh en su propio contenedor.

# Contenedor	Nombre VM	Puertos de escucha en la IP 10.10.10.4	
Contenedor 1	indexer.atalaia.local	9200/tcp	API RESTful del indexador
Contenedor 2	manager.atalaia.local	1514/tcp 1515/tcp 55000/tcp	Servicio para la conexión de los agentes Servicio para el registro de agentes API RESTful del gestor de Wazuh
Contenedor 3	dashboard.atalaia.local	443/tcp	Interfaz web del panel de control

Tabla 10: Puertos de escucha asociados a los servicios disponibles

Se hace necesario contemplar las distintas opciones de conectividad de red que ofrece Docker Engine, para escoger la que se adecúe de forma óptima según sus particularidades y funcionamiento.

El driver de red *host* destaca porque los contenedores utilizan directamente la pila de red del host anfitrión Docker para abrir los puertos con los que se ofrecen sus servicios. Esto provoca que no necesiten una dirección IP propia y que presenten un buen rendimiento al prescindir de reglas NAT para cada puerto disponible.

La modalidad de red *bridge* de Docker inserta reglas de filtrado de paquetes de tipo DNAT mediante la utilidad iptables cuando se publican los puertos abiertos de los contenedores en el host anfitrión, con el inconveniente de que sus recursos pueden estar accesibles a pesar de que el cortafuegos local no tiene explícitamente permitido dicho tráfico. En esta situación es necesario añadir las reglas necesarias al comienzo de la cadena personalizada de filtrado DOCKER-USER para cumplir los requisitos de seguridad deseados [47].

Las modalidades de red *ipvlan* permiten un buen control sobre el direccionamiento IP, pero para la propuesta indicada presenta inconvenientes a la hora de filtrar tráfico de capa 3 en su modalidad L2, a causa de que trabaja con tramas de capa 2 del modelo OSI, y de conectividad en su modalidad L3, debido a limitaciones de la red NAT de Virtualbox con el manejo rutas estáticas.

En este caso se ha optado por el driver *host* porque es la alternativa que combina la flexibilidad para la implementación y el control del tráfico de red sin añadir complejidad al entorno de Virtualbox elegido: no es necesario publicar los puertos asociados a los servicios, permite el filtrado de paquetes de capa 3 a través de cortafuegos del sistema (Firewalld) y no presenta los problemas de conectividad de ipvlan L3S cuando se dispone de varias subredes.

Drivers de Docker	Publicación de puertos transparente	Filtrado de paquetes de red de capa 3	Sin problemas de conectividad en la red NAT de Virtualbox
host	✓	✓	✓
bridge	✗	✗	✓
ipvlan L2	✓	✗	✓
ipvlan L3S	✓	✓	✗

Tabla 11: Comparativa de modalidades de red de Docker

Con respecto a las máquinas virtuales que simulan a dos endpoints, se ha optado por partir de una instalación básica de Microsoft Windows 10 y de Debian GNU/Linux 12 con GUI sin aplicarles ningunas medidas de protección adicionales. Ambos sistemas operativos están soportados por Wazuh para la ejecución de su agente y la configuración indicada servirá para hacer una evaluación inicial de su estado de seguridad.

La empresa Lansweeper, desarrolladora de su plataforma de gestión e inventariado de activos de tecnologías de la información, indica que la adopción mayoritaria es para Windows 10 de entre los sistemas operativos de escritorio de Microsoft. Esto se debe a distintos factores como, la falta de madurez de Windows 11, los nuevos requisitos de hardware que provocan que no todo el hardware sea elegible y la existencia de soporte para Windows 10 hasta el 14 de octubre de 2025 [48,49], que se extiende al 9 de enero de 2029 para su versión Enterprise LTSC 2019 (canal de mantenimiento a largo plazo) [50].

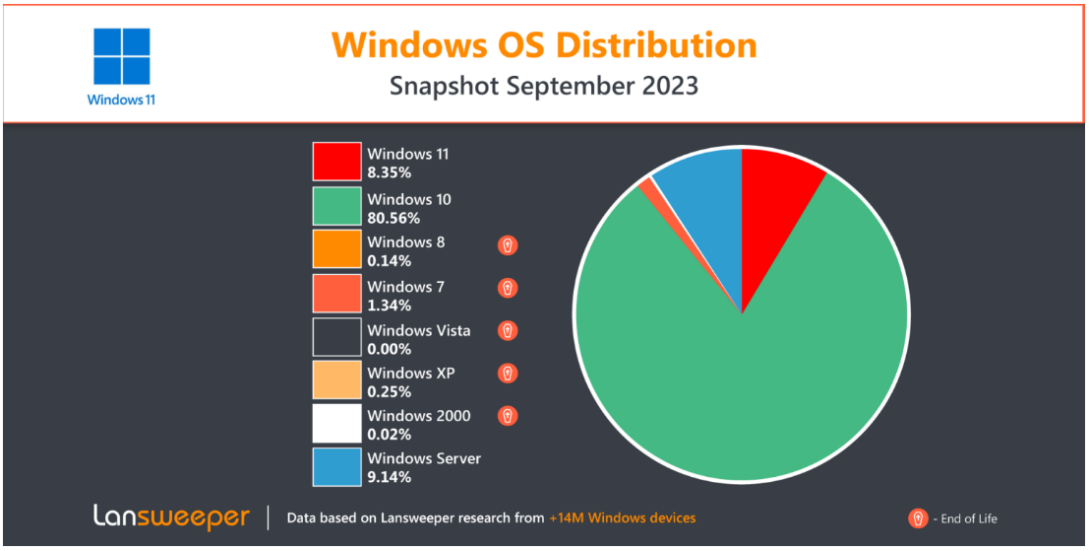


Figura 21: Adopción del SO de escritorio Microsoft Windows [49]

La elección de Debian GNU/Linux 12 como SO de escritorio se debe a que se trata de una veterana distribución con una comunidad de usuarios consolidada que se puede adaptar a entornos corporativos al ofrecer versiones de soporte a largo plazo. Además, incluye software libre y de código abierto de uso gratuito.

4.4. Generación de certificados

OpenSSL es un conjunto de herramientas y librerías criptográficas de código abierto que se emplea ampliamente para la generación de certificados digitales, con los cuales se pueden proteger la confidencialidad, autenticidad e integridad de las comunicaciones de los distintos servicios de la plataforma.

Un aspecto crucial a la hora de generar los certificados digitales es seleccionar una longitud de clave adecuada para el algoritmo criptográfico asimétrico utilizado, por lo que se tendrán en cuenta las recomendaciones de la Capacidad de Respuesta a Incidentes de Seguridad de la Información del Centro Criptológico Nacional (CCN-CERT) y del propio Centro Criptológico Nacional (CCN) [51,52].

Las claves RSA son las más utilizadas actualmente, pero ofrecen un rendimiento inferior a las de los algoritmos ECDSA (Elliptic Curve Digital Signature Algorithm) al necesitar una longitud de clave superior para ofrecer un nivel de seguridad equivalente. Por su parte, ECDSA se caracteriza por soportar distintas curvas elípticas, de entre las cuales las del NIST presentan un mejor rendimiento que las Brainpool [53], y presenta el inconveniente de que puede no funcionar en navegadores o clientes web antiguos

En la siguiente tabla se recogen los tamaños de clave recomendados para un nivel de seguridad equivalente de los algoritmos citados.

Algoritmo	Longitud de clave (bits)	Tipo de algoritmo
RSA	3072	Factorización de número enteros
ECDSA - secp256r1	256	Criptografía de curva elíptica - Curva del NIST [FIPS186-4, Apéndice D.1.2]
ECDSA - brainpoolP256r1	256	Criptografía de curva elíptica - Curva Brainpool [RFC5639]

Tabla 12: Tamaños de clave recomendados [52]

Otro aspecto que se debe tener en cuenta con los certificados para dispositivos finales es la función resumen o hash que se utiliza para firmarlos, debiéndose prescindir de MD5 y SHA-1 para optar por SHA2 o SHA3 [51,52].

Siguiendo las buenas prácticas indicadas, los certificados necesarios para la implementación del diseño propuesto se basan en el algoritmo RSA, al no estar soportados los de curva elíptica en la plataforma de Wazuh, y se firman con la función resumen SHA512 (SHA2-512). Además, su creación se automatiza mediante el script mk-atalaia-certificates.sh, cuyo código y funcionamiento se detallan en el Anexo II.

- Creación del certificado de la autoridad de certificación (CA):

```
[boole@atl004s atalaia]$ ./mk-atalaia-certificates.sh rsa root_ca
```

- Fichero que contiene el nombre común (CN) y la dirección IP que se usan para generar los certificados necesarios.

```
[boole@atl004s atalaia]$ cat certificados_atalaia.txt
admin           10.10.10.4
indexer         10.10.10.4
manager         10.10.10.4
dashboard       10.10.10.4
pc033w          10.10.10.33
pc034l          10.10.10.34
```

- Generación de los certificados indicados en el fichero anterior.

```
[boole@atl004s atalaia]$ ./mk-atalaia-certificates.sh rsa certificados_atalaia.txt
```

Los certificados y sus claves privadas asociadas se crean en los siguientes directorios:

Directorio	Descripción
/opt/atalaia/backups/root-certs	Certificados y claves privadas de CA
/opt/atalaia/backups/certs	Certificados y claves privadas de dispositivos y servicios
/opt/atalaia/config/certs	Certificados y claves privadas necesarias para el despliegue de la plataforma de Wazuh a través de Docker Compose

Tabla 13: Rutas de los certificados y claves privadas

No se copia la clave privada de la CA al directorio /opt/atalaia/config/certs porque su único fin es la firma de los nuevos certificados que emita, por lo que se debe mantener a buen recaudo.

4.5. Despliegue de los componentes centrales

Docker Compose es una herramienta que permite definir y ejecutar aplicaciones multicontenedor a través de un fichero de configuración YAML ubicado en el directorio de ejecución. Su utilización es la base desde la que Wazuh proporciona sus despliegues a través de Docker y sirve como punto de partida para realizar las adaptaciones necesarias al diseño propuesto [54].

La implementación de los componentes centrales de la plataforma Wazuh se realiza en el directorio /opt/atalaia/ y consta de los siguientes archivos organizados en subdirectorios.

```
[boole@atl004s atalaia]$ tree -I backups --matchdirs -a /opt/atalaia/
/opt/atalaia/
├── certificados_atalaia.txt
├── config
│   ├── certs
```

```
│   │   ├── admin.crt
│   │   ├── admin.key
│   │   ├── atalaia-root-ca.crt
│   │   ├── dashboard.crt
│   │   ├── dashboard.key
│   │   ├── indexer.crt
│   │   ├── indexer.key
│   │   ├── manager.crt
│   │   └── manager.key
│   ├── wazuh_dashboard
│   │   ├── opensearch_dashboards.keystore
│   │   ├── opensearch_dashboards.yml
│   │   └── wazuh.yml
│   ├── wazuh_indexer
│   │   ├── internal_users.yml
│   │   └── wazuh.indexer.yml
│   └── wazuh_manager
│       ├── api.yaml
│       ├── authd.pass
│       ├── cp-filebeat-files.sh
│       ├── filebeat.keystore
│       ├── filebeat.yml
│       └── wazuh_manager.conf
├── docker-compose.yml
├── .env
└── mk-atalaia-certificates.sh
```

El archivo docker-compose.yml, descrito en el Anexo III, tiene uno de los nombres que Docker Compose busca por defecto en el directorio de trabajo para interpretarlo y proceder al despliegue en contenedores de los distintos servicios definidos [55]. Consta de dos elementos de nivel superior: services, obligatorio y a partir del cual se definen los servicios concretos, y volumes, que se utiliza para establecer los volúmenes que posibilitan la persistencia de la información. Complementariamente, en cada servicio concreto se utilizan puntos de montaje de tipo bind con la finalidad de proporcionarles archivos de configuración, almacenes de claves (keystores) y certificados disponibles en los distintos subdirectorios indicados.

Los servicios wazuh-indexer, wazuh-manager y wazuh-dashboard especificados disponen de una sección de construcción, denominada build, que permite la declaración de comandos mediante el atributo dockerfile_inline para adaptar las distintas imágenes a las necesidades de la implementación. Además, dichos servicios utilizan la propiedad healthcheck para realizar comprobaciones periódicas de su correcto funcionamiento, es decir, se verifica si presentan un estado saludable (healthy), y sirve para establecer las dependencias de inicio y apagado existente entre los servicios (Figura 19) a través de la definición de la opción depends_on.

El fichero oculto .env contiene variables que afectan de manera global al fichero de docker-compose.yml y, adicionalmente, las credenciales de acceso a la API del gestor de Wazuh debido a que no se dispone de un keystore propio para su almacenamiento [56].

```
DNS_SUFFIX="atalaia.local"
IMAGE_VERSION="4.7.0"
MANAGER_API_USERNAME="wazuh-wui"
MANAGER_API_PASSWORD="risc-23*C1sC"
```

Fichero 1: .env

La carpeta config contiene los archivos necesarios para el despliegue, agrupados en sus correspondientes subdirectorios: certs, con los certificados y claves privadas, y wazuh_dashboard, wazuh_indexer y wazuh_manager, con los ficheros asociados a cada uno de los servicios definidos en el archivo de Docker Compose, cuya información detallada se muestra en los anexos IV, V y VI.

El despliegue de los componentes centrales se reduce al siguiente comando, aunque es posible separar la fase de construcción de la de ejecución.

```
[boole@atl004s atalaia]$ docker compose up --build --detach
[+] Building 175.3s (20/20) FINISHED
docker:default
...
[+] Running 15/15
 ✓ Volume "atalaia_wazuh_logs"               Created                                    0.0s
 ✓ Volume "atalaia_wazuh_queue"              Created                                    0.0s
 ✓ Volume "atalaia_wazuh_var_multigroups"    Created                                    0.0s
 ✓ Volume "atalaia_filebeat_etc"             Created                                    0.0s
 ✓ Volume "atalaia_filebeat_var"             Created                                    0.0s
 ✓ Volume "atalaia_wazuh_etc"                Created                                    0.0s
 ✓ Volume "atalaia_wazuh_active_response"    Created                                    0.0s
 ✓ Volume "atalaia_wazuh_agentless"          Created                                    0.0s
 ✓ Volume "atalaia_wazuh_wodles"             Created                                    0.0s
 ✓ Volume "atalaia_wazuh_api_configuration"  Created                                    0.0s
 ✓ Volume "atalaia_wazuh-indexer-data"       Created                                    0.0s
 ✓ Volume "atalaia_wazuh_integrations"       Created                                    0.0s
 ✓ Container indexer                         Healthy                                    0.0s
 ✓ Container manager                         Healthy                                    0.0s
 ✓ Container dashboard                       Started                                    0.0s
```

El estado de los servicios es funcional (healthy), tal y como se puede consultar con la siguiente instrucción modificada para restringir los campos de información que se muestran por defecto.

```
$ docker compose ps --format 'table {{.Name}}\t{{.Service}}\t{{.Status}}\t{{.Command}}'
NAME         SERVICE            STATUS                    COMMAND
dashboard    wazuh-dashboard    Up 8 minutes (healthy)    "/entrypoint.sh"
indexer      wazuh-indexer      Up 9 minutes (healthy)    "/entrypoint.sh opensearchwrapper"
manager      wazuh-manager      Up 9 minutes (healthy)    "/init"
```

Aparte de las comprobaciones del estado de salud definidas para los servicios en el fichero de Docker Compose, se incluyen otras adicionales en el Anexo VII para el diagnóstico y resolución de problemas.

4.6. Despliegue de los agentes en los endpoints

El servicio de registro de los agentes [57], disponible a través del puerto 1515/tcp del gestor de Wazuh, permite registrarlos en la plataforma de seguridad Wazuh como miembros autorizados y ofrece las siguientes ventajas: generación de claves únicas para cada agente registrado, comunicación cifrada mediante el uso de dichas claves únicas y validación de la identidad de los agentes.

Los ajustes que se indican a continuación añaden mecanismos adicionales de seguridad consistentes en la autenticación basada en contraseña junto con la verificación tanto de la identidad de gestor Wazuh como la de los agentes mediante el uso de certificados digitales.

4.6.1. Configuración en el gestor de Wazuh

Para empezar, se define una contraseña para autenticar a los agentes durante el proceso de su registro en el fichero ./config/wazuh_manager/authd.pass, el cual se proporciona al contenedor a través de un volumen de tipo bind para que acabe disponible en /var/ossec/etc/authd.pass.

```
# Puntos de montaje de ficheros de configuración en el servicio wazuh-manager
# del fichero docker-compose.yml
  - ./config/wazuh_manager/wazuh_manager.conf:/wazuh-config-mount/etc/ossec.conf
  - ./config/wazuh_manager/api.yaml:/wazuh-config-mount/api/configuration/api.yaml
  - ./config/wazuh_manager/authd.pass:/wazuh-config-mount/etc/authd.pass
  - ./config/wazuh_manager/filebeat.yml:/filebeat-config-mount/etc/filebeat/filebeat.yml

# Contenido del fichero authd.pass
[boole@atl004s atalaia]$ cat ./config/wazuh_manager/authd.pass
ag3nt-At
```

A continuación, se realizan ajustes en el servicio de registro de agentes a través de la sección XML auth del fichero ./config/wazuh_manager/wazuh_manager.conf, que se acaba cargando en /var/ossec/etc/ossec.conf cuando se inicia o reinicia el contenedor del gestor de Wazuh.

```
<ossec_config>
  ...
  <!-- Configuration for wazuh-authd -->
  <auth>
    <disabled>no</disabled>
    <port>1515</port>
    <use_source_ip>no</use_source_ip>
    <purge>yes</purge>
    <use_password>yes</use_password>
    <ciphers>HIGH:!ADH:!EXP:!MD5:!RC4:!3DES:!CAMELLIA:@STRENGTH</ciphers>
    <ssl_agent_ca>/etc/ssl/root-ca.pem</ssl_agent_ca>
    <ssl_verify_host>yes</ssl_verify_host>
    <ssl_manager_cert>/etc/ssl/filebeat.pem</ssl_manager_cert>
    <ssl_manager_key>/etc/ssl/filebeat.key</ssl_manager_key>
    <ssl_auto_negotiate>no</ssl_auto_negotiate>
  </auth>
  ...
</ossec_config>
```

Fichero 2: ossec.conf (servicio de registro de agentes)

Opción	Descripción
use_password	Uso de contraseña para el registro de los agentes.
ssl_agent_ca	Ruta del certificado de la Autoridad de Certificación para comprobar la identidad de los agentes.
ssl_verify_host	Se comprueba la identidad de los agentes.
ssl_manager_cert	Ruta del certificado del gestor de Wazuh.
ssl_manager_key	Ruta de la clave privada asociada al certificado del gestor de Wazuh.

Tabla 14: Opciones modificadas en el fichero ossec.conf

Finalmente, es necesario reiniciar el contenedor para aplicar los cambios realizados.

```
boole@atl004s atalaia]$ docker compose restart wazuh-manager
[+] Restarting 1/1
 ✓ Container manager  Started                                    3.5s
```

Es posible agrupar los agentes para simplificar la gestión de su configuración y se realice de manera centralizada, por lo que se crean dos grupos en función del tipo de sistema operativo disponible, tarea que puede realizar vía interfaz web o CLI.

```
root@manager:/var/ossec/etc/shared# /var/ossec/bin/agent_groups -a -g Linux
Do you want to create the group 'Linux'? [y/N]: y
Group 'Linux' created.
root@manager:/var/ossec/etc/shared# /var/ossec/bin/agent_groups -a -g Windows
Do you want to create the group 'Windows'? [y/N]: y
Group 'Windows' created.

root@manager:/var/ossec/etc/shared# ls -la /var/ossec/etc/shared/
total 28
drwxrwx---. 5 root  wazuh 4096 Dec  6 20:35 .
drwxrwx---. 7 wazuh wazuh 4096 Dec  6 20:14 ..
drwx------. 2 wazuh wazuh 4096 Dec  6 20:35 Linux
drwx------. 2 wazuh wazuh 4096 Dec  6 20:35 Windows
-rw-rw----. 1 wazuh wazuh   76 Nov 23 17:02 agent-template.conf
-rw-r-----. 1 root  wazuh  228 Dec  6 20:15 ar.conf
drwxrwx---. 2 wazuh wazuh 4096 Dec  6 20:15 default
```

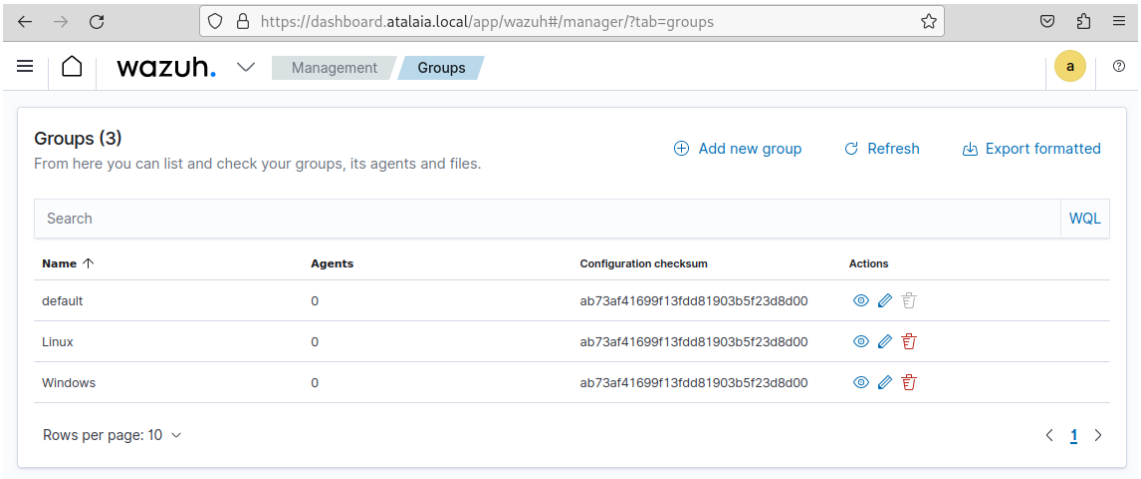


Figura 22: Visualización de grupos disponibles en el panel de control de Wazuh

4.6.2. Instalación y configuración de los agentes

En el SO Debian GNU/Linux 12, la instalación del agente propuesto se realiza mediante variables de entorno para configurar los parámetros necesarios en el fichero /var/ossec/etc/ossec.conf y la contraseña de registro establecida en el archivo /var/ossec/etc/authd.pass [58].

```
root@pc034l:~# wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-
agent_4.7.0-1_amd64.deb && \
WAZUH_MANAGER="manager.atalaia.local" \
WAZUH_AGENT_GROUP="Linux" \
WAZUH_REGISTRATION_PASSWORD="ag3nt-At" \
WAZUH_REGISTRATION_CA="etc/atalaia-root-ca.crt" \
WAZUH_REGISTRATION_CERTIFICATE="etc/pc034l.crt" \
WAZUH_REGISTRATION_KEY="etc/pc034l.key" \
dpkg -i ./wazuh-agent_4.7.0-1_amd64.deb
--2023-12-06 20:39:29--  https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-
agent_4.7.0-1_amd64.deb
Resolviendo packages.wazuh.com (packages.wazuh.com)... 18.154.48.50, 18.154.48.117,
18.154.48.95, ...
```

```
Conectando con packages.wazuh.com (packages.wazuh.com)[18.154.48.50]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 9265962 (8,8M) [binary/octet-stream]
Grabando a: «wazuh-agent_4.7.0-1_amd64.deb»

wazuh-agent_4.7.0-1_amd64.deb
100%[=============================================================================================
============================================================>]   8,84M  54,2MB/s    en
0,2s

2023-12-06 20:39:29 (54,2 MB/s) - «wazuh-agent_4.7.0-1_amd64.deb» guardado
[9265962/9265962]

Seleccionando el paquete wazuh-agent previamente no seleccionado.
(Leyendo la base de datos ... 186557 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../wazuh-agent_4.7.0-1_amd64.deb ...
Desempaquetando wazuh-agent (4.7.0-1) ...
Configurando wazuh-agent (4.7.0-1) ...
```

```
root@pc034l:~# cat /var/ossec/etc/authd.pass
ag3nt-At
```

La sección XML client del archivo /var/ossec/etc/ossec.conf contiene los ajustes necesarios para que el agente pueda registrarse y conectarse al gestor de Wazuh [59], y, como puede observarse, se usan rutas a los ficheros que son relativas con respecto a /var/ossec.

```
<ossec_config>
  ...
  <client>
    <server>
      <address>manager.atalaia.local</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>debian, debian12</config-profile>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <crypto_method>aes</crypto_method>
    <enrollment>
      <enabled>yes</enabled>
      <groups>Linux</groups>
      <server_ca_path>etc/atalaia-root-ca.crt</server_ca_path>
      <agent_certificate_path>etc/pc034l.crt</agent_certificate_path>
      <agent_key_path>etc/pc034l.key</agent_key_path>
      <authorization_pass_path>etc/authd.pass</authorization_pass_path>
    </enrollment>
  </client>
  ...
<ossec_config>
```

Fichero 3: ossec.conf (conexión de agente GNU/Linux)

La opción de configuración server_ca_path define la ubicación del certificado de CA necesario para verificar la identidad del gestor de Wazuh, mientras que agent_certificate_path y agent_key_path lo hacen para el certificado y la clave privada de los agentes respectivamente, los cuales deben ser copiados antes de iniciar el servicio del agente.

```
# Ajuste de permisos
root@pc034l:/var/ossec/etc# chown root:wazuh atalaia-root-ca.crt pc034l.*
```

```
root@pc034l:/var/ossec/etc# chmod 640 atalaia-root-ca.crt pc034l.*
root@pc034l:/var/ossec/etc# ls -la pc034l.* atalaia-root-ca.crt
-rw-r----- 1 root wazuh 1785 dic  6 20:41 atalaia-root-ca.crt
-rw-r----- 1 root wazuh 1838 dic  6 20:41 pc034l.crt
-rw-r----- 1 root wazuh 2484 dic  6 20:41 pc034l.key

# Configuración e inicio del servicio
root@pc034l:/var/ossec/etc# systemctl status wazuh-agent.service
○ wazuh-agent.service - Wazuh agent
     Loaded: loaded (/lib/systemd/system/wazuh-agent.service; disabled; preset: enabled)
     Active: inactive (dead)
root@pc034l:/var/ossec/etc# systemctl daemon-reload
root@pc034l:/var/ossec/etc# systemctl enable wazuh-agent.service
Synchronizing state of wazuh-agent.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service →
/lib/systemd/system/wazuh-agent.service.
root@pc034l:/var/ossec/etc# systemctl start wazuh-agent.service
```

De manera análoga, se realiza la instalación del agente en entornos Microsoft Windows desde una consola de PowerShell con permisos de administrador:

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.7.0-1.msi `
-OutFile ${env:tmp}\wazuh-agent.msi; `
msiexec.exe /i ${env:tmp}\wazuh-agent.msi /q `
WAZUH_MANAGER="manager.atalaia.local" `
WAZUH_REGISTRATION_PASSWORD="ag3nt-At" `
WAZUH_AGENT_GROUP="Windows" `
WAZUH_REGISTRATION_CA="atalaia-root-ca.crt" `
WAZUH_REGISTRATION_CERTIFICATE="pc033w.crt" `
WAZUH_REGISTRATION_KEY="pc033w.key"
```

Los ajustes aplicados se encuentran en la sección XML client del archivo C:\Program Files (x86)\ossec-agent\ossec.conf y la contraseña de registro en el fichero C:\Program Files (x86)\ossec-agent\auth.pass.

```
<ossec_config>
  ...
  <client>
    <server>
      <address>manager.atalaia.local</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>windows, windows10</config-profile>
    <crypto_method>aes</crypto_method>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <enrollment>
      <enabled>yes</enabled>
      <authorization_pass_path>authd.pass</authorization_pass_path>
      <server_ca_path>atalaia-root-ca.crt</server_ca_path>
      <agent_certificate_path>pc033w.crt</agent_certificate_path>
      <agent_key_path>pc033w.key</agent_key_path>
      <groups>Windows</groups>
    </enrollment>
  </client>
  ...
<ossec_config>
```

Fichero 4: ossec.conf (conexión de agente Microsoft Windows)

```
PS C:\Program Files (x86)\ossec-agent> cat .\authd.pass
ag3nt-At
```

Los certificados y clave privada indicados en el comando se deben copiar en esta ocasión en el directorio C:\Program Files (x86)\ossec-agent antes de iniciar el servicio del agente.

```
PS C:\Program Files (x86)\ossec-agent> Get-Service WazuhSvc

Status   Name               DisplayName
------   ----               -----------
Stopped  WazuhSvc           Wazuh

PS C:\Program Files (x86)\ossec-agent> Start-Service -Name WazuhSvc

PS C:\Program Files (x86)\ossec-agent> Get-Service WazuhSvc | Select-Object Name,
DisplayName, StartType, Status

Name     DisplayName StartType  Status
----     ----------- ---------  ------
WazuhSvc Wazuh       Automatic Running
```

4.6.3. Estado de los agentes

El estado de los agentes registrados es posible consultarlo a través del dashboard de Wazuh, la línea de comandos o la API del servicio Wazuh manager, y, como se puede observar, se encuentran en estado activo.

```
root@manager:/# /var/ossec/bin/agent_control -l

Wazuh agent_control. List of available agents:
   ID: 000, Name: manager (server), IP: 127.0.0.1, Active/Local
   ID: 001, Name: pc034l, IP: any, Active
   ID: 002, Name: pc033w, IP: any, Active

List of agentless devices:
```

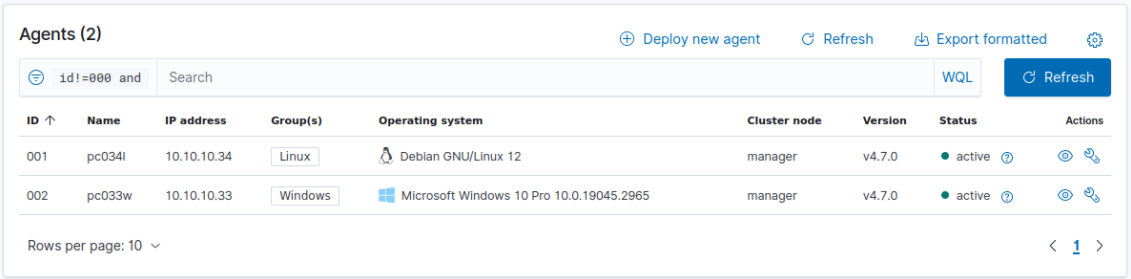


Figura 23: Estado de los agentes registrados en el dashboard

5. Casos de uso

Los casos de uso que se proponen a continuación con el fin de proteger dispositivos de punto final se basan en 6 ejes fundamentales para hacer frente a las Tácticas, Técnicas y Procedimientos empleados en ciberataques según el análisis de informes recientes [2,10,11].

Figura 24: Seguridad en endpoints

5.1. Funcionamiento como SIEM

Las soluciones de tipo SIEM permiten recolectar logs de distintas fuentes para su análisis y ofrecer visibilidad de amenazas en el entorno TIC de las organizaciones, siendo considerado por la CISA como una medida que ayuda a mitigar los riesgos asociados a la práctica totalidad de las tácticas utilizadas por los atacantes.

El gestor de Wazuh viene preconfigurado por defecto para reenviar únicamente las alertas generadas al indexador, pero es de utilidad registrar todos los eventos que reciba de los agentes para poder correlacionarlos con otras fuentes de información y, así, deducir patrones de comportamiento, observar cambios de tendencias o detectar amenazas [60]. Un aspecto fundamental que se debe tener en cuenta en soluciones SIEM es el espacio de almacenamiento necesario, debido al número de dispositivos totales, a la cantidad de eventos que genera cada uno de ellos y a la ventana temporal que se necesita disponer.

El componente OSSEC del gestor de Wazuh puede almacenar tanto las alertas generadas como los eventos que recibe en dos formatos diferentes, texto plano y JSON, siendo necesario este último para que el servicio Filebeat reenvíe los eventos al indexador con un nombre de índice definido en los siguientes ficheros [61].

```
root@manager:/# cat /usr/share/filebeat\
/module/wazuh/alerts/manifest.yml
module_version: 0.1

var:
  - name: paths
    default:
      - /var/ossec/logs/alerts/alerts.json
  - name: index_prefix
    default: wazuh-alerts-4.x-

input: config/alerts.yml

ingest_pipeline: ingest/pipeline.json
```

```
root@manager:/# cat /usr/share/filebeat\
/module/wazuh/archives/manifest.yml
module_version: 0.1

var:
  - name: paths
    default:
      - /var/ossec/logs/archives/archives.json
  - name: index_prefix
    default: wazuh-archives-4.x-

input: config/archives.yml

ingest_pipeline: ingest/pipeline.json
```

Fichero 5: manifest.yml (prefijo de los índices)

```
root@manager:/# cat /usr/share/filebeat/module/wazuh/alerts/ingest/pipeline.json
root@manager:/# cat /usr/share/filebeat/module/wazuh/archives/ingest/pipeline.json
{
  "description": "Wazuh events pipeline",
  "processors": [
    ...
    {
      "date_index_name": {
        "field": "timestamp",
        "date_rounding": "d",
        "index_name_prefix": "{{fields.index_prefix}}",
        "index_name_format": "yyyy.MM.dd",
        "ignore_failure": false
      }
    }
    ...
  ],
  "on_failure" : [{
    "drop" : { }
  }]
}
```

Fichero 6: pipeline.json (frecuencia de rotación y sufijo de los índices)

La propiedad index_prefix define el prefijo del nombre del índice, date_rounding establece la frecuencia con la que se crea, diaria (d), semanal (w) o mensual (M), e index_name_format configura el sufijo del nombre del índice.

El funcionamiento de OSSEC se fundamenta en la utilización de un conjunto de archivos de configuración denominados (rulesets), que se clasifican en dos grupos principales: los descodificadores (decoders), que extraen la información relevante de los eventos procedentes de los dispositivos finales, y las reglas (rules), que determinan las condiciones bajo las cuales se generan las alertas en aspectos tan variados como los ciberataques, presencia de malware, problemas de configuración o violaciones de las políticas de seguridad. Wazuh ofrece un conjunto de rulesets preconfiguradas y listas para usar que soportan una amplia variedad de tecnologías, entre las que se incluyen productos referentes en su segmento de mercado como FortiGate, Palo Alto, Cisco, Sophos o Office 365 [62].

El gestor de Wazuh permite crear descodificadores y reglas personalizadas para que se adapten a las necesidades del entorno TIC de la organización, debiéndose utilizar los números de identificadores comprendidos entre 100000 y 120000.

```
# Conjunto de reglas (ruleset) proporcionadas por Wazuh
/var/ossec/ruleset/decoders/
/var/ossec/ruleset/rules

# Ruta para crear descodificadores y reglas personalizadas
/var/ossec/etc/decoders/local_decoder.xml
/var/ossec/etc/rules/local_rules.xml
```

```
Nota: Los archivos predefinidos local_decoder.xml y local_rules.xml están pensados para realizar pequeños ajustes. Por ello, se recomienda crear archivos individuales para descodificadores y reglas cuando las personalizaciones se realizan a gran escala.
```

Los pasos necesarios para reenviar los eventos registrados al indexador desde el gestor de Wazuh son:

- Se habilita la opción logall_json en la sección global del fichero ossec.conf para que los eventos registrados se almacenen en formato JSON en el directorio /var/ossec/logs/archives/ [63].

```
root@manager:/# cat /var/ossec/etc/ossec.conf
<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>yes</logall_json>
    ...
  </global>
  ...
</ossec_config>
```

Fichero 7: ossec.conf (configuración global en el gestor de Wazuh)

- Se establece el reenvío de eventos al indexador mediante el archivo de configuración del servicio Filebeat (/etc/filebeat/filebeat.yml).

```
root@manager:/# cat /etc/filebeat/filebeat.yml
# Wazuh - Filebeat configuration file

filebeat.modules:
  - module: wazuh
    alerts:
      enabled: true
    archives:
      enabled: true
...
```

Fichero 8: filebeat.yml (reenvío de eventos al dashboard de Wazuh)

- Se reinicia el contenedor para aplicar los cambios realizados.

```
[boole@atl004s atalaia]$ docker compose restart wazuh-manager
[+] Restarting 1/1
 ✓ Container manager  Started                                    3.6s
```

- Se define el siguiente patrón de índice (index pattern) en la sección Stack Management del dashboard de Wazuh.

Opción de configuración	Valor
Nombre	wazuh-archives-*
Campo de tiempo	timestamp

Tabla 15: Opciones de configuración de los patrones de índice

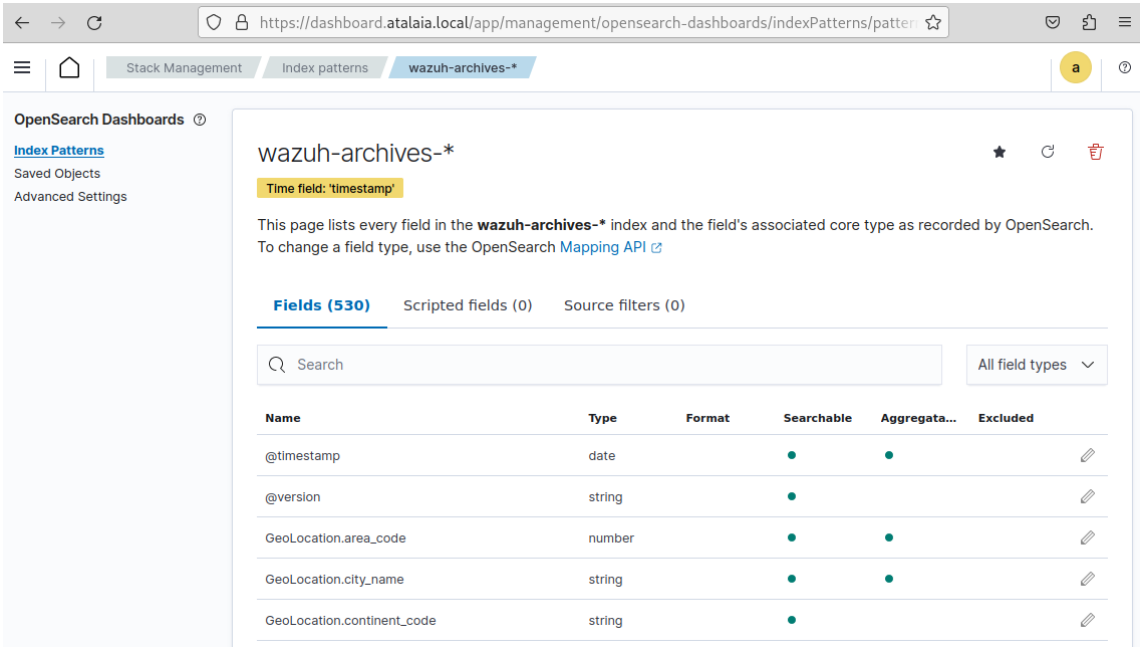


Figura 25: Creación de un patrón de índice a través del panel de control de Wazuh

- Se verifica que se están archivando los eventos recibidos en Wazuh manager.

```
root@manager:/# ls -lah /var/ossec/logs/archives/
total 4.0M
drwxr-x---. 3 wazuh wazuh 4.0K Dec  6 21:38 .
drwxrwx---. 8 wazuh wazuh 4.0K Dec  6 20:15 ..
drwxr-x---. 3 wazuh wazuh 4.0K Dec  6 20:15 2023
-rw-r-----. 2 wazuh wazuh 4.0M Dec  6 21:39 archives.json
-rw-r-----. 2 wazuh wazuh    0 Dec  6 20:15 archives.log
```

- Se comprueba que se visualizan los eventos reenviados al indexador con el nombre del índice correcto.

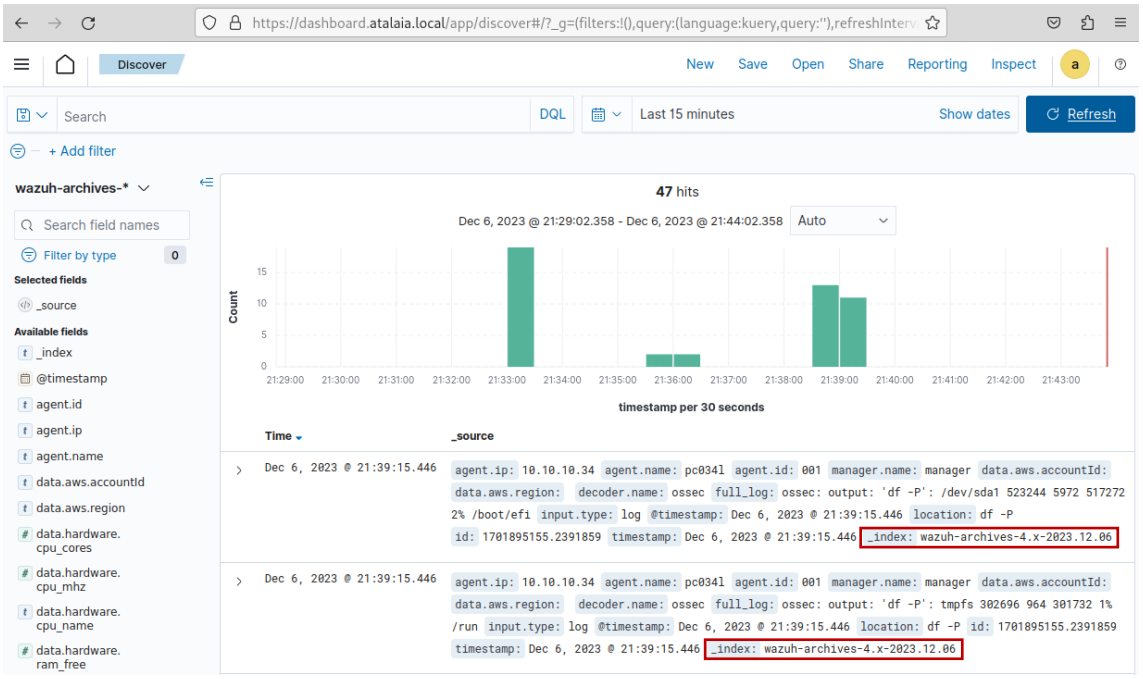


Figura 26: Visualización de eventos en el panel de control de Wazuh

La plataforma de Wazuh permite habilitar el servicio Syslog en su componente gestor para recoger eventos de dispositivos que no soportan la instalación de su agente [64].

```
root@manager:/# cat /var/ossec/etc/ossec.conf
<ossec_config>
  ...
  <!-- Syslog service -->
  <remote>
    <connection>syslog</connection>
    <port>514</port>
    <protocol>tcp</protocol>
    <allowed-ips>10.10.10.0/24</allowed-ips>
    <local_ip>10.10.10.4</local_ip>
  </remote>
  ...
</ossec_config>
```

Fichero 9: ossec.conf (servicio Syslog)

La configuración de Fichero 9 tiene efecto cuando se reinicia el servicio wazuh-manager, de tal forma que se habilita el puerto 514/tcp en la dirección IP 10.10.10.4 para recibir eventos de estilo syslog desde dispositivos de la subred indicada en la opción de configuración allowed-ips.

```
[boole@atl004s atalaia]$ docker compose restart wazuh-manager
[+] Restarting 1/1
 ✓ Container manager  Started                                                    3.6s

[boole@atl004s atalaia]$ ss -patnl |grep -w 514
LISTEN 0      128             10.10.10.4:514        0.0.0.0:*
```

El componente de panel de control de Wazuh cumple una doble función vital dentro de una solución SIEM con su conglomerado de funciones: por una parte, proporciona distintas herramientas que permiten consultar, visualizar, analizar y procesar la información, y, por otra, gestiona la plataforma.

5.2. Inventariado de los dispositivos finales

Los agentes de Wazuh utilizan el módulo Syscollector para recoger información de los dispositivos finales de manera periódica, entre la que se incluye el hardware y sistema operativo disponible, interfaces de red y su configuración, puertos de red abiertos, aplicaciones o paquetes instalados, procesos en ejecución y, para entornos Microsoft Windows, los parches aplicados (hotfixes) [65].

La configuración que se aplica por defecto a los agentes se encuentra en su fichero local ossec.conf y se traspasa a los archivos agent.conf en el gestor de Wazuh para disponer de una gestión centralizada [66].

```
root@pc034l:~# cat /var/ossec/etc/ossec.conf
PS C:\> more '.\Program Files (x86)\ossec-agent\ossec.conf'
```

```
<ossec_config>
  ...
  <!-- System inventory -->
  <wodle name="syscollector">
    <disabled>no</disabled>
    <interval>1h</interval>
    <scan_on_start>yes</scan_on_start>
    <hardware>yes</hardware>
    <os>yes</os>
    <network>yes</network>
```

```
root@manager:/# cat /var/ossec/etc/shared\
/Linux/agent.conf
root@manager:/# cat /var/ossec/etc/shared\
/Windows/agent.conf
```

```
<agent_config>
  <!-- System inventory -->
  <wodle name="syscollector">
    <disabled>no</disabled>
    <interval>1h</interval>
    <scan_on_start>yes</scan_on_start>
    <hardware>yes</hardware>
    <os>yes</os>
    <network>yes</network>
    <packages>yes</packages>
```

```
    <packages>yes</packages>
    <ports all="no">yes</ports>
    <processes>yes</processes>

    <!-- Database synchronization settings -->
    <synchronization>
      <max_eps>10</max_eps>
    </synchronization>
  </wodle>
  ...
</ossec_config>
```

```
    <ports all="yes">yes</ports>
    <processes>yes</processes>

    <!-- Database synchronization settings -->
    <synchronization>
      <max_eps>10</max_eps>
    </synchronization>
  </wodle>
</agent_config>
```

Fichero 10: ossec.conf (Inventariado de los agentes de Wazuh)
Fichero 11: agent.conf (Inventariado de los agentes de Wazuh)

Los cambios realizados en los archivos agent.conf se pueden verificar con el comando verify-agent-conf.

```
root@manager:~# /var/ossec/bin/verify-agent-conf -f /var/ossec/etc/shared/Linux/agent.conf
verify-agent-conf: OK
root@manager:~# /var/ossec/bin/verify-agent-conf -f /var/ossec/etc/shared/Windows/agent.conf
verify-agent-conf: OK
```

Los datos recopilados por el módulo Syscollector sirven tanto para conocer la superficie de ataque que suponen los dispositivos de punto final como para detectar la presencia de conexiones, procesos o programas asociados a artefactos maliciosos.

La opción ports dispone del parámetro all habilitado para obtener toda la información disponible de los puertos, de tal forma que no se restrinja a los que están únicamente en estado abierto y preparados para aceptar conexiones (listening). Además, no es necesario indicar explícitamente el ajuste de parches aplicados en entornos de Microsoft Windows a través de la opción XML hotfixes correspondiente, ya que por defecto está habilitada.

Toda la información recogida se almacena en bases de datos identificadas con el ID del agente y ubicadas en el directorio /var/ossec/queue/db/ del gestor de Wazuh.

```
[boole@atl004s atalaia]$ docker compose exec wazuh-manager /bin/bash
root@manager:/# /var/ossec/bin/agent_control -l

Wazuh agent_control. List of available agents:
   ID: 000, Name: manager (server), IP: 127.0.0.1, Active/Local
   ID: 001, Name: pc034l, IP: any, Active
   ID: 002, Name: pc033w, IP: any, Active
...
root@manager:/# ls -la /var/ossec/queue/db/00*.db
-rw-r-----. 1 wazuh wazuh  1466368 Dec  8 15:29 /var/ossec/queue/db/000.db
-rw-r-----. 1 wazuh wazuh  2539520 Dec  8 15:30 /var/ossec/queue/db/001.db
-rw-r-----. 1 wazuh wazuh 24432640 Dec  8 15:30 /var/ossec/queue/db/002.db
root@manager:/# sqlite3 /var/ossec/queue/db/001.db
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
sqlite> SELECT scan_time,name,version,checksum FROM sys_programs WHERE name="wazuh-agent";
2023/12/06 19:46:42|wazuh-agent|4.7.0-1|956b9ea911408e8bf57c4c6ecd85e5a6103ae47b
```

5.3. Detección de vulnerabilidades

El módulo de detección de vulnerabilidades, que no viene habilitado por defecto, depende de la recopilación de aplicaciones instaladas en los endpoints que realiza el módulo Syscollector para, así, poder verificar si son vulnerables a partir de la

información de una base de datos construida en el gestor de Wazuh con repositorios de CVE disponibles públicamente [67].

La configuración necesaria se realiza en la sección XML vulnerabiliy-detector del fichero /var/ossec/etc/ossec.conf del gestor de Wazuh [68].

```
root@manager:/# cat /var/ossec/etc/ossec.conf
<ossec_config>
  ...
  <vulnerability-detector>
    <enabled>yes</enabled>
    <interval>5m</interval>
    <min_full_scan_interval>6h</min_full_scan_interval>
    <run_on_start>yes</run_on_start>
    ...
    <!-- Debian OS vulnerabilities -->
    <provider name="debian">
      <enabled>yes</enabled>
      <os>buster</os>
      <os>bullseye</os>
      <os>bookworm</os>
      <update_interval>1h</update_interval>
    </provider>
    ...
    <!-- Windows OS vulnerabilities -->
    <provider name="msu">
      <enabled>yes</enabled>
      <update_interval>1h</update_interval>
    </provider>

    <!-- Aggregate vulnerabilities -->
    <provider name="nvd">
      <enabled>yes</enabled>
      <update_interval>1h</update_interval>
    </provider>

  </vulnerability-detector>
  ...
</ossec_config>
```

Fichero 12: ossec.conf (detector de vulnerabilidades)

Opción	Descripción
enabled	Permite habilitar el módulo con el valor yes.
interval	Se establece 5 minutos entre escaneos parciales de vulnerabilidades asociados a nuevos paquetes.
min_full_scan_interval	Se establece un tiempo mínimo de 6 horas para realizar el siguiente escaneo completo, siempre que se haya actualizado la base de datos de los CVE.
run_on_start	Se habilita para actualizar la base de datos de los CVE y el escaneo de vulnerabilidades cuando el servicio se inicia.
provider	Cada sección XML de tipo provider define las fuentes necesarias, predefinidas o personalizadas, para actualizar la base de datos de los CVE y la frecuencia con la que se consultan.
provider name="debian"	Proveedor de información de vulnerabilidades para la distribución de Debian GNU/Linux.
provider name="msu"	Proveedor de información de vulnerabilidades para Microsoft Windows
provider name="nvd"	Proveedor de información de vulnerabilidades para los SO soportados que se obtiene de la base de datos de vulnerabilidades nacional (NVD) del NIST. Debe de estar habilitado para que el detector de vulnerabilidades funcione correctamente.

Tabla 16: Opciones de configuración del detector de vulnerabilidades

Una vez reiniciado el servicio wazuh-manager se recarga la configuración, se actualiza la base de datos con los CVE y se realiza el primer escaneo completo de vulnerabilidades, cuyos resultados se visualizan en Figura 27 y Figura 28 a través del panel de control.

```
[boole@atl004s atalaia]$ docker compose restart wazuh-manager
[+] Restarting 1/1
 ✓ Container manager  Started                                                    3.6s
[boole@atl004s atalaia]$ docker compose exec wazuh-manager ls -lah \
/var/ossec/queue/vulnerabilities/cve.db
-rw-rw----. 1 root wazuh 895M Dec  8 14:29 /var/ossec/queue/vulnerabilities/cve.db
root@manager:/# sqlite3 /var/ossec/queue/vulnerabilities/cve.db
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
sqlite> .tables 'VULNERABILITIES%'
VULNERABILITIES       VULNERABILITIES_INFO
```

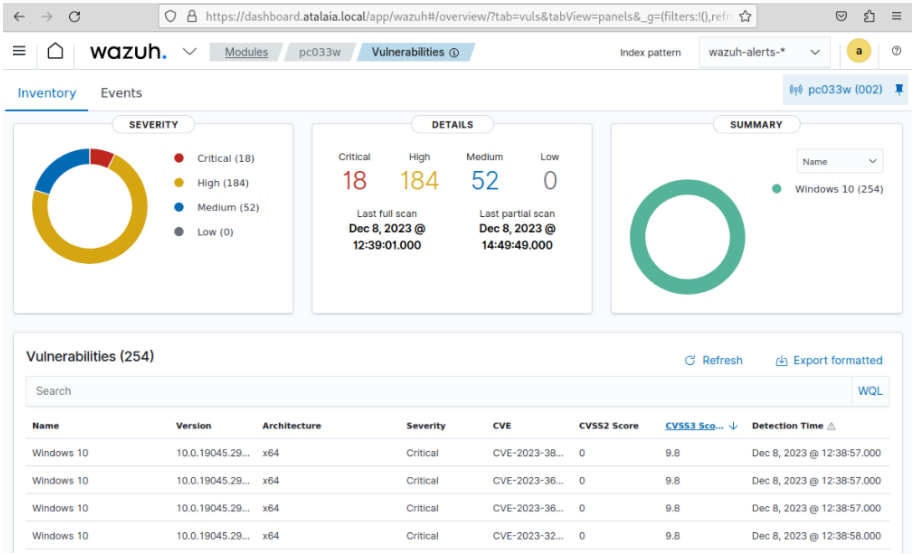


Figura 27: Vulnerabilidades detectadas en pc033w

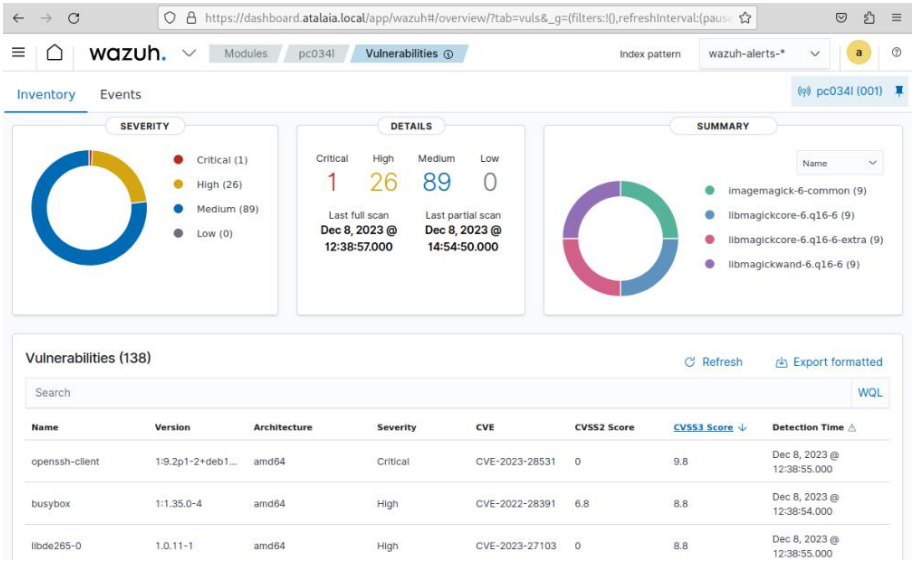


Figura 28: Vulnerabilidades detectadas en pc034l

La revisión de la vulnerabilidad CVE-2023-28531 de OpenSSH en la página web de seguimiento de fallos de seguridad de Debian GNU/Linux indica que es vulnerable para su version 12 (bookworm) [69], que es la disponible en el agente.

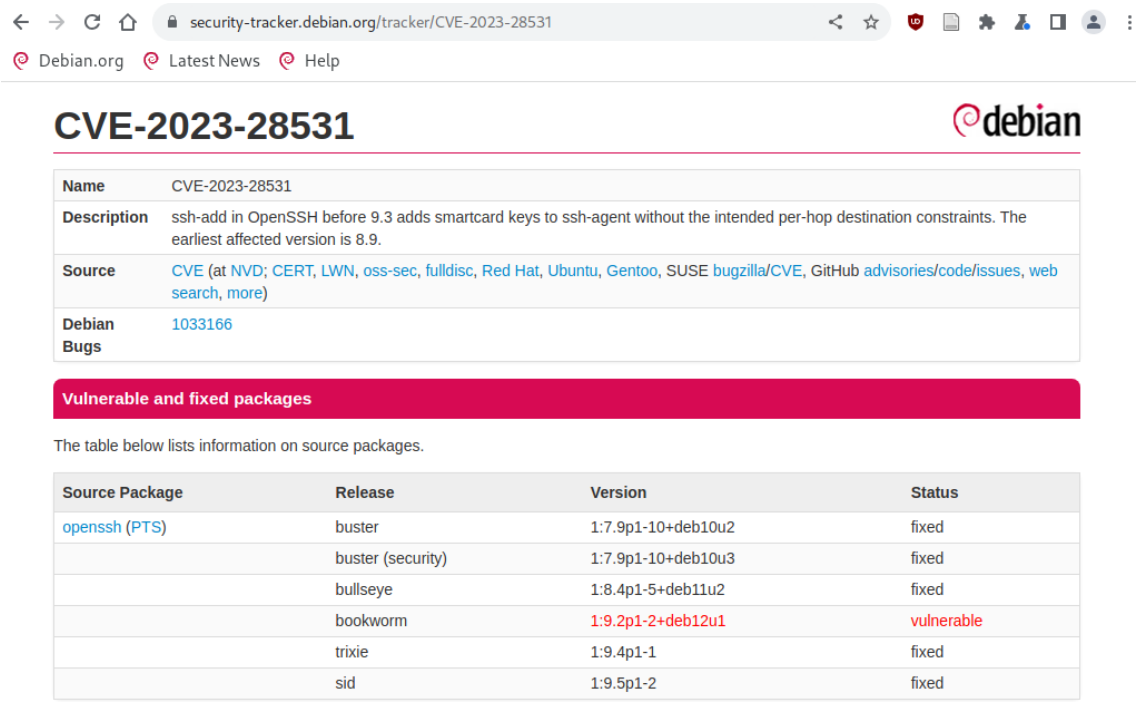


Figura 29: Vulnerabilidad CVE-2023-28531 [69]

5.4. Detección de artefactos maliciosos

La solución XDR de Wazuh permite combinar sus capacidades de detección de malware a través de sus módulos Rootcheck y FIM (monitorización de integridad de archivos) con las de otras soluciones de seguridad como los antivirus tradicionales (EPP), siendo necesario en este último caso la recolección de sus logs [70].

Módulo/Capacidad	Técnicas empleadas
Rootcheck	Basado en firmas de rootkits y troyanos Monitorización de anomalías
FIM	Reglas de detección de amenazas Integración con fuentes de inteligencia de amenazas
Recolección de logs	Descodificadores y reglas de serie para productos EPP como Sophos Antivirus, ClamAV, Symantec AV o Antivirus de Microsoft Defender [62].

Tabla 17: Módulos y capacidades para la detección de artefactos maliciosos

5.4.1. Módulo Rootcheck.

El funcionamiento del módulo Rootcheck tiene un doble enfoque para la detección de posible malware:

- Búsqueda de anomalías en procesos, puertos abiertos, interfaces de red, permisos y sistema de ficheros.
- Utilización de ficheros de texto con firmas de malware, aplicaciones no deseadas o configuraciones que se quieren auditar al ser modificadas cuando existe una infección o ciberataque.

En las versiones actuales de Wazuh, la auditoría del estado de seguridad pasa a realizarse desde su módulo propio denominado SCA, tal y como se detalla en la sección 5.6.

Un ejemplo de configuración centralizada para entornos GNU/Linux y Microsoft Windows se puede observar en Fichero 13 [71].

```
root@manager:/# cat \
/var/ossec/etc/shared/Linux/agent.conf
<agent_config>
  ...
  <!-- Rootcheck -->
  <rootcheck>
    <disabled>no</disabled>
    <check_files>yes</check_files>
    <check_trojans>yes</check_trojans>
    <check_dev>yes</check_dev>
    <check_sys>yes</check_sys>
    <check_pids>yes</check_pids>
    <check_ports>yes</check_ports>
    <check_if>yes</check_if>

    <!-- Frequency that rootcheck is executed
    every 12 hours -->
    <frequency>43200</frequency>

    <rootkit_files>
      etc/shared/Rootkit_files.txt
    </rootkit_files>
    <rootkit_trojans>
      etc/shared/Rootkit_trojans.txt
    </rootkit_trojans>

    <skip_nfs>yes</skip_nfs>
  </rootcheck>
  ...
</agent_config>
```

```
root@manager:/# cat \
/var/ossec/etc/shared/Windows/agent.conf
<agent_config>
  ...
  <!-- Rootcheck -->
  <rootcheck>
    <disabled>no</disabled>
    <check_files>yes</check_files>
    <check_trojans>yes</check_trojans>
    <check_dev>yes</check_dev>
    <check_sys>yes</check_sys>
    <check_pids>yes</check_pids>
    <check_ports>yes</check_ports>
    <check_if>yes</check_if>

    <!-- Frequency that rootcheck is executed
    every 12 hours -->
    <frequency>43200</frequency>

    <windows_apps>
      shared/Win_applications_rcl.txt
    </windows_apps>
    <windows_malware>
      shared/Win_malware_rcl.txt
    </windows_malware>
    <windows_audit>
      shared/Win_audit_rcl.txt
    </windows_audit>

    <skip_nfs>yes</skip_nfs>
  </rootcheck>
  ...
</agent_config>
```

Fichero 13: agent.conf (módulo rootcheck)

Opción	Descripción
disabled	Permite deshabilitar el módulo Rootcheck con el valor yes.
check_	Las opciones que comienzan con el prefijo check_ establecen las distintas opciones de búsqueda de anomalías y de malware que dispone el módulo Rootcheck. Están habilitadas por defecto si no se indican explícitamente y pueden deshabilitarse individualmente de no ser de interés.
frequency	Frecuencia con la que se ejecuta el módulo Rootcheck, que por defecto tiene un valor de 43200 segundos (12 horas).
rootkit_files **rootkit_trojans**	Establecen los ficheros de firmas de rootkits y troyanos.
windows_malware **windows_apps** **windows_audit**	Establecen los ficheros de firmas que se utilizan en entornos Microsoft Windows para detectar la presencia de malware, la existencia de ciertas aplicaciones o determinadas configuraciones en el sistema.
skip_nfs	Excluye la revisión de puntos de montaje CIFS y NFS cuando se habilita.

Tabla 18: Opciones de configuración del módulo Rootcheck

5.4.2. Módulo FIM

El módulo FIM permite monitorizar archivos de los directorios que se le indican y verificarlos con indicadores de compromiso (IOC) disponibles en listas CDB o en servicios externos como Virustotal. Viene preconfigurado a través de la sección XML syscheck del fichero local ossec.conf con unos valores diferentes en función de la versión del sistema operativo de los agentes o de si se trata del gestor de Wazuh.

Los ajustes personalizados pueden realizarse de manera centralizada en el gestor de Wazuh a través de los ficheros agent.conf ubicados en los directorios asociados a los grupos de agentes correspondientes [72].

```
root@manager:/# cat /var/ossec/etc/shared/Linux/agent.conf
<agent_config>
  ...
  <!-- File integrity monitoring -->
  <syscheck>
    <!-- Directories to check  (perform all possible verifications) -->
    <directories realtime="yes">/etc</directories>
  </syscheck>
  ...
</agent_config>
```

Fichero 14: agent.conf (configuración FIM para GNU/Linux)

```
root@manager:/# cat /var/ossec/etc/shared/Windows/agent.conf
<agent_config>
  ...
  <!-- File integrity monitoring -->
  <syscheck>
    <frequency>300</frequency>
    <windows_registry arch="both">
      HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
    </windows_registry>
    <windows_registry arch="both">
      HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
    </windows_registry>
  </syscheck>
  ...
</agent_config>
```

Fichero 15: agent.conf (configuración FIM para Microsoft Windows)

En Fichero 14 se monitoriza en tiempo real el directorio /etc de entornos GNU/Linux mientras que en Fichero 15 se hace lo mismo con las ramas de registro Run y RunOnce, que son utilizadas para definir las aplicaciones que se ejecutarán al iniciar sesión en entornos Windows.

```
root@manager:~# /var/ossec/bin/verify-agent-conf -f /var/ossec/etc/shared/Linux/agent.conf
2023/12/23 17:35:21 sca: WARNING: File 'etc/shared/cis_debian12.yml' not found.
verify-agent-conf: OK
root@manager:~# /var/ossec/bin/verify-agent-conf -f /var/ossec/etc/shared/Windows/agent.conf
verify-agent-conf: OK
```

Las listas CDB se crean a partir de ficheros de texto con el formato clave:valor, tal y como se puede ver a continuación.

- Se crea el archivo de texto malware-hashes que contiene el hash de la muestra de malware EICAR [73].

```
boole@pc034l:~/Descargas$ sha256sum eicar.com
275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f  eicar.com

root@manager:/# cat /var/ossec/etc/lists/malware-hashes
275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f:EICAR
```

- Al reiniciar wazuh-manager es cuando se genera la lista CDB malware-hashes.cdb que se utilizará.

```
[boole@atl004s atalaia]$ docker compose restart wazuh-manager
[+] Restarting 1/1
 ✓ Container manager  Started                                    3.6s
[boole@atl004s atalaia]$ docker compose exec wazuh-manager /bin/bash
root@manager:/# ls -la /var/ossec/etc/lists/malware-hashes*
-rw-r-----. 1 wazuh wazuh   71 Dec 22 17:09 /var/ossec/etc/lists/malware-hashes
-rw-rw----. 1 wazuh wazuh 2141 Dec 22 17:56 /var/ossec/etc/lists/malware-hashes.cdb
root@manager:~# file /var/ossec/etc/lists/malware-hashes
/var/ossec/etc/lists/malware-hashes: ASCII text
root@manager:~# file /var/ossec/etc/lists/malware-hashes.cdb
/var/ossec/etc/lists/malware-hashes.cdb: data
```

Finalmente, se define una regla personalizada en el gestor de Wazuh que genera una alerta cuando se añade o modifica un fichero monitorizado por el módulo FIM cuyo hash se encuentra en la lista CDB creada anteriormente.

```
root@manager:/# cat /var/ossec/etc/rules/local_rules_malware_cdb.xml
<group name="malware,">
  <rule id="100015" level="11">
    <!-- The if_sid tag references the built-in FIM rules -->
    <if_sid>554, 550</if_sid>
    <list field="sha256" lookup="match_key">etc/lists/malware-hashes</list>
    <description>File with known malware hash detected: $(file) - Owner: $(uname)</description>
    <mitre>
      <id>T1608.001</id>
      <id>T1204.002</id>
    </mitre>
    <group>
      syscheck,virus,pci_dss_5.1,pci_dss_5.2,pci_dss_11.4,gpg13_4.2,gdpr_IV_35.7.d,
      nist_800_53_SI.3,nist_800_53_SI.4,tsc_A1.2,tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,
    </group>
  </rule>
</group>
```

Fichero 16: local_rules_malware_cdb.xml

Opción	Descripción
rule	El bloque XML rule define la regla personalizada con ID 100015 y la clasifica con el nivel 11 [74].
if_sid	La regla personalizada se dispara cuando el módulo FIM detecta un nuevo archivo (regla con ID 554) o un cambio de integridad (regla con ID 550).
list	La opción list realiza la búsqueda de hashes sha256 en el campo clave de la lista CDB definida.
description	Descripción que se muestra cuando salta la regla.
mitre	Esta opción define las técnicas MITRE ATT&CK [10] utilizadas por los atacantes que encajan con la regla definida: T1608.001 hace referencia a la carga de malware y T1204.002 a la ejecución de archivo malicioso.
group	Se utilizan los grupos definidos en las reglas con ID 550 y 554 para categorizar las alertas.

Tabla 19: Opciones de configuración de la regla con ID 100015 [75]

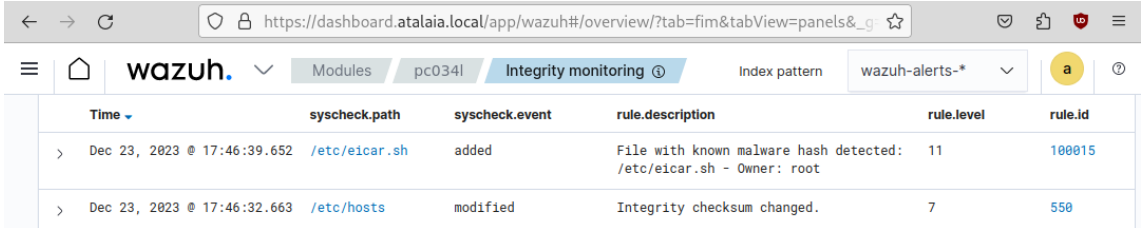


Figura 30: Alertas FIM en pc034l

Por otra parte, la modificación de la clave de registro Run del dispositivo de punto final pc033w hace saltar los eventos de aviso que aparecen en la Figura 31.

```
PS C:\> New-ItemProperty -Path "HKLM:\Software\Microsoft\Windows\CurrentVersion\Run" `
-Name "EICAR" -Value c:\eicar.com

EICAR        : c:\eicar.com
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion
PSChildName  : Run
PSDrive      : HKLM
PSProvider   : Microsoft.PowerShell.Core\Registry
```

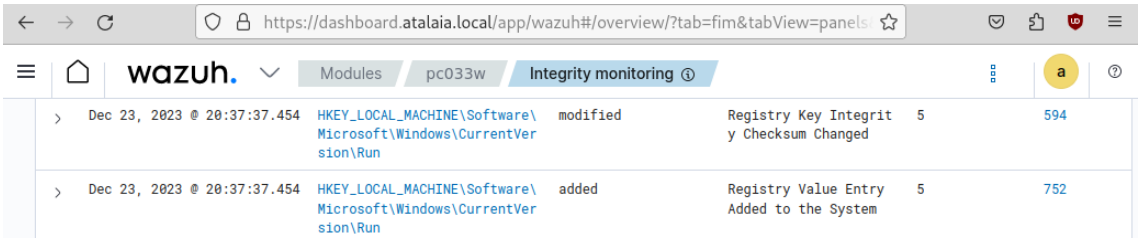


Figura 31: Alertas FIM en pc033w

5.4.3. Detección de malware con ClamAV.

ClamAV [76] es un antivirus de código abierto con licencia GPL que proporciona protección en tiempo real en entornos GNU/Linux y que viene incluido en los repositorios de Debian.

```
root@pc034l:~# apt install clamav clamav-daemon
root@pc034l:~# systemctl daemon-reload
root@pc034l:~# systemctl enable clamav-freshclam.service
Synchronizing state of clamav-freshclam.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable clamav-freshclam
Created symlink /etc/systemd/system/multi-user.target.wants/clamav-freshclam.service →
/lib/systemd/system/clamav-freshclam.service.
root@pc034l:~# systemctl start clamav-freshclam.service
root@pc034l:~# systemctl start clamav-daemon.service
```

Los eventos asociados con ClamAV, incluidas sus detecciones de malware, se registran por defecto en el archivo /var/log/clamav/clamav.log cuando, por ejemplo, se ejecuta un escaneo con el siguiente comando.

```
boole@pc034l:~$ clamdscan /tmp/eicar.com
/tmp/eicar.com: Win.Test.EICAR_HDB-1 FOUND

----------- SCAN SUMMARY -----------
Infected files: 1
Time: 0.002 sec (0 m 0 s)
Start Date: 2023:12:10 21:23:45
End Date:   2023:12:10 21:23:45

boole@pc034l:~$ grep -i "21:23:45" /var/log/clamav/clamav.log
Sun Dec 10 21:23:45 2023 -> /tmp/eicar.com: Win.Test.EICAR_HDB-
1(44d88612fea8a8f36de82e1278abb02f:68) FOUND
```

Sin embargo, la configuración lista para usar que proporciona Wazuh se basa en ficheros de registro basados en servicios Syslog, que en la actualidad no suelen venir instalados por defecto porque el nuevo sistema de arranque de los SO GNU/Linux, Systemd, incluye su propia implementación de diario de eventos a través del demonio

journald. Por este motivo, se procede a la creación de un descodificador y regla personalizados para hacer saltar alertas basadas en detecciones de malware a través del ClamAV, teniendo en cuenta las tres fases que se realizan durante el análisis de eventos: pre-descodificación, descodificación y búsqueda de correspondencia con las reglas definidas [75,77].

- Se crea el descodificador que incluye las dos primeras fases del proceso de revisión de logs que realiza Wazuh: pre-descodificación, para obtener una correspondencia previa del evento con la opción prematch, y descodificación, para obtener los campos de información de fecha, archivo malicioso junto a su hash y nombre del malware. Los grupos de paréntesis de la expresión regular regex permiten extraer la información de interés y asociarla a los campos definidos en la opción order.

```
root@manager:~# cat /var/ossec/etc/decoders/local_decoder_clamd.xml
<decoder name="clamd-custom">
  <!-- Malware detection -->
  <prematch>^\w+\s\w+\s\d+\s\d+:\d+:\d+\s\d+\s->\.*FOUND$</prematch>
  <regex>^(\w+\s\w+\s\d+\s\d+:\d+:\d+\s\d+)\s->\s(\.+):\s(\.+)\((\S+):\d+\)\sFOUND$</regex>
  <order>date,file,malware,hash</order>
</decoder>
```

Fichero 17: local_decoder_clamd.xml

- Se crean dos reglas personalizadas basadas en el descodificador anterior: una con ID 100010, que hace saltar una alerta cuando ClamAV detecta malware, y otra con ID 100011, que genera un aviso cuando se producen cinco detecciones para el mismo malware (hash) en las últimas 24 horas (86400 segundos). Los mensajes de las alertas incluyen el nombre del malware para informar de la amenaza de que se trata.

```
root@manager:~# cat /var/ossec/etc/rules/local_rules_clamd.xml
<group name="clamav,">
  <rule id="100010" level="8">
    <decoded_as>clamd-custom</decoded_as>
    <description>ClamAV: Virus $(malware) detected</description>
    <group>
      virus,pci_dss_5.1,pci_dss_5.2,pci_dss_11.4,gpg13_4.2,gdpr_IV_35.7.d,nist_800_53_SI.3,
      nist_800_53_SI.4,tsc_A1.2,tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,
    </group>
  </rule>

  <rule id="100011" level="10" frequency="5" timeframe="86400">
    <if_matched_sid>100010</if_matched_sid>
    <same_field>hash</same_field>
    <description>ClamAV: Virus $(malware) detected multiple times</description>
    <group>
      virus,pci_dss_5.1,pci_dss_5.2,pci_dss_11.4,gpg13_4.2,gdpr_IV_35.7.d,nist_800_53_SI.3,
      nist_800_53_SI.4,tsc_A1.2,tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,
    </group>
  </rule>
</group>
```

Fichero 18: local_rules_clamd.xml

- Se prueba el descodificador y las reglas creadas a través del siguiente comando CLI, aunque también es posible realizarlo vía API o a través de la herramienta Ruleset Test incluida en el panel de control de Wazuh.

```
root@manager:/# /var/ossec/bin/wazuh-logtest
Starting wazuh-logtest v4.7.0
Type one log per line

Sun Dec 10 21:23:45 2023 -> /tmp/eicar.com: Win.Test.EICAR_HDB-
1(44d88612fea8a8f36de82e1278abb02f:68) FOUND
```

```
**Phase 1: Completed pre-decoding.
       full event: 'Sun Dec 10 21:23:45 2023 -> /tmp/eicar.com: Win.Test.EICAR_HDB-
1(44d88612fea8a8f36de82e1278abb02f:68) FOUND'

**Phase 2: Completed decoding.
       name: 'clamd-custom'
       date: 'Sun Dec 10 21:23:45 2023'
       file: '/tmp/eicar.com'
       hash: '44d88612fea8a8f36de82e1278abb02f'
       malware: 'Win.Test.EICAR_HDB-1'

**Phase 3: Completed filtering (rules).
       id: '100010'
       level: '8'
       description: 'ClamAV: Virus detected'
       groups: '['clamav', 'virus']'
       firedtimes: '1'
       gdpr: '['IV_35.7.d']'
       gpg13: '['4.2']'
       mail: 'False'
       nist_800_53: '['SI.3', 'SI.4']'
       pci_dss: '['5.1', '5.2', '11.4']'
       tsc: '['A1.2', 'CC6.1', 'CC6.8', 'CC7.2', 'CC7.3']'
**Alert to be generated.
```

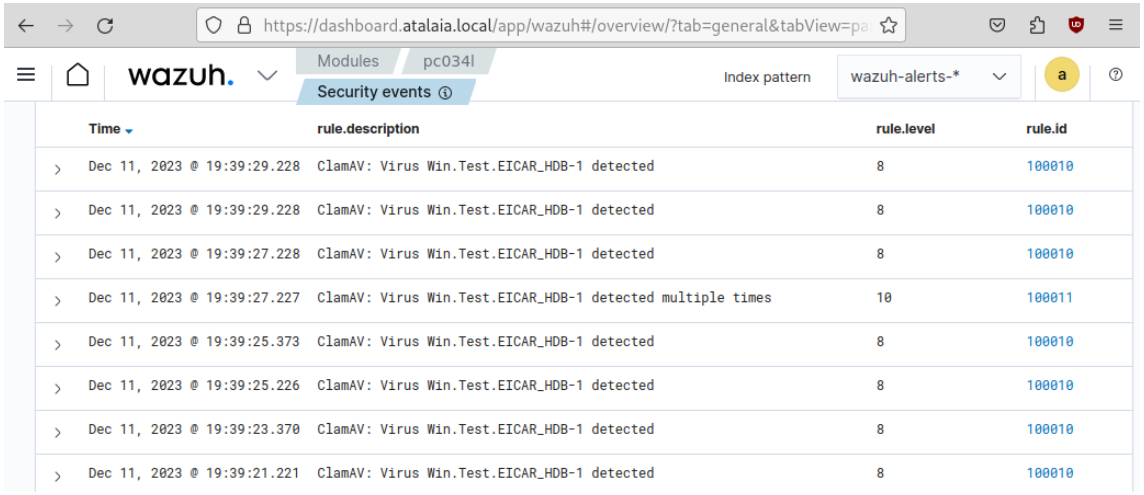
- Es necesario reiniciar Wazuh manager para que se apliquen las nuevas configuraciones realizadas.

```
boole@atl004s atalaia]$ docker compose restart wazuh-manager
[+] Restarting 1/1
 ✓ Container manager  Started                                                3.6s
```

- Finalmente, se realiza una configuración centralizada para los agentes del grupo Linux, de tal forma que se acceda a los eventos de su fichero /var/log/clamav/clamav.log. La opción only-future-events se deshabilita con el fin de obtener eventos antiguos no recogidos por el servicio wazuh-logcollector, siempre que no excedan el tamaño por defecto de 10MB, valor que se puede incrementar hasta 2GB con el atributo max-size.

```
root@manager:/# cat /var/ossec/etc/shared/Linux/agent.conf
<agent_config>

  <!-- Shared agent configuration here -->

  ...

  <localfile>
    <location>/var/log/clamav/clamav.log</location>
    <log_format>syslog</log_format>
    <only-future-events>no</only-future-events>
  </localfile>

  ...

</agent_config>
```

Fichero 19: agent.conf (recolección de logs de ClamAV)



Time	rule.description	rule.level	rule.id
Dec 11, 2023 @ 19:39:29.228	ClamAV: Virus Win.Test.EICAR_HDB-1 detected	8	100010
Dec 11, 2023 @ 19:39:29.228	ClamAV: Virus Win.Test.EICAR_HDB-1 detected	8	100010
Dec 11, 2023 @ 19:39:27.228	ClamAV: Virus Win.Test.EICAR_HDB-1 detected	8	100010
Dec 11, 2023 @ 19:39:27.227	ClamAV: Virus Win.Test.EICAR_HDB-1 detected multiple times	10	100011
Dec 11, 2023 @ 19:39:25.373	ClamAV: Virus Win.Test.EICAR_HDB-1 detected	8	100010
Dec 11, 2023 @ 19:39:25.226	ClamAV: Virus Win.Test.EICAR_HDB-1 detected	8	100010
Dec 11, 2023 @ 19:39:23.370	ClamAV: Virus Win.Test.EICAR_HDB-1 detected	8	100010
Dec 11, 2023 @ 19:39:21.221	ClamAV: Virus Win.Test.EICAR_HDB-1 detected	8	100010

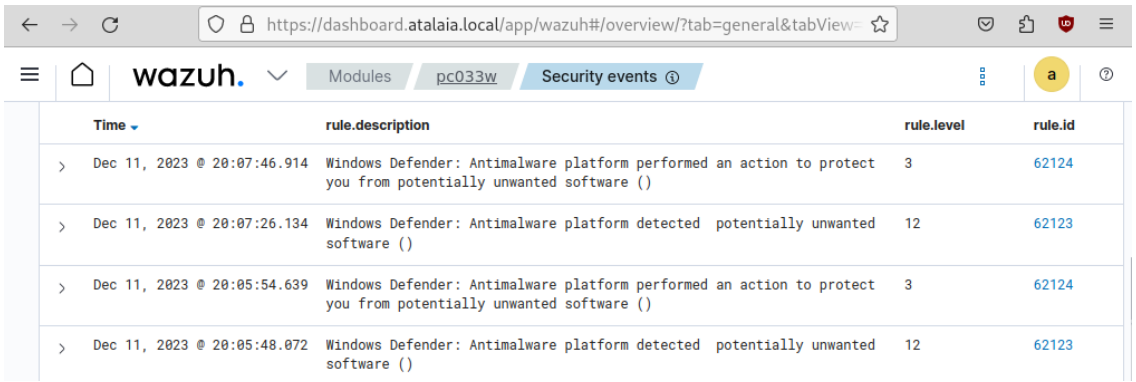
Figura 32: Detecciones de malware con ClamAV

5.4.4. Detecciones de malware con Antivirus de Microsoft Defender.

Las reglas para el Antivirus de Windows Defender ya vienen preconfiguradas y listas para funcionar, por lo que sólo es necesario configurar de manera centralizada el canal de eventos de Microsoft Defender para los agentes del grupo Windows, donde también se ha deshabilitado la opción only-future-events.

```
root@manager:/# cat /var/ossec/etc/shared/Windows/agent.conf
<agent_config>

  <!-- Shared agent configuration here -->

  ...

  <!-- Windows Defender logs -->
  <localfile>
    <location>Microsoft-Windows-Windows Defender/Operational</location>
    <log_format>eventchannel</log_format>
    <only-future-events>no</only-future-events>
  </localfile>

  ...
</agent_config>
```

Fichero 20: agent.conf (recogida de eventos de Microsoft Windows Defender)



Time	rule.description	rule.level	rule.id
Dec 11, 2023 @ 20:07:46.914	Windows Defender: Antimalware platform performed an action to protect you from potentially unwanted software ()	3	62124
Dec 11, 2023 @ 20:07:26.134	Windows Defender: Antimalware platform detected potentially unwanted software ()	12	62123
Dec 11, 2023 @ 20:05:54.639	Windows Defender: Antimalware platform performed an action to protect you from potentially unwanted software ()	3	62124
Dec 11, 2023 @ 20:05:48.072	Windows Defender: Antimalware platform detected potentially unwanted software ()	12	62123

Figura 33: Detecciones de malware con Antivirus de Microsoft Defender

5.5. Respuesta activa frente amenazas

La configuración por defecto de la plataforma de Wazuh incluye reglas que permiten detectar evidencias de actividad maliciosa o comportamientos anómalos, que deben ser

tenidos en cuenta y tratados adecuadamente para ofrecer una respuesta adecuada ante una posible amenaza. Por ejemplo, reglas que suponen un indicio de ataque de fuerza bruta a los servicios SSH y RDP, aunque sea de manera accidental, son las que se muestran en la Tabla 20.

ID	Clase de evento	Descripción
5712	ssh (Secure Shell)	sshd: brute force trying to get access to the system. Non existent user.
5758	ssh (Secure Shell)	Maximum authentication attempts exceeded.
60204	Eventos de Windows	Multiple Windows logon failures.

Tabla 20: Reglas con errores de inicio de sesión en los servicios SSH y RDP

5.5.1. Configuración en el gestor Wazuh

El descodificador que viene configurado por defecto en la plataforma para detectar cuando se alcanza el número máximo de intentos da autenticación SSH permitidos no extrae la información necesaria cuando se trata de usuarios inválidos en el sistema, por lo que se deshabilita y se crean dos nuevos siguiendo el mismo esquema de configuración (Fichero 22). De esta manera se dispondrá de la dirección IP desde la que se realizaron los intentos de validación, independientemente de si es una cuenta existente o no del sistema.

```
# Número máximo de intentos de autenticación alcanzado para usuario válido en el sistema
2023-12-17T21:06:24.106045+01:00 pc034l sshd[5535]: error: maximum authentication attempts
exceeded for boole from 10.10.10.35 port 41366 ssh2 [preauth]

# Número máximo de intentos de autenticación alcanzado para usuario inválido en el sistema
2023-12-17T21:16:02.995852+01:00 pc034l sshd[5753]: error: maximum authentication attempts
exceeded for invalid user baduser from 10.10.10.35 port 54238 ssh2 [preauth]
```

```
root@manager:/# cat /var/ossec/ruleset/decoders/0310-ssh_decoders.xml
<decoder name="sshd-exceed">
  <parent>sshd</parent>
  <prematch> exceeded for </prematch>
  <regex offset="after_prematch">^(\S+) from (\S+) port (\d+) </regex>
  <order>user, srcip, srcport</order>
</decoder>
```

Fichero 21: 0310-ssh_decoders.xml

```
root@manager:/# cat /var/ossec/etc/decoders/local_decoder_ssh.xml
<!-- Maximum authentication attempts exceeded for invalid user -->
<decoder name="sshd-exceed-invalid-user">
  <parent>sshd</parent>
  <prematch> exceeded for invalid user </prematch>
  <regex offset="after_prematch">^(\S+) from (\S+) port (\d+) </regex>
  <order>user, srcip, srcport</order>
</decoder>

<!-- Maximum authentication attempts exceeded for existent user -->
<decoder name="sshd-exceed">
  <parent>sshd</parent>
  <prematch> exceeded for </prematch>
  <regex offset="after_prematch">^(\S+) from (\S+) port (\d+) </regex>
  <order>user, srcip, srcport</order>
</decoder>
```

Fichero 22: local_decoder_ssh.xml

La regla personalizada para el servicio SSH creada en el Fichero 23, que activará la respuesta activa en el caso de que se salte tres veces la regla con ID 5758 en las últimas

24 horas (86400 segundos), permitirá aplicar bloqueos más restrictivos al presentar el mismo comportamiento el número de veces estipulado.

```
root@manager:/# cat /var/ossec/etc/rules/local_rules_ssh.xml
<group name="syslog,sshd,">
  <rule id="100012" level="10" frequency="3" timeframe="86400">
    <if_matched_sid>5758</if_matched_sid>
    <same_source_ip />
    <description>Repeated maximum authentication attempts from $(srcip).</description>
    <group>authentication_failed,gpg13_7.1,</group>
  </rule>
</group>
```

Fichero 23: local_rules_ssh.xml

La definición de la respuesta activa elegida se realiza en su totalidad en el gestor de Wazuh, aunque existe la opción de configuración repeated_offenders que, si se usa, debe ser aplicada en el fichero ossec.conf local de los agentes [78].

```
root@manager:/# cat /var/ossec/etc/ossec.conf
<ossec_config>

  <command>
    <name>firewalld-drop</name>
    <executable>firewalld-drop</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>

  <!-- Active response -->
  <active-response>
    <disabled>no</disabled>
    <command>firewalld-drop</command>
    <location>local</location>
    <rules_id>5712,5758</rules_id>
    <timeout>120</timeout>
  </active-response>

  <active-response>
    <disabled>no</disabled>
    <command>firewalld-drop</command>
    <location>local</location>
    <rules_id>100012</rules_id>
    <timeout>240</timeout>
  </active-response>

  <active-response>
    <disabled>no</disabled>
    <command>netsh</command>
    <location>local</location>
    <rules_id>60204</rules_id>
    <timeout>300</timeout>
  </active-response>

</ossec_config>
```

Fichero 24: ossec.conf (respuesta activa)

Opción	Descripción
command	Bloque XML que define el comando firewalld-drop para que esté disponible para la respuesta activa.
name	Nombre del comando
executable	Nombre del binario del comando que se encuentra en el directorio /var/ossec/active-response/bin/.
timeout_allowed	Especifica que el comando puede revertir sus acciones si se habilita.

Opción	Descripción
active-response	Bloque XML que establece la respuesta activa deseada.
disabled	Se establece a yes si se desea deshabilitar la respuesta activa.
command	Nombre del comando que se definió en el bloque XML command.
location	Establece en donde se ejecuta el comando de la respuesta activa, en el que el valor local indica que se hará en los agentes que han generado la alerta.
rules_id	Establece los identificadores de reglas que harán saltar la respuesta activa.
timeout	Establece el número de segundos que se mantendrá la acción de la respuesta activa.
repeated_offenders	Permite establecer hasta cinco valores de tiempo de espera ordenados de manera creciente para direcciones IP reincidentes. Esta opción de configuración solamente se puede definir en el fichero de configuración local ossec.conf del agente y actualmente no es soportada por Microsoft Windows.

Tabla 21: Opciones de configuración para la respuesta activa

Además, se debe realizar una configuración centralizada para los agentes del grupo Linux con el fin de que se recojan sus eventos de autenticación, que se encuentran en el fichero /var/log/auth.conf una vez se ha instalado el paquete rsyslog.

```
root@manager:/# cat /var/ossec/etc/shared/Linux/agent.conf
<agent_config>
  ...
  <localfile>
    <location>/var/log/auth.log</location>
    <log_format>syslog</log_format>
    <only-future-events>no</only-future-events>
  </localfile>
  ...
</agent_config>
```

Fichero 25: agent.conf (recogida de eventos de autenticación en GNU/Linux)

Finalmente, los cambios no tendrán efecto hasta que se reinicia el contenedor wazuh-manager.

```
[boole@atl004s atalaia]$ docker compose restart wazuh-manager
[+] Restarting 1/1
 ✓ Container manager  Started                                    3.6s
```

5.5.2. Simulación de ataque de fuerza bruta al servicio SSH

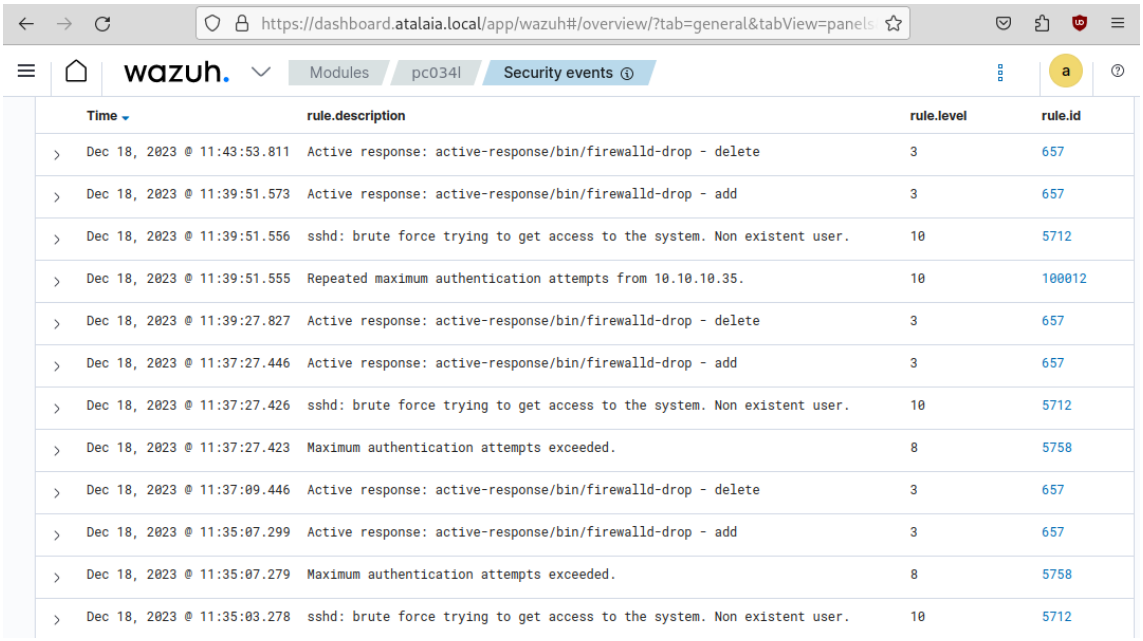
La simulación se ataque de fuerza bruta se puede realizar con la herramienta Hydra [79], crackeador de contraseñas que soporta una amplia variedad de protocolos y permite realizar ataques basados en diccionarios de palabras para el nombre de usuario y para las contraseñas.

```
$ hydra -t 1 -l boole -P /usr/share/wordlists/nmap.lst ssh://10.10.10.34
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or
secret (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-17 19:29:52
[DATA] max 1 task per 1 server, overall 1 task, 5007 login tries (l:1/p:5007), ~5007 tries
per task
[DATA] attacking ssh://10.10.10.34:22/
[STATUS] 6.00 tries/min, 6 tries in 00:01h, 5001 to do in 13:54h, 1 active
```

La activación de la regla con ID 5758, Maximum authentication attempts exceeded, debido a los errores de autenticación provoca la respuesta activa definida de bloquear la dirección IP atacante con el comando firewalld-drop.

```
root@pc034l:~# firewall-cmd --list-all |grep -i "rule family"
        rule family="ipv4" source address="10.10.10.35" drop
```

La duración de cada bloqueo viene determinada por la opción timeout de la respuesta activa y, tal y como se puede ver en la Figura 34, es de 2 minutos (120 segundos) para cuando salta la regla con ID 5758 y de 4 minutos (240 segundos) cuando se activa la regla personalizada con ID 100012.



Time	rule.description	rule.level	rule.id
Dec 18, 2023 @ 11:43:53.811	Active response: active-response/bin/firewalld-drop - delete	3	657
Dec 18, 2023 @ 11:39:51.573	Active response: active-response/bin/firewalld-drop - add	3	657
Dec 18, 2023 @ 11:39:51.556	sshd: brute force trying to get access to the system. Non existent user.	10	5712
Dec 18, 2023 @ 11:39:51.555	Repeated maximum authentication attempts from 10.10.10.35.	10	100012
Dec 18, 2023 @ 11:39:27.827	Active response: active-response/bin/firewalld-drop - delete	3	657
Dec 18, 2023 @ 11:37:27.446	Active response: active-response/bin/firewalld-drop - add	3	657
Dec 18, 2023 @ 11:37:27.426	sshd: brute force trying to get access to the system. Non existent user.	10	5712
Dec 18, 2023 @ 11:37:27.423	Maximum authentication attempts exceeded.	8	5758
Dec 18, 2023 @ 11:37:09.446	Active response: active-response/bin/firewalld-drop - delete	3	657
Dec 18, 2023 @ 11:35:07.299	Active response: active-response/bin/firewalld-drop - add	3	657
Dec 18, 2023 @ 11:35:07.279	Maximum authentication attempts exceeded.	8	5758
Dec 18, 2023 @ 11:35:03.278	sshd: brute force trying to get access to the system. Non existent user.	10	5712

Figura 34: Eventos de ataque de fuerza bruta al servicio SSH

5.5.3. Simulación de ataque de fuerza bruta al servicio RDP

De manera análoga al servicio SSH, se simula un ataque de fuerza bruta contra el servicio RDP de la máquina virtual pc033w con la herramienta Hydra.

```
$ hydra -t 1 -l baduser -P /usr/share/wordlists/nmap.lst rdp://10.10.10.33
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or
secret service organizations, or for illegal purposes (this is non-binding, these ***
ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-18 15:42:09
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 1 task per 1 server, overall 1 task, 5007 login tries (l:1/p:5007), ~5007 tries
per task
[DATA] attacking rdp://10.10.10.33:3389/
```

En este caso, la regla con ID 60204 se dispara por defecto cuando se producen ocho eventos de errores de autenticación en 240 segundos (4 minutos) desde la misma dirección IP de origen, lo que provoca su bloqueo durante cinco minutos según está definido en la repuesta activa asociada.

```
PS C:\> Get-NetFirewallRule -DisplayName "WAZUH ACTIVE RESPONSE BLOCKED IP" | Get-
NetFirewallAddressFilter

LocalAddress  : Any
RemoteAddress : 10.10.10.35
```

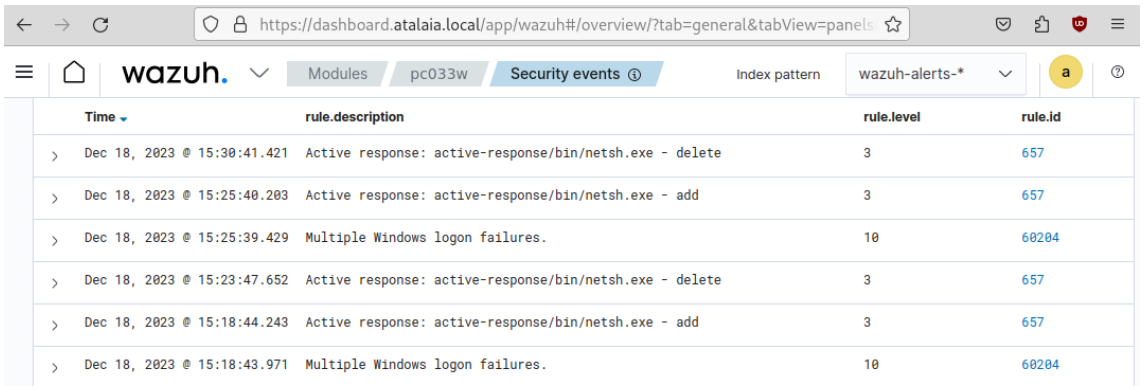


Figura 35: Eventos de ataque de fuerza bruta al servicio RDP

5.6. Evaluación del estado de seguridad

Wazuh añade por defecto políticas de evaluación del estado de la seguridad (SCA) [80] que sean compatibles con el SO en donde se instala el agente y, en el diseño propuesto, sólo ocurre para el SO Windows 10 al no existir una configuración de serie válida para Debian GNU/Linux 12.

```
root@pc034l:~# ls -la /var/ossec/ruleset/sca/
total 8
drwxr-x--- 2 root wazuh 4096 dic  6 20:39 .
drwxr-x--- 3 root wazuh 4096 dic  6 20:39 ..

PS C:\> dir 'C:\Program Files (x86)\ossec-agent\ruleset\sca\'

    Directorio: C:\Program Files (x86)\ossec-agent\ruleset\sca


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        23/11/2023     12:17         678759 cis_win10_enterprise.yml
```

Se aprovecha la circunstancia para realizar una configuración centralizada de la política SCA que se aplicará a los agentes con Debian GNU/Linux 12.

- Se descarga la política SCA para Debian GNU/Linux 12 proporcionada por Wazuh [81] y se pone a disposición de los agentes del grupo Linux con los permisos adecuados.

```
[boole@atl004s atalaia]$ docker compose cp ./config/wazuh_manager/cis_debian12.yml wazuh-
manager:/var/ossec/etc/shared/Linux/
[+] Copying 1/0
 ✓ manager copy ./config/wazuh_manager/cis_debian12.yml to
manager:/var/ossec/etc/shared/Linux/ Copied                                    0.0s


[boole@atl004s atalaia]$ docker compose exec wazuh-manager /bin/bash

root@manager:/# chown wazuh:wazuh /var/ossec/etc/shared/Linux/cis_debian12.yml
root@manager:/# chmod 640 /var/ossec/etc/shared/Linux/cis_debian12.yml
root@manager:/# ls -la /var/ossec/etc/shared/Linux/cis_debian12.yml
-rw-r-----. 1 wazuh wazuh 342660 Dec 18 17:19 /var/ossec/etc/shared/Linux/cis_debian12.yml
```

- Se establece la política SCA que se utilizará para evaluar la seguridad con el parámetro enabled para facilitar su activación y desactivación [82].

```
root@manager:/# cat /var/ossec/etc/shared/Linux/agent.conf
<agent_config>
  ...
  <!-- SCA policies -->
  <sca>
    <policies>
        <policy enabled="yes">etc/shared/cis_debian12.yml</policy>
    </policies>
  </sca>
  ...
</agent_config>
```

- Se verifica que se ha sincronizado la información con el agente pc034l, que tiene el identificador 001.

```
root@manager:/# /var/ossec/bin/agent_groups -i 001 -S
Agent '001' is synchronized.

root@manager:/# ls -la /var/ossec/etc/shared/Linux/
total 688
drwx------. 2 wazuh wazuh   4096 Dec 18 17:40 .
drwxrwx---. 5 root  wazuh   4096 Dec  6 20:35 ..
-rw-rw----. 1 wazuh wazuh   1012 Dec 18 17:40 agent.conf
-rw-r-----. 1 wazuh wazuh 342660 Dec 18 17:19 cis_debian12.yml
-rw-r--r--. 1 wazuh wazuh 344098 Dec 18 17:40 merged.mg

root@pc034l:/# ls -la /var/ossec/etc/shared/
total 692
drwxrwx--- 2 root  wazuh   4096 dic 18 17:40 .
drwxrwx--- 3 wazuh wazuh   4096 dic  6 20:44 ..
-rw------- 1 wazuh wazuh   1012 dic 18 17:40 agent.conf
-rw------- 1 wazuh wazuh    364 dic 18 17:40 ar.conf
-rw------- 1 wazuh wazuh 342660 dic 18 17:40 cis_debian12.yml
-rw-r--r-- 1 wazuh wazuh 344098 dic 18 17:40 merged.mg
```

- Se habilita la directiva sca.remote_commands para permitir la ejecución de los comandos definidos en la política SCA proporcionada de manera centralizada, que no tendrá efecto hasta que se reinicie el servicio wazuh-agent.

```
root@pc034l:~# echo "sca.remote_commands=1" >> \
/var/ossec/etc/local_internal_options.conf
root@pc034l:~# cat /var/ossec/etc/local_internal_options.conf
# local_internal_options.conf
#
# This file should be handled with care. It contains
# run time modifications that can affect the use
# of OSSEC. Only change it if you know what you
# are doing. Look first at ossec.conf
# for most of the things you want to change.
#
# This file will not be overwritten during upgrades.
sca.remote_commands=1
```

Fichero 26: local_internal_options.conf

```
root@pc034l:~# systemctl restart wazuh-agent.service
root@pc034l:~# systemctl status wazuh-agent.service
● wazuh-agent.service - Wazuh agent
     Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; preset: enabled)
     Active: active (running) since Tue 2023-12-18 17:42:50 CET; 55s ago
    Process: 5328 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited,
status=0/SUCCESS)
      Tasks: 32 (limit: 4648)
     Memory: 218.0M
```

```
     CPU: 14.282s
  CGroup: /system.slice/wazuh-agent.service
          ├─5350 /var/ossec/bin/wazuh-execd
          ├─5361 /var/ossec/bin/wazuh-agentd
          ├─5375 /var/ossec/bin/wazuh-syscheckd
          ├─5390 /var/ossec/bin/wazuh-logcollector
          └─5422 /var/ossec/bin/wazuh-modulesd
```

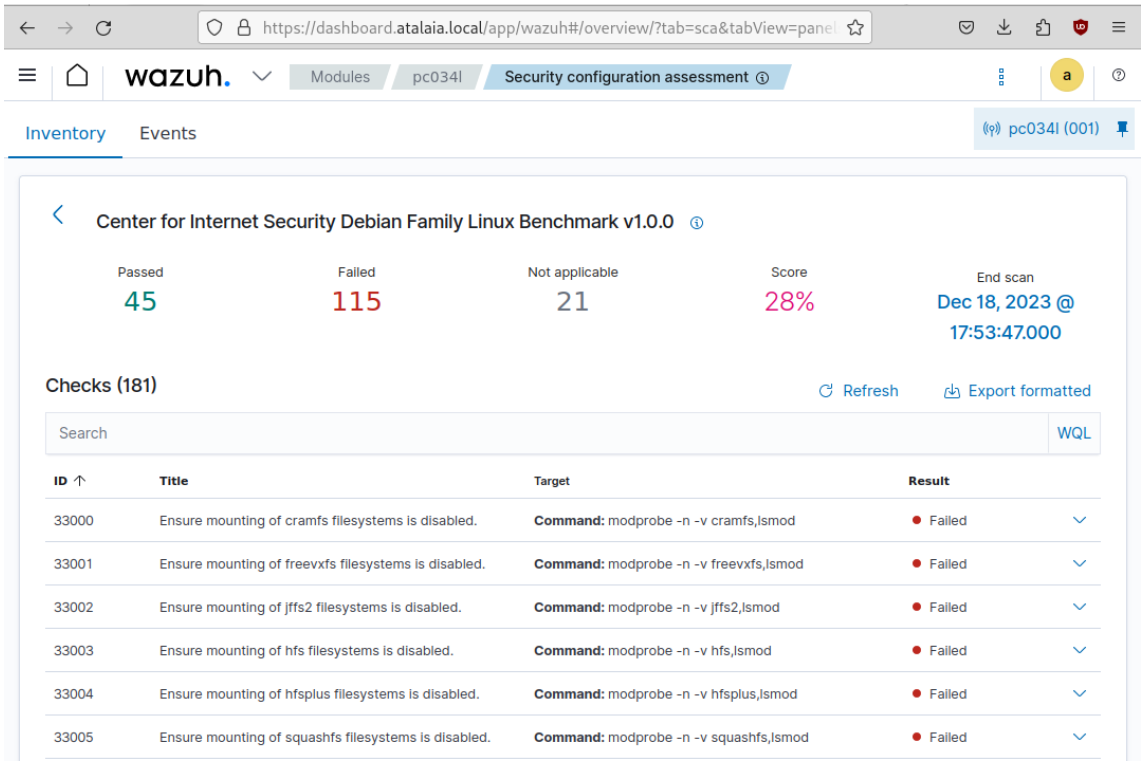


Figura 36: Evaluación del estado de la seguridad en pc034l

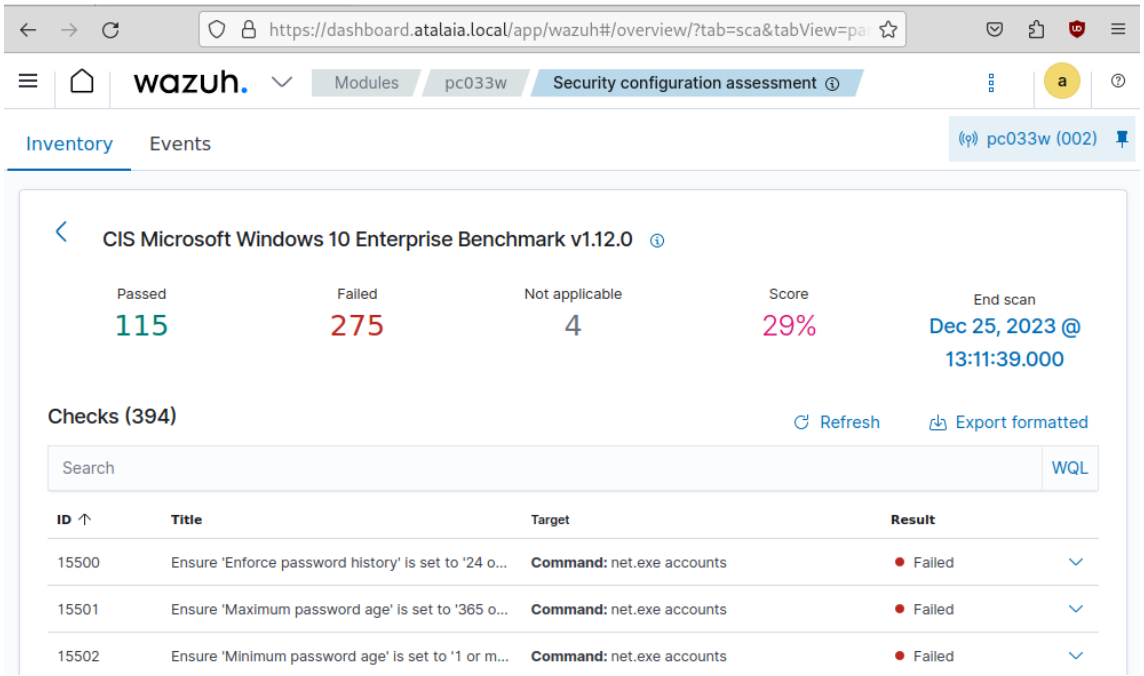


Figura 37: Evaluación del estado de la seguridad en pc033w

5.7. Integración con soluciones de terceros

Las capacidades de la plataforma de seguridad de Wazuh pueden complementarse con soluciones de terceros a través de sus API para disponer de funcionalidades tan variadas como inteligencia de amenazas, gestión de incidentes, SOAR o bloqueos de red a través de los cortafuegos corporativos. Para ello, su integración se realiza a través del gestor de Wazuh con los siguientes pasos [83].

- Se añade la sección XML integration en el archivo /var/ossec/etc/ossec.conf [84].

```
<!--Custom external Integration -->
<integration>
  <name>custom-integration</name>
  <hook_url>WEBHOOK</hook_url>
  <api_key>APIKEY</api_key> <!-- Replace with your external service API key -->
  <alert_format>json</alert_format>
  <level>10</level>
  <group>multiple_drops,authentication_failures</group>
  <options>{"data": "Custom data"}</options> <!-- Replace with your custom JSON object -->
</integration>
```

Fichero 27: ossec.conf (Integración de soluciones de terceros)

Opción	Descripción
name	Nombre de la integración, que debe coincidir con el del script que se utilizará para interactuar con la API de la solución de terceros. Los nombres de los scripts personalizados deben empezar por el prefijo “custom-“.
hook_url	URL a través de la cual se interactúa con la API del servicio que se quiere integrar.
api_key	Clave necesaria para el uso de la API
alert_format	Establece en que formato recibe las alertas el script que procesa la integración. Por defecto lo hace en estilo syslog y puede definirse en json.
level	Indica el nivel mínimo de la alerta para que se envíe la petición.
group	Especifica los grupos asociados a las reglas para los que se envíe la petición.
options	Permite sobreescribir o personalizar campos JSON.

Tabla 22: Opciones de configuración para la integración de soluciones de terceros

- Se crea el script que realizará las peticiones a la API del producto de terceros o se utilizan los disponibles para los servicios que vienen preconfigurados de serie.

```
root@manager:~# ls -la /var/ossec/integrations/
total 88
drwxr-x---. 2 root wazuh  4096 Dec  6 20:14 .
drwxr-x---. 1 root wazuh  4096 Dec  6 20:14 ..
-rwxr-x---. 1 root wazuh  1045 Nov 23 17:02 maltiverse
-rwxr-x---. 1 root wazuh 17358 Nov 23 17:02 maltiverse.py
-rwxr-x---. 1 root wazuh  1045 Nov 23 17:02 pagerduty
-rwxr-x---. 1 root wazuh  7078 Nov 23 17:02 pagerduty.py
-rwxr-x---. 1 root wazuh  1045 Nov 23 17:02 shuffle
-rwxr-x---. 1 root wazuh  7688 Nov 23 17:02 shuffle.py
-rwxr-x---. 1 root wazuh  1045 Nov 23 17:02 slack
-rwxr-x---. 1 root wazuh  7277 Nov 23 17:02 slack.py
-rwxr-x---. 1 root wazuh  1045 Nov 23 17:02 virustotal
-rwxr-x---. 1 root wazuh  9785 Nov 23 17:02 virustotal.py
```

Cualquier script personalizado se debe añadir al directorio /var/ossec/integrations con los mismos permisos que los de los archivos disponibles: root como propietario, wazuh como grupo y permisos 750 en formato numérico octal.

Maltiverse [85] es un servicio de inteligencia de amenazas que viene preconfigurado en la plataforma de Wazuh para interactuar con su API, por lo que solamente es necesario

ajustar la sección de integración necesaria para disponer de indicadores de compromiso de dominios, direcciones URL, hashes y direcciones IP.

```
root@manager:/# cat /var/ossec/etc/ossec.conf
<ossec_config>
  ...
  <!-- Maltiverse integration -->
  <integration>
    <name>maltiverse</name>
    <hook_url>https://api.maltiverse.com</hook_url>
    <api_key>APIKEY</api_key>
    <rule_id>550,554</rule_id>
    <alert_format>json</alert_format>
  </integration>
  ...
</ossec_config>
```

Fichero 28: ossec.conf (integración de Maltiverse)

La clave necesaria para utilizar la API del servicio está disponible en el apartado de subscripción de la sección de gestión de la cuenta de usuario.

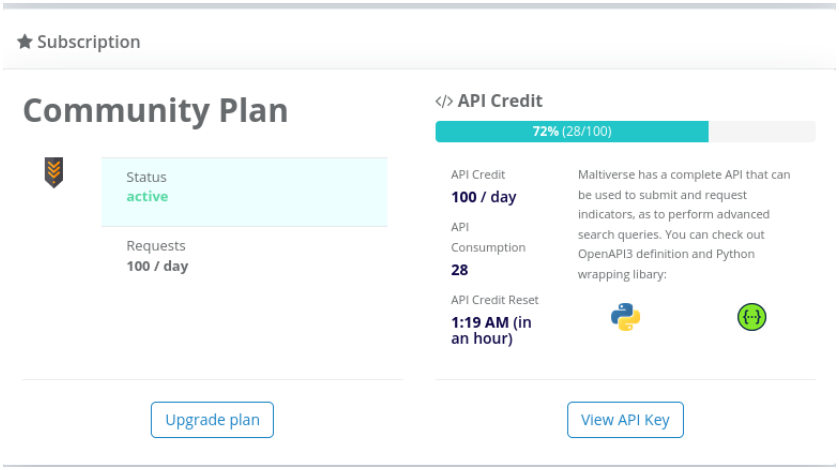


Figura 38: Consulta de la clave API del servicio Maltiverse

Finalmente, se pueden ver las alertas generadas por la plataforma Maltiverse al copiarse una muestra de malware de Linux.Mirai.B [86] en el directorio /etc/ monitorizado por el módulo FIM del dispositivo de punto final pc034l, donde los hashes de tamaños de 16 bytes y de 20 bytes que aparecen en la descripción están asociados respectivamente a los algoritmos MD5 y SHA1.

```
root@pc034l:~# md5sum /etc/Mirai.B.sh
390f1382237b5a01dd46bf1404c223e7  /etc/Mirai.B.sh
root@pc034l:~# sha1sum /etc/Mirai.B.sh
28976d0de5260fcdc620240bbad78424addd6232  /etc/Mirai.B.sh
```

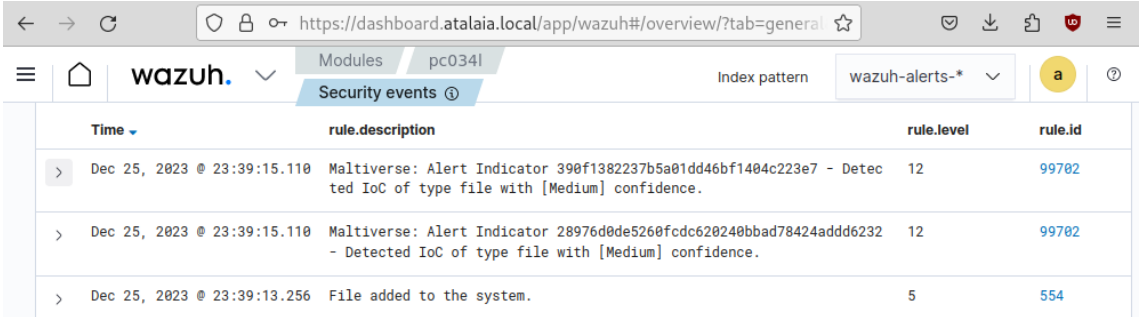


Figura 39: Detección de la muestra Linux.Mirai.B

6. Resultados

Los resultados obtenidos en este Trabajo se han ido detallando tanto en los capítulos 3, 4 y 5 como en los distintos anexos, por lo que en este apartado se realiza un resumen de los resultados obtenidos.

El primer resultado alcanzado en ese Trabajo es una comparativa cualitativa y cuantitativa (Tabla 7 y Figura 16) con la finalidad de ofrecer una visión global de las soluciones SIEM de uso gratuito existentes en la actualidad junto con sus fortalezas y debilidades. Los criterios seleccionados (Tabla 6) se han basado en la revisión bibliográfica y estado del arte de esta clase de productos, teniendo en cuenta que son complejos y en constante evolución.

El plan de implantación de la plataforma de seguridad Wazuh destaca por desplegar cada uno de los componentes principales de la plataforma, indexador, gestor y panel de control, en su propio contenedor y se basa en dos ejes fundamentales: el desarrollo del script mk-atalaia-certificates.sh (Anexo II), que permite generar los certificados necesarios para el entorno, y una completa personalización del despliegue en contenedores basado en Docker Compose. Una estructura de archivos y directorios definida en /opt/atalaia contiene tanto el fichero docker-compose.yml (Anexo III), que permite la configuración multicontenedor, como los certificados, ficheros de configuración y almacenes de claves necesarios para cada uno de los tres contenedores necesarios (Anexo IV, Anexo V y Anexo VI).

El estudio de medidas mitigadoras de las Tácticas, Técnicas y Procedimientos utilizadas por los ciberatacantes según estudios recientes sirve de base para, primero, obtener criterios fundamentales para la protección de los dispositivos finales (Figura 24), y, segundo, analizar las funcionalidades de la plataforma de seguridad Wazuh que tengan correspondencia con dichos criterios. Esto da lugar a un conjunto de casos de uso (capítulo 5) en donde se destacan los aspectos más relevantes de su configuración y se detallan sus resultados para verificar que son efectivos.

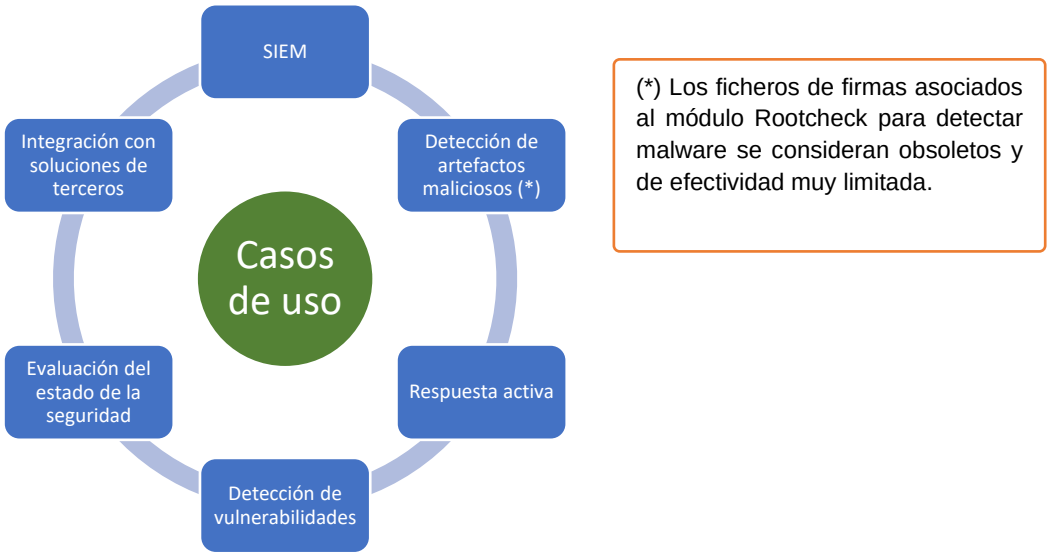


Figura 40: Casos de uso

7. Conclusiones y trabajos futuros

La temática de este trabajo se ha dividido en tres secciones claramente diferenciadas e interrelacionadas para, primero, disponer de un contexto actual de la solución Wazuh frente a posibles alternativas gratuitas, tanto de código abierto como propietarias, segundo, realizar una implantación del producto en contenedores que tenga en cuenta las buenas prácticas en materia de seguridad, y tercero, realizar una selección de casos de uso en base a las TTP utilizadas por los ciberatacantes.

7.1. Soluciones SIEM

El análisis de soluciones SIEM de uso gratuito (objetivo OE.1) pone de manifiesto que el número de alternativas de código abierto suficientemente completas es muy limitado en la actualidad, mientras que con los productos propietarios se debe tener cuidado con su licencia, porque algunos de ellos presentan características o limitaciones muy restrictivas.

La selección de criterios de comparación o selección de un SIEM no es trivial debido a su amplia variedad de funcionalidades y a su carácter transversal dentro de organizaciones que tienen distintos requisitos. Es por ello, que se considera necesario aunar el conocimiento del estado del arte y las tendencias del mercado de la ciberseguridad con los casos de uso de las organizaciones.

Dos de los criterios a los que se les ha dado un mayor peso son el soporte y la documentación, dado que se consideran imprescindibles en una solución tan compleja como el SIEM, que se integra en la infraestructura tecnológica de la empresa y cuyo tiempo de vida suele ser prolongado. El soporte desempeña un papel importante cuando la documentación no es lo suficientemente detallada a la hora de realizar ciertos ajustes o resolver algún problema, tal y como se ha observado durante la implantación y el análisis de casos de uso de Wazuh.

La frecuencia de actualización de las soluciones SIEM para corregir errores y añadir nuevas funcionalidades puede servir como indicador de cómo se adaptan a la evolución de los distintos productos TIC.

7.2. Implantación de Wazuh

La implantación de la plataforma de seguridad Wazuh en contenedores (objetivo OE.3) ha supuesto un desafío por toda la información previa que ha sido necesario analizar y recopilar, incluida su arquitectura (objetivo OE.2), para ofrecer una propuesta viable que cifre sus comunicaciones (objetivo OE.4). En este Trabajo ha sido necesario conjugar conocimientos de distintos ámbitos de las TIC como redes, sistemas, contenedores, criptografía y parametrización/programación de scripts.

La configuración Docker Compose proporcionada por Wazuh difícilmente tiene cabida en un entorno en producción, por lo que su personalización se hace obligatoria para eliminar ajustes no necesarios, añadir capas de seguridad cruciales y realizar las adaptaciones necesarias para que funcione de manera óptima en el entorno TIC disponible.

El objetivo OE.4 busca la protección de las comunicaciones de los distintos componentes mediante certificados digitales, por lo que se ha automatizado su creación mediante un script que tiene una doble finalidad: primero, rellenar sus campos con información personalizada y, segundo, aplicar las recomendaciones del CCN en lo referente a los algoritmos de clave pública, sus tamaños de clave y las funciones resumen de firma. Aunque la plataforma de seguridad Wazuh soporta actualmente sólo claves privadas RSA, dicho script permite también la creación de certificados firmados con algoritmos ECDSA, al ser cada vez más utilizados y ofrecer un mejor rendimiento para tamaños de clave equivalentes.

Otro aspecto relevante que se debe valorar al trabajar con contenedores es la selección adecuada de la modalidad de red en Docker Engine, de tal forma que se adapte a las políticas de seguridad de la red subyacente y no suponga accesos no contemplados o no autorizados.

No es menos importante la gestión de información sensible como las credenciales o claves privadas que se deben mantener a buen recaudo, que en un entorno de Docker no distribuido no admite las ventajas de los denominados secretos, es decir, un repositorio centralizado cifrado que transmite los datos de manera segura a los contenedores que deban utilizarlos. Ante este inconveniente, se ha seguido un doble enfoque: uso de los almacenes de claves o keystores que proporciona Wazuh siempre que sea posible y la definición del resto de credenciales en un fichero oculto independiente de la configuración principal, de tal manera que sea posible excluirlo de los procesos de seguimiento y confirmación que se utilizan en los repositorios.

7.3. Casos de uso

Los casos de uso que se pueden aplicar con Wazuh para la protección de dispositivos finales es amplia, variada y dependiente de cada entorno TIC, por lo que su selección (objetivo OE.5) ha necesitado de un proceso de abstracción que tuviese en cuenta las TTP utilizadas por los ciberatacantes y radica en seis ejes fundamentales: visibilidad de la actividad de interés, detección de artefactos maliciosos, respuesta activa frente a eventos definidos, conocimiento tanto de las vulnerabilidades existentes como de la configuración de seguridad de los endpoints y, finalmente, la integración con herramientas de terceros que amplíe las capacidades de la plataforma. El objetivo OE.5 pretende, por tanto, ofrecer una visión global de las funcionalidades relevantes disponibles junto con sus opciones de configuración, pero sin olvidar que la implantación en una organización implica un alto componente de personalización.

La detección de malware mediante ficheros de firmas que proporciona el módulo Rootcheck se considera escasa y anticuada, de manera que se hace imprescindible complementarlo con una solución EPP de terceros o mediante la obtención de ficheros de firmas actualizados que se integren con la plataforma de seguridad de Wazuh.

Los tiempos de las duraciones de las acciones de las respuestas activas junto a la frecuencia con la que saltan las reglas en una determinada ventana temporal deben adaptarse a la tipología y patrones de tráfico de la red. Los valores en los ejemplos de casos de uso correspondientes se eligieron para mostrar su efectividad en un tiempo reducido.

7.4. Planificación y metodología

La metodología y planificación adoptada se consideran en general adecuadas para el tipo de trabajo realizado y para la curva de aprendizaje de un producto tan complejo como es el SIEM con sus funcionalidades añadidas de XDR, de tal forma que se ha conseguido un refinamiento continuo de los resultados, una revisión bibliográfica y análisis de la documentación detallada, la parametrización de la solución para determinados casos de usos y la aplicación de buenas prácticas en el ámbito de la seguridad informática.

El pico de enfermedades respiratorias de estas Navidades del año 2023 [87] ha puesto en evidencia que este riesgo no se ha identificado y valorado adecuadamente, sobre todo teniendo en cuenta la época del año y el hecho de que solamente existía una única persona como responsable del Trabajo.

7.5. Impactos en sostenibilidad, ético-social y de diversidad

Los impactos positivos en sostenibilidad, ético-social y de diversidad son los previstos, debido a que el entorno virtual propuesto sobre el que se ejecuta la plataforma Wazuh y las herramientas utilizadas, ambas de código abierto de uso gratuito, no aportan efectos adversos. Se obtiene, por tanto, una plataforma moderna, eficiente, portable y escalable al alcance de todo el mundo que protege los activos de las organizaciones para que puedan seguir desempeñando su misión y responsabilidad social, favoreciendo su crecimiento económico y el mantenimiento de los puestos de trabajo.

7.6. Líneas de trabajo futuro

El objetivo general (OG) de este Trabajo pretende servir como punto de partida desde el que las organizaciones puedan realizar su propia implantación, aunque hay líneas de trabajo futuro que quedan fuera del alcance del este proyecto como las que se indican a continuación.

- Plan de manejo de la solución adecuado a los distintos perfiles de usuario de la organización y en el que se tenga en cuenta las distintas posibilidades que ofrece el panel de control de Wazuh.
- Plan de escalabilidad de la solución que permita adaptarse a las necesidades de las organizaciones.
- Plan de copias de seguridad y plan de recuperación ante desastres de la plataforma.
- Utilización de herramientas de automatización para el despliegue de los componentes centrales y los agentes.
- Mejora de la gestión de secretos de la plataforma, optando por soluciones distribuidas de sistemas de contenedores de ser factible.
- Integración con soluciones de terceros y con dispositivos de la infraestructura TIC de la organización que ayuden a la plataforma a ser más efectiva en su detección y respuesta, con el uso de productos que proporcionen capacidades SOAR o de aprendizaje automático (ML) e inteligencia artificial (IA).
- Recogida de mensajes del diario de eventos journald para prescindir de los logs equivalentes en formato syslog.

8. Glosario

Access-as-a-Service
: Modalidad de negocio conocido como AaaS en el que los ciberdelincuentes venden credencials de acceso robadas, 1

bastionado
: También conocido como hardening, hace referencia al refuerzo de medidas de seguridad para reducir las vulnerabilidades existentes en un sistema informático., 11

CDB
: Lista que hace referencia a una base de datos constante que se caracteriza por no cambiar mientras está en uso., 44, 45

CVE
: Información de vulnerabilidades y exposiciones comunes, donde cada una de ellas tiene un identificador alfanumérico propio., 40, 41, 42, 69

EPP
: Plataforma de protección de dispositivos de punto final, conocida como antivirus tradicional, 12, 42, 61

Hacker-as-a-Service
: Modelo de negocio conocido como HaaS en el cual se contrata a un hacker a cambio de una contraprestación económica, 1

HIDS
: Sistema de detección de intrusiones a nivel de host, 20

malware
: Software malicioso que tiene el objetivo de dañar o infiltrarse en un dispositivo o sistema, 1

Malware-as-a-Service
: Modelo de negocio conocido como MaaS en el cual los ciberdelincuentes proporcionan malware a sus clientes a cambio del pago de una tarifa, 1

on-premise
: Instalación realizada en las dependencias y en la infraestructura tecnológica de la organización., 15

SIEM
: Gestión de eventos e información de seguridad, 1, 2, 4, 5, 8, 10, 12, 15, 16, 18, 19, 34, 38, 59, 60

TTP
: Tácticas, Técnicas y Procedimientos utilizados por los ciberatacantes según MITRE ATT&CK., 60, 61

vCPU
: Procesador o CPU virtual asignada a una máquina virtual, 5

XDR
: Detección y Respuesta Extendida. También conocido como Detección y Respuesta Ampliada., 1, 2, 8, 12

9. Bibliografía

1. Deloitte Development LLC. Deloitte Cyber Threat Trends. Deloitte; 2023 mar.

2. ENISA. ENISA Threat Landscape 2022 [Internet]. European Union Agency for Cybersecurity (ENISA); 2023 [citado 29 de septiembre de 2023]. Disponible en: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2022

3. CCN-CERT. Ciberamenazas y Tendencias. Edición 2022 [Internet]. CCN-CERT; 2022 sep [citado 29 de septiembre de 2023]. Disponible en: https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/6786-ccn-cert-ia-24-22-ciberamenazas-y-tendencias-edicion-2022-1.html

4. INCIBE. Balance de ciberseguridad 2022 [Internet]. Instituto Nacional de Ciberseguridad; [citado 29 de septiembre de 2023] p. 1. Disponible en: https://portal.mineco.gob.es/RecursosNoticia/mineco/prensa/noticias/2023/Balance_de_Ciberseguridad_2022_INCIBE.pdf

5. Wazuh, Inc. Wazuh: Unified XDR and SIEM protection for endpoints and cloud workloads. [Internet]. 2023. Disponible en: https://wazuh.com/

6. Rogers Bruce. Can Wazuh Become The World's Largest Open Source Cybersecurity Platform And IPO Without VC Funding? [Internet]. 2023 [citado 5 de noviembre de 2023]. Disponible en: https://www.forbes.com/sites/brucerogers/2023/01/05/can-wazuh-become-the-worlds-largest-open-source-cybersecurity-platform-and-ipo-without-vc-funding/

7. Naciones Unidas. Objetivos de desarrollo sostenible [Internet]. [citado 29 de septiembre de 2023]. Disponible en: https://www.un.org/sustainabledevelopment/es/

8. TeamGantt. TeamGantt [Internet]. 2023. Disponible en: https://www.teamgantt.com/

9. Atlassian. Trello [Internet]. 2023. Disponible en: https://trello.com/

10. The MITRE Corporation. MITRE ATT&CK® [Internet]. 2023 [citado 15 de octubre de 2023]. Disponible en: https://attack.mitre.org/

11. CISA. Risk and Vulnerability Assessments | CISA [Internet]. 2023 [citado 1 de noviembre de 2023]. Disponible en: https://www.cisa.gov/resources-tools/resources/risk-and-vulnerability-assessments

12. Corporation for Digital Scholarship. Zotero: Your personal research assistant [Internet]. Disponible en: https://www.zotero.org/

13. Wazuh Inc. Release notes · Wazuh documentation [Internet]. 2023 [citado 24 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/release-notes/index.html

14. Vivancos, Elisa (INCIBE). Los 10 vectores de ataque más utilizados por los ciberdelincuentes [Internet]. 2022 [citado 7 de octubre de 2023]. Disponible en: https://www.incibe.es/empresas/blog/los-10-vectores-ataque-mas-utilizados-los-ciberdelincuentes

15. INCIBE. Bastionado de sistemas: el caso de Linux [Internet]. 2023 [citado 7 de octubre de 2023]. Disponible en: https://www.incibe.es/incibe-cert/blog/bastionado-sistemas-el-caso-linux

16. Comerford, Linda. Why small businesses are vulnerable to cyberattacks [Internet]. 2022 [citado 19 de octubre de 2023]. Disponible en: https://www.securitymagazine.com/blogs/14-security-blog/post/97694-why-small-businesses-are-vulnerable-to-cyberattacks

17. ENISA. SME Cybersecurity [Internet]. 2023 [citado 19 de octubre de 2023]. Disponible en: https://www.enisa.europa.eu/topics/cybersecurity-education/sme_cybersecurity

18. CISA. Small and Medium Businesses [Internet]. [citado 19 de octubre de 2023]. Disponible en: https://www.cisa.gov/audiences/small-and-medium-businesses

19. INCIBE. Empresas [Internet]. [citado 19 de octubre de 2023]. Disponible en: https://www.incibe.es/empresas

20. Leibovich, Mariela. 5 Tips for SMBs to Avoid the Next Cyber-Attack [Internet]. 2020 [citado 19 de octubre de 2023]. Disponible en: https://www.cyrebro.io/blog/5-tips-for-smbs-to-avoid-the-next-cyber-attack/

21. Statista Market Insights. Endpoint Security: Revenue. 2023 sep.

22. Fortinet, Inc. Detección y respuesta extendida (XDR) [Internet]. [citado 8 de octubre de 2023]. Disponible en: https://www.fortinet.com/lat/resources/cyberglossary/what-is-XDR

23. Mellen, Allie; Blankenship, Joseph; Pollard, Jeff; Turner, Steve; Cser, Andras; Tatro, Alexis; et al. Adapt Or Die: XDR Is On A Collision Course With SIEM And SOAR. 28 de abril de 2021;

24. Palo Alto Networks. What is the Difference Between XDR vs. SIEM? [Internet]. [citado 8 de octubre de 2023]. Disponible en: https://www.paloaltonetworks.com/cyberpedia/what-is-xdr-vs-siem

25. OSI. Open Source Initiative [Internet]. Disponible en: https://opensource.org/

26. AT&T Business. AlienVault OSSIM [Internet]. 2023. Disponible en: https://cybersecurity.att.com/products/ossim

27. VMware. Which technologies and solutions are currently in use, planned for upgrade, or planned for initial use by your organization within the next 12 months? [Internet]. 2022 may [citado 5 de noviembre de 2023]. Disponible en:

https://www.statista.com/statistics/1331684/cyber-security-technologies-in-use-upgrade-worldwide/

28. The Linux Foundation ®. Kubernetes [Internet]. 2023. Disponible en: https://kubernetes.io/

29. Docker Inc. Use containers to Build, Share and Run your applications [Internet]. 2023 [citado 8 de octubre de 2023]. Disponible en: https://www.docker.com/resources/what-container/

30. CNCF. CNCF Annual Survey 2022 [Internet]. 2023 ene [citado 18 de noviembre de 2023]. Disponible en: https://www.cncf.io/reports/cncf-annual-survey-2022/#

31. Red Hat. State of Kubernetes security report [Internet]. 2022 [citado 18 de noviembre de 2023] p. 18. Disponible en: https://www.redhat.com/rhdc/managed-files/cl-state-of-kubernetes-security-report-2022-ebook-f31209-202205-en.pdf

32. Docker Inc. Alternative container runtimes [Internet]. Docker Documentation. [citado 18 de noviembre de 2023]. Disponible en: https://docs.docker.com/engine/alternative-runtimes/

33. Datanyze. Containerization Market Share Report [Internet]. 2023 [citado 18 de noviembre de 2023]. Disponible en: https://www.datanyze.com/market-share/containerization--321

34. Katz School of Science and Health, Yeshiva University. The 10 Best Open Source SIEM Tools for Cybersecurity Experts [Internet]. Katz. 2022 [citado 12 de octubre de 2023]. Disponible en: https://online.yu.edu/katz/blog/the-10-best-open-source-siem-tools

35. Tariq, Aamna; Manzoor, Jawad; Aziz, Muhammad Ammar; Tariq, Zain Ul Abideen; Masood, Ammar. Open source SIEM solutions for an enterprise. Inf Comput Secur. 1 de enero de 2023;31(1):88-107.

36. Graylog. Graylog: Industry Leading Log Management & SIEM [Internet]. Graylog; 2023. Disponible en: https://graylog.org/

37. Security Onion Solutions, LLC. Security Onion Solutions [Internet]. Security Onion Solutions, LLC; 2023. Disponible en: https://securityonionsolutions.com/

38. Splunk Inc. Splunk | The Key to Enterprise Resilience [Internet]. Splunk Inc.; 2024. Disponible en: https://www.splunk.com

39. Elasticsearch B.V. Elasticsearch Platform: Encuentra respuestas en tiempo real a escala [Internet]. Elasticsearch B.V.; 2023. Disponible en: https://www.elastic.co/es/

40. Wazuh Inc. Getting started with Wazuh · Wazuh documentation [Internet]. 2023 [citado 28 de octubre de 2023]. Disponible en: https://documentation.wazuh.com/current/getting-started/index.html

41. OpenSearch. OpenSearch documentation [Internet]. OpenSearch documentation. [citado 28 de octubre de 2023]. Disponible en: https://opensearch.org/docs/latest/

42. Red Hat, Inc. Converting from an RPM-based Linux distribution to RHEL Red Hat Enterprise Linux 8 | Red Hat Customer Portal [Internet]. [citado 3 de noviembre de 2023]. Disponible en: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html-single/converting_from_an_rpm-based_linux_distribution_to_rhel/index

43. The CentOS Project. End dates are coming for CentOS Stream 8 and CentOS Linux 7 – Blog.CentOS.org [Internet]. 2023 [citado 3 de noviembre de 2023]. Disponible en: https://blog.centos.org/2023/04/end-dates-are-coming-for-centos-stream-8-and-centos-linux-7/

44. CIQ, Inc. Migration Landing Page | CIQ [Internet]. [citado 3 de noviembre de 2023]. Disponible en: https://ciq.com/migrate/centos-to-rocky-linux/

45. Rocky Enterprise Software Foundation. Rocky Linux Release Version Guide - Rocky Linux Wiki [Internet]. [citado 3 de noviembre de 2023]. Disponible en: https://wiki.rockylinux.org/rocky/version/

46. Center for Internet Security. CIS Benchmarks™ FAQ [Internet]. [citado 4 de noviembre de 2023]. Disponible en: https://www.cisecurity.org/cis-benchmarks/cis-benchmarks-faq/

47. Docker Inc. Packet filtering and firewalls [Internet]. Docker Documentation. [citado 3 de noviembre de 2023]. Disponible en: https://docs.docker.com/network/packet-filtering-firewalls/

48. Microsoft. Windows 10, version 22H2 end of support date updated - Microsoft Lifecycle [Internet]. 2023 [citado 6 de noviembre de 2023]. Disponible en: https://learn.microsoft.com/en-us/lifecycle/announcements/windows-10-22h2-end-of-support-update

49. Dochy Esben. Is Your Business Ready for Windows 11? [Internet]. Lansweeper IT Asset Management. 2023 [citado 6 de noviembre de 2023]. Disponible en: https://www.lansweeper.com/itam/is-your-business-ready-for-windows-11/

50. Microsoft. Windows 10 Enterprise LTSC 2019 - Microsoft Lifecycle [Internet]. [citado 6 de noviembre de 2023]. Disponible en: https://learn.microsoft.com/en-us/lifecycle/products/windows-10-enterprise-ltsc-2019

51. CCN-CERT. Recomendaciones de Implementación de HTTPS. Informe de buenas prácticas [Internet]. Paseo de la Castellana 109, 28046 Madrid: CCN-CERT; 2022 ene [citado 11 de noviembre de 2023]. Disponible en: https://www.ccn-cert.cni.es/es/pdf/informes-de-ciberseguridad-ccn-cert/informes-ccn-cert-publicos/2112-ccn-cert-bp-07-recomendaciones-implementacion-https/file.html

52. Centro Criptológico Nacional. Criptología de empleo en el Esquema Nacional de Seguridad [Internet]. P.º de la Castellana 109, 28046 Madrid: Centro Criptológico

Nacional; 2022 [citado 11 de noviembre de 2023]. Disponible en: https://www.ccn-cert.cni.es/es/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/513-ccn-stic-807-criptologia-de-empleo-en-el-ens/file?format=html

53. Tschofenig Hannes; Pégourié-Gonnard Manuel. Performance Investigations [Internet]. 2015 mar 25 [citado 11 de noviembre de 2023]. Disponible en: https://www.ietf.org/proceedings/92/slides/slides-92-lwig-3.pdf

54. Wazuh Inc. Docker installation - Deployment on Docker · Wazuh documentation [Internet]. 2023 [citado 3 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/deployment-options/docker/docker-installation.html

55. Docker Inc. Compose file [Internet]. Docker Documentation. [citado 25 de noviembre de 2023]. Disponible en: https://docs.docker.com/compose/compose-file/03-compose-file/

56. Docker Inc. Ways to set environment variables in Compose [Internet]. Docker Documentation. [citado 25 de noviembre de 2023]. Disponible en: https://docs.docker.com/compose/environment-variables/set-environment-variables/

57. Wazuh Inc. Wazuh agent enrollment - User manual · Wazuh documentation [Internet]. 2023 [citado 25 de noviembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/agent-enrollment/index.html

58. Wazuh Inc. Deployment variables - User manual · Wazuh documentation [Internet]. 2023 [citado 25 de noviembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/deployment-variables/deployment-variables.html

59. Wazuh Inc. Local configuration (ossec.conf) - Reference · Wazuh documentation [Internet]. 2023 [citado 26 de noviembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/index.html

60. Wazuh Inc. Wazuh archives - Wazuh server administration · Wazuh documentation [Internet]. 2023 [citado 7 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/manager/wazuh-archives.html

61. Wazuh Inc. Wazuh indexer indices - Wazuh indexer · Wazuh documentation [Internet]. 2023 [citado 7 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html

62. Wazuh Inc. Ruleset - User manual · Wazuh documentation [Internet]. 2023 [citado 14 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/ruleset/index.html

63. Wazuh Inc. global - Local configuration (ossec.conf) · Wazuh documentation [Internet]. 2023 [citado 7 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/global.html

64. Wazuh Inc. Configuring syslog on the Wazuh server - Log data collection [Internet]. 2023 [citado 14 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/capabilities/log-data-collection/syslog.html

65. Wazuh Inc. System inventory - Capabilities · Wazuh documentation [Internet]. 2023 [citado 7 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/capabilities/system-inventory/index.html

66. Wazuh Inc. wodle name="syscollector" - Local configuration (ossec.conf) [Internet]. 2023 [citado 7 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/wodle-syscollector.html

67. Wazuh Inc. Vulnerability detection - Capabilities · Wazuh documentation [Internet]. 2023 [citado 8 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/capabilities/vulnerability-detection/index.html

68. Wazuh Inc. vulnerability-detector - Local configuration (ossec.conf) [Internet]. 2023 [citado 8 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/vuln-detector.html

69. Debian Security. CVE-2023-28531 [Internet]. 2023 [citado 8 de diciembre de 2023]. Disponible en: https://security-tracker.debian.org/tracker/CVE-2023-28531

70. Wazuh Inc. Malware detection - Capabilities · Wazuh documentation [Internet]. 2023 [citado 19 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/capabilities/malware-detection/index.html

71. Wazuh Inc. rootcheck - Local configuration (ossec.conf) · Wazuh documentation [Internet]. 2023 [citado 8 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/rootcheck.html

72. Wazuh Inc. syscheck - Local configuration (ossec.conf) · Wazuh documentation [Internet]. 2023 [citado 9 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/syscheck.html

73. EICAR. Download Anti Malware Testfile [Internet]. 2023 [citado 10 de diciembre de 2023]. Disponible en: https://www.eicar.org/download-anti-malware-testfile/

74. Wazuh Inc. Rules classification - Ruleset · Wazuh documentation [Internet]. 2023 [citado 18 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/ruleset/rules-classification.html

75. Wazuh Inc. Ruleset XML syntax - Ruleset · Wazuh documentation [Internet]. 2023 [citado 10 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/ruleset/ruleset-xml-syntax/index.html

76. Cisco. ClamAV [Internet]. 2023 [citado 23 de diciembre de 2023]. Disponible en: https://www.clamav.net/

77. Wazuh Inc. Log data analysis - Log data collection · Wazuh documentation [Internet]. 2023 [citado 10 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/capabilities/log-data-collection/log-data-analysis.html

78. Wazuh Inc. active-response - Local configuration (ossec.conf) [Internet]. 2023 [citado 16 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/active-response.html

79. van Hauser Heuse, Marc; Maciejak, David. Hydra [Internet]. 2023 [citado 18 de diciembre de 2023]. Disponible en: https://github.com/vanhauser-thc/thc-hydra

80. Wazuh Inc. Security Configuration Assessment - Capabilities · Wazuh documentation [Internet]. 2023 [citado 18 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/capabilities/sec-config-assessment/index.html

81. Wazuh Inc. cis_debian12.yml [Internet]. 2023 [citado 18 de diciembre de 2023]. Disponible en: https://github.com/wazuh/wazuh/blob/cddff85038a4ba75460bee8e3fab1df22185f1e5/ruleset/sca/debian/cis_debian12.yml

82. Wazuh Inc. sca - Local configuration (ossec.conf) · Wazuh documentation [Internet]. 2023 [citado 18 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/sca.html

83. Wazuh Inc. Integration with external APIs - Wazuh server administration [Internet]. 2023 [citado 25 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/manager/manual-integration.html

84. Wazuh Inc. integration - Local configuration (ossec.conf) · Wazuh documentation [Internet]. 2023 [citado 25 de diciembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/integration.html

85. Maltiverse. Maltiverse - Actionable Threat Intelligence [Internet]. Disponible en: https://maltiverse.com/start

86. ytisf. theZoo - A Live Malware Repository. Linux.Mirai.B [Internet]. 2021 [citado 25 de diciembre de 2023]. Disponible en: https://github.com/ytisf/theZoo/tree/master/malware/Binaries/Linux.Mirai.B

87. Instituto de Salud Carlos III. Temporada_Gripe_23-24 [Internet]. 2023 [citado 6 de enero de 2023]. Disponible en: https://www.isciii.es/QueHacemos/Servicios/VigilanciaSaludPublicaRENAVE/EnfermedadesTransmisibles/Paginas/Temporada_Gripe_23-24.aspx

88. The Rocky Enterprise Software Foundation. System Startup - Documentation [Internet]. 2023 [citado 1 de noviembre de 2023]. Disponible en: https://docs.rockylinux.org/books/admin_guide/10-boot/#the-grub2-bootloader

89. Docker Inc. Change Docker Desktop settings on Linux [Internet]. Docker Documentation. [citado 29 de octubre de 2023]. Disponible en: https://docs.docker.com/desktop/settings/linux/

90. Docker Inc. Docker daemon configuration overview [Internet]. Docker Documentation. [citado 29 de octubre de 2023]. Disponible en: https://docs.docker.com/config/daemon/

91. Docker Inc. dockerd [Internet]. Docker Documentation. [citado 29 de octubre de 2023]. Disponible en: https://docs.docker.com/engine/reference/commandline/dockerd/

92. Debian. 5.3. Boot Parameters [Internet]. Debian GNU/Linux Installation Guide. [citado 7 de noviembre de 2023]. Disponible en: https://www.debian.org/releases/stable/amd64/ch05s03.en.html

93. Docker Inc. Start containers automatically [Internet]. Docker Documentation. [citado 25 de noviembre de 2023]. Disponible en: https://docs.docker.com/config/containers/start-containers-automatically/

94. OpenSearch. Security configuration [Internet]. OpenSearch documentation. 2023 [citado 25 de noviembre de 2023]. Disponible en: https://opensearch.org/docs/latest/security/configuration/index/

95. Wazuh Inc. Deployment on Docker - Installation alternatives · Wazuh documentation [Internet]. 2023 [citado 25 de noviembre de 2023]. Disponible en: https://documentation.wazuh.com/current/deployment-options/docker/index.html

96. Wazuh Inc. Wazuh containers for Docker [Internet]. Wazuh; 2023 [citado 25 de noviembre de 2023]. Disponible en: https://github.com/wazuh/wazuh-docker

97. OpenSearch. Configuring OpenSearch [Internet]. OpenSearch documentation. 2023 [citado 25 de noviembre de 2023]. Disponible en:

https://opensearch.org/docs/latest/install-and-configure/configuring-opensearch/index/

98. Wazuh Inc. Configuration - RESTful API · Wazuh documentation [Internet]. 2023 [citado 26 de noviembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/api/configuration.html

99. OpenSearch. OpenSearch Dashboards multi-tenancy [Internet]. OpenSearch documentation. 2023 [citado 3 de diciembre de 2023]. Disponible en: https://opensearch.org/docs/1.3/security/access-control/multi-tenancy/

100. OpenSearch. OpenSearch Dashboards [Internet]. OpenSearch documentation. 2023 [citado 3 de diciembre de 2023]. Disponible en: https://opensearch.org/docs/latest/dashboards/index/

101. OpenSearch. Multi-tenancy configuration [Internet]. OpenSearch documentation. 2023 [citado 3 de diciembre de 2023]. Disponible en: https://opensearch.org/docs/latest/security/multi-tenancy/multi-tenancy-config/

102. Wazuh Inc. Configuration file - Wazuh dashboard · Wazuh documentation [Internet]. 2023 [citado 26 de noviembre de 2023]. Disponible en: https://documentation.wazuh.com/current/user-manual/wazuh-dashboard/config-file.html

10. Anexos

Anexo I: Configuración del entorno necesario para Wazuh

I.1. Instalación de la máquina virtual para los componentes centrales de Wazuh

Toda la infraestructura que presta servicios en la plataforma de seguridad de Wazuh se configura en un sistema de contenedores Docker disponible en una máquina virtual de Oracle VM VirtualBox. Los pasos que se indican a continuación pueden realizarse en cualquiera máquina física que soporte la instalación del sistema operativo Rocky Linux 9.2.

I.1.1. Máquina virtual

La configuración relevante de la máquina virtual es:

Elemento	Configuración
BIOS EFI	Activada
Secure Boot	Activado
vCPU	4
Memoria RAM	8 GB
Memoria de vídeo	16 MB
Disco duro virtual	60 GB
Audio	Deshabilitado
Red	Red NAT para la red 10.10.10.0/24. DHCP activado.

Tabla 23: Configuración de la MV para atl004s

I.1.2. Instalación de Rocky Linux 9.2

Se realiza una instalación mínima de Rocky Linux 9.2 en arquitectura x86_64 (amd64) de 64 bits con la siguiente configuración:

- Opciones de la instalación

En este apartado se enumeran distintas opciones de configuración que no incluyen aspectos de red o de particionado.

Nombre	Configuración
Regionalización	
Teclado	Español; Castellano (Español)
Soporte de Idiomas	Español (España), English (United States)
Fecha y hora	Huso horario de Europa/Madrid
Software	
Fuente de instalación	Réplica de red más cercana
Selección de software	Instalación mínima/Estándar
Sistema	
Destino de la instalación	Disco local Particionado manual
KDUMP	Habilitado/Configuración automática
Perfil de seguridad	CIS RHEL 9 Benchmark for Level 2 - Server
Ajustes de usuario	
Contraseña de root	Cuenta de root deshabilitada

Nombre	Configuración
Creación de usuario	Nombre de usuario: boole Hacer este usuario como administrador Se requiere contraseña para usar esta cuenta Pertenece al grupo *wheel*

Tabla 24: Opciones de instalación de atl004s

La instalación estándar definida ya incluye de serie el servidor OpenSSH, para poder administrar el sistema de manera remota, y el paquete OpenSSL, que permite crear certificados y claves privadas para cifrar las comunicaciones.

```
dnf list installed "openss*"
Error al cargar el complemento "config_manager": '*prog'
Paquetes instalados
openssh.x86_64                          8.7p1-30.el9_2                  @baseos
openssh-clients.x86_64                  8.7p1-30.el9_2                  @baseos
openssh-server.x86_64                   8.7p1-30.el9_2                  @baseos
openssl.x86_64                          1:3.0.7-17.el9_2                @baseos
openssl-libs.x86_64                     1:3.0.7-17.el9_2                @baseos
```

Se ha escogido el perfil de seguridad *CIS RHEL 9 Benchmark for Level 2 – Server* que se incluye en el proceso de instalación con el fin de partir de una configuración de seguridad elevada.

- Configuración de red:

Se realiza la configuración manual de la red con los siguientes valores:

Nombre	Configuración
Nombre de equipo	atl004s
IPv4	Manual
Dirección IP	10.10.10.4
Máscara de subred	255.255.255.0 (/24)
Pasarela	10.10.10.1
Servidores DNS	10.10.10.1
Dominios de búsqueda	atalaia.local
IPv6	Desactivado
Servidor NTP	2.rocky.pool.ntp.org

Tabla 25: Configuración de red de atl004s

- Particiones definidas

Se aplica el siguiente particionado manual de estilo GPT con opciones de montaje específicas para restringir las acciones que se puede realizar en determinadas particiones.

Nombre	FSTYPE	Mounpoint	Tamaño	Opciones de montaje
sda1	vfat	/boot/efi	512MiB	umask=0077,shortname=winnt
sda2	ext4	/boot	1GiB	defaults
sda3	LVM	-	52GiB	-
atalaia--vg-root	ext4	/	40GiB	defaults
atalaia--vg-home	ext4	/home	3GiB	defaults,nodev,nosuid
atalaia--vg-var_log	ext4	/var/log	1GiB	defaults,nodev,noexec,nosuid
atalaia--vg-var_log_audit	ext4	/var/log/audit	2GiB	defaults,nodev,noexec,nosuid
atalaia--vg-var_tmp	ext4	/var/tmp	1GiB	defaults,nodev,noexec,nosuid
atalaia--vg-tmp	ext4	/tmp	1GiB	defaults,nodev,noexec,nosuid
atalaia--vg-var	ext4	/var	4GB	defaults,nodev,nosuid

Tabla 26: Particiones definidas en atl004s

La opción de montaje defaults equivale al conjunto de opciones por defecto rw, suid, dev, exec, auto, nouser y async, algunas de las cuales son restringidas con los siguientes parámetros:

- nodev para impedir la instalación de dispositivos.
- noexec para no permitir la ejecución de binarios.
- nosuid para ignorar los bits SUID y/o SGID de los ficheros.

No se ha definido una partición de *swap* al disponer de suficiente cantidad de memoria RAM para el montaje que se plantea. Será posible añadirla posteriormente en caso de ser necesario.

- Arranque del sistema

Se forzará el cambio de contraseña al iniciar sesión con el usuario creado si no se cumple la política de complejidad definida en el sistema: un mínimo de 14 caracteres que combinen dígitos, letras tanto en minúsculas como en mayúsculas y signos no alfanuméricos.

```
# grep -B1 -v ^# /etc/security/pwquality.conf
# Per : Set minclass = 4 in /etc/security/pwquality.conf
minclass = 4
# Per : Set minlen = 14 in /etc/security/pwquality.conf
minlen = 14
# Per : Set retry = 3 in /etc/security/pwquality.conf
retry = 3
```

I.1.3. Configuración del cortafuegos

El cortafuegos que trae por defecto Rocky Linux 9.2 es Firewalld y se basa en la configuración de zonas y políticas de seguridad.

- Se crea la zona redAtalaia que representa la red 10.10.10.0/24.

```
# Crear zona redAtalaia
firewall-cmd --permanent --new-zone=redAtalaia
# Política por defecto de la zona
firewall-cmd --permanent --zone=redAtalaia --set-target=DROP
# Descripción
firewall-cmd --permanent --zone=redAtalaia \
--set-description="Zona asociada a la red Atalaia 10.10.10.0/24"
# Descripción corta
firewall-cmd --permanent --zone=redAtalaia --set-short="Red Atalaia"
# Añadir interfaz de red del equipo a la zona redAtalaia
firewall-cmd --permanent --zone=redAtalaia --add-interface=enp0s3
# Descripción
firewall-cmd --permanent --zone=redAtalaia --set-description="Zona que representa \
la red 10.10.10.10/24"
# Descripción corta
firewall-cmd --permanent --zone=redAtalaia --set-short="Red Atalaia"
# Recarga de la nueva configuración manteniendo la información de estado
# de las conexiones existentes.
firewall-cmd –reload
```

- Se cambia la zona por defecto por la que se acaba de crear.

```
# Zona por defecto
firewall-cmd --set-default-zone=redAtalaia
```

- Se crean los servicios necesarios que se usarán en las reglas que permitirán su tráfico.

```
# Servicio del indexador (9200/tcp)
firewall-cmd --permanent --new-service=wazuh-indexer-api
firewall-cmd --permanent --service=wazuh-indexer-api \
--set-description="Wazuh indexer RESTful API"
firewall-cmd --permanent --service=wazuh-indexer-api \
--set-short="Wazuh indexer RESTful API"
firewall-cmd --permanent --service=wazuh-indexer-api --add-port=9200/tcp
# Servicio API RESTful del gestor Wazuh (55000/tcp)
firewall-cmd --permanent --new-service=wazuh-manager-api
firewall-cmd --permanent --service=wazuh-manager-api \
--set-description="Wazuh server RESTful API"
firewall-cmd --permanent --service=wazuh-manager-api \
--set-short="Wazuh server RESTful API"
firewall-cmd --permanent --service=wazuh-manager-api --add-port=55000/tcp
# Servicio de conexión de agentes (1514/tcp)
firewall-cmd --permanent --new-service=wazuh-manager-agent-connection
firewall-cmd --permanent --service=wazuh-manager-agent-connection \
--set-description="Agent connection service"
firewall-cmd --permanent --service=wazuh-manager-agent-connection \
--set-short="Agent connection service"
firewall-cmd --permanent --service=wazuh-manager-agent-connection --add-port=1514/tcp
# Servicio de registro de agentes (1515/tcp)
firewall-cmd --permanent --new-service=wazuh-manager-agent-enrollment
firewall-cmd --permanent --service=wazuh-manager-agent-enrollment \
--set-description="Agent enrollment service"
firewall-cmd --permanent --service=wazuh-manager-agent-enrollment \
--set-short="Agent enrollment service"
firewall-cmd --permanent --service=wazuh-manager-agent-enrollment --add-port=1515/tcp
```

- Se crea la política que va a regular el tráfico entrante desde la red 10.10.10.0/24 y que tiene como destino el host atl004s.atalaia.local.

```
# Crear política de tráfico
firewall-cmd --permanent --new-policy=redAtalaiaIn
# Política por defecto de la política
firewall-cmd --permanent --policy=redAtalaiaIn --set-target=DROP
# Zona de tráfico de entrada
firewall-cmd --permanent --policy=redAtalaiaIn --add-ingress-zone=redAtalaia
# Zona de tráfico de salida que tiene como destino el host que ejecuta firewalld
# (atl004s.atalaia.local)
firewall-cmd --permanent --policy=redAtalaiaIn --add-egress-zone=HOST
# Tráfico SSH permitido
firewall-cmd --permanent --policy=redAtalaiaIn --add-rich-rule='rule family="ipv4" \
source address="10.10.10.0/24" destination address="10.10.10.4/32" \
service name="ssh" accept'
# Tráfico ICMP permitido
firewall-cmd --permanent --policy=redAtalaiaIn --add-rich-rule='rule family="ipv4" \
source address="10.10.10.0/24" destination address="10.10.10.4/32" \
protocol value="icmp" accept'
# Tráfico al indexador Wazuh (9200/tcp)
firewall-cmd --permanent --policy=redAtalaiaIn --add-rich-rule='rule family="ipv4" \
source address="10.10.10.0/24" destination address="10.10.10.4/32" \
service name="wazuh-indexer-api" accept'
# Tráfico a la API RESTful del gestor Wazuh (55000/tcp)
firewall-cmd --permanent --policy=redAtalaiaIn --add-rich-rule='rule family="ipv4" \
source address="10.10.10.0/24" destination address="10.10.10.4/32" \
service name="wazuh-manager-api" accept'
# Tráfico web al panel de control Wazuh (443/tcp)
```

```
firewall-cmd --permanent --policy=redAtalaiaIn --add-rich-rule='rule family="ipv4" \
source address="10.10.10.0/24" destination address="10.10.10.4/32" \
service name="https" accept'
# Tráfico al servicio de conexión de agentes (1514/tcp)
firewall-cmd --permanent --policy=redAtalaiaIn --add-rich-rule='rule family="ipv4" \
source address="10.10.10.0/24" destination address="10.10.10.4/32" \
service name="wazuh-manager-agent-connection" accept'
# Tráfico al servicio de registro de agentes (1515/tcp)
firewall-cmd --permanent --policy=redAtalaiaIn --add-rich-rule='rule family="ipv4" \
source address="10.10.10.0/24" destination address="10.10.10.4/32" \
service name="wazuh-manager-agent-enrollment" accept'
# Descripción
firewall-cmd --permanent --policy=redAtalaiaIn --set-description="Política que \
filtra el tráfico que procede de la red Atalaia 10.10.10.0/24 y tiene como \
destino el HOST atl004s.atalaia.local"
# Descripción corta
firewall-cmd --permanent --policy=redAtalaiaIn --set-short="Tráfico entrante"
# Recarga de la nueva configuración manteniendo la información de estado de las
# conexiones existentes.
firewall-cmd --reload
```

- Se visualiza la configuración de la política

```
# Visualizar configuración de la zona redAtalaia
firewall-cmd --permanent --zone=redAtalaia --list-all
redAtalaia (active)
  target: DROP
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services:
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

# Visualizar configuración de la política redAtalaiaIn
firewall-cmd --permanent --policy=redAtalaiaIn --list-all
redAtalaiaIn (active)
  priority: -1
  target: DROP
  ingress-zones: redAtalaia
  egress-zones: HOST
  services:
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule family="ipv4" source address="10.10.10.0/24" destination
address="10.10.10.4/32" service name="ssh" accept
        rule family="ipv4" source address="10.10.10.0/24" destination
address="10.10.10.4/32" protocol value="icmp" accept
        rule family="ipv4" source address="10.10.10.0/24" destination
address="10.10.10.4/32" service name="wazuh-indexer-api" accept
```

```
        rule family="ipv4" source address="10.10.10.0/24" destination
address="10.10.10.4/32" service name="wazuh-manager-api" accept
        rule family="ipv4" source address="10.10.10.0/24" destination
address="10.10.10.4/32" service name="https" accept
        rule family="ipv4" source address="10.10.10.0/24" destination
address="10.10.10.4/32" service name="wazuh-manager-agent-connection" accept
        rule family="ipv4" source address="10.10.10.0/24" destination
address="10.10.10.4/32" service name="wazuh-manager-agent-enrollment" accept
# Mostrar zonas activas
firewall-cmd --get-active-zones
docker
  interfaces: docker0
redAtalaia
  interfaces: enp0s3
```

Una recarga de la configuración de FirewallD, incluso completa, puede que no interrumpa conexiones activas que se acaban de denegar. Se puede solucionar liberando la tabla de conexiones existentes, de tal forma que las prohibidas finalizan y las permitidas continúan activas.

```
# Instalación del paquete que contiene utilidades para interactuar con el sistema
# de seguimiento de conexiones de red.
dnf install conntrack-tools

# Revisión de las conexiones existentes
conntrack -L
tcp      6 87 TIME_WAIT src=10.10.10.4 dst=147.156.223.157 sport=37982 dport=80
src=147.156.223.157 dst=10.10.10.4 sport=80 dport=37982 [ASSURED] mark=0
secctx=system_u:object_r:unlabeled_t:s0 use=1
tcp      6 431999 ESTABLISHED src=10.10.10.14 dst=10.10.10.4 sport=50742 dport=22
src=10.10.10.4 dst=10.10.10.14 sport=22 dport=50742 [ASSURED] mark=0
secctx=system_u:object_r:unlabeled_t:s0 use=1
tcp      6 87 TIME_WAIT src=10.10.10.4 dst=147.156.223.157 sport=37974 dport=80
src=147.156.223.157 dst=10.10.10.4 sport=80 dport=37974 [ASSURED] mark=0
secctx=system_u:object_r:unlabeled_t:s0 use=1
tcp      6 87 TIME_WAIT src=10.10.10.4 dst=147.156.223.157 sport=37994 dport=80
src=147.156.223.157 dst=10.10.10.4 sport=80 dport=37994 [ASSURED] mark=0
secctx=system_u:object_r:unlabeled_t:s0 use=1
icmp     1 29 src=10.10.10.14 dst=10.10.10.4 type=8 code=0 id=29599 src=10.10.10.4
dst=10.10.10.14 type=0 code=0 id=29599 mark=0 secctx=system_u:object_r:unlabeled_t:s0
use=1

# Vaciar la tabla de conexiones existentes
conntrack -F
conntrack v1.4.7 (conntrack-tools): connection tracking table has been emptied.
```

I.1.4. Configuración del gestor de arranque GRUB

Rocky Linux dispone de documentación para solicitar credenciales cuando se quiere realizar modificaciones en la configuración del gestor de arranque GRUB o acceder a su consola [88]. Sin embargo, presenta el inconveniente de que está limitado a un único usuario que tiene de nombre root, por lo que se realiza la siguiente configuración que es más flexible al poder definir múltiples usuarios y utilizar nombres no utilizados de manera frecuente.

- Se genera él hash de la contraseña para el usuario de GRUB.

```
grub2-mkpasswd-pbkdf2
Introduzca la contraseña:
Reintroduzca la contraseña:
```

```
El hash PBKDF2 de su contraseña es grub.pbkdf2.sha512.10000.2B…222E6111
```

- Se define el usuario y la contraseña asociada en el fichero /etc/grub.d/40_custom.

```
cat /etc/grub.d/40_custom
#!/usr/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.

set superusers="boole"
password_pbkdf2 boole grub.pbkdf2.sha512.10000.2B…222E6111
```

- Se recrea la configuración de GRUB.

```
grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Adding boot menu entry for UEFI Firmware Settings ...
Done
ls -la /boot/grub2/grub.cfg
-rwx------. 1 root root 7332 oct 31 19:51 /boot/grub2/grub.cfg
```

I.1.5. Instalación de Docker Engine

Se opta por la instalación de Docker Engine porque se desplegará únicamente su tecnología de contenedores de código abierto mediante Docker Compose, prescindiéndose, por tanto, de las utilidades e interfaz gráfica que proporciona Docker Desktop.

- Se añade el repositorio de Docker

```
LANG=en_US.utf8 dnf config-manager --add-repo \
https://download.docker.com/linux/centos/docker-ce.repo
Adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
```

- Se instala Docker Engine, el gestor del ciclo de vida de los contenedores containerd y Docker Compose.

```
dnf install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

- Se edita el fichero /etc/docker/daemon.json para configurar el demonio dockerd [89–91].

```
cat /etc/docker/daemon.json
{
   "data-root": "/opt/docker-data",
   "builder": {
       "gc": {
       "defaultKeepStorage": "20GB",
       "enabled": true
       }
   },
   "experimental": false,
   "iptables": false
}
```

Fichero 29: daemon.json

La clave data-root indica la ruta donde se almacenarán los datos persistentes de Docker.

El objeto builder permite habilitar el recolector de basura (clave gc) cuando se llega al límite de 20GB.

La clave experimental establecida a false indica que no se habilitan las características experimentales.

La clave iptables con valor false impide que Docker Engine añada reglas de filtrado y manipulación de paquetes de red.

- Se verifica la configuración realizada

```
dockerd --validate --config-file=/etc/docker/daemon.json
configuration OK
```

- Se inicia el servicio dockerd y se configura para que lo haga automáticamente al arrancar el sistema.

```
systemctl --now enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service →
/usr/lib/systemd/system/docker.service.
```

- Se añade el usuario boole del sistema al grupo docker para que pueda utilizar Docker Engine.

```
[root@atl004s ~]# usermod -aG docker boole
…
[boole@atl004s ~]$ newgrp docker
[boole@atl004s ~]$ docker version
Client: Docker Engine - Community
 Version:           24.0.7
 API version:       1.43
 Go version:        go1.20.10
 Git commit:        afdd53b
 Built:             Thu Oct 26 09:09:13 2023
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          24.0.7
  API version:      1.43 (minimum version 1.12)
  Go version:       go1.20.10
  Git commit:       311b9ff
  Built:            Thu Oct 26 09:07:45 2023
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.24
  GitCommit:        61f9fd88f79f081d64d6fa3bb1a0dc71ec870523
 runc:
  Version:          1.1.9
  GitCommit:        v1.1.9-0-gccaecfc
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

- Se realizan ajustes y comprobaciones en Firewalld.

Docker Engine crea la zona docker en Firewalld para agrupar sus interfaces en modo puente, la cual es utilizada en la siguiente política redAtalaiaToDocker para restringir el tráfico que le llega procedente de la red Atalaia (10.10.10.0/24).

```
# Descripción de la zona docker
firewall-cmd --permanent --zone=docker --get-description
zone for docker bridge network interfaces
# Crear política de tráfico
firewall-cmd --permanent --new-policy=redAtalaiaToDocker
# Acción por defecto de la política
```

```
firewall-cmd --permanent --policy=redAtalaiaToDocker --set-target=DROP
# Zona de tráfico de entrada
firewall-cmd --permanent --policy=redAtalaiaToDocker --add-ingress-zone=redAtalaia
# Zona de tráfico de salida
firewall-cmd --permanent --policy=redAtalaiaToDocker --add-egress-zone=docker
# Visualizar configuración de la política redAtalaiaIn
firewall-cmd --permanent --policy=redAtalaiaToDocker --list-all
redAtalaiaToDocker (active)
  priority: -1
  target: DROP
  ingress-zones: redAtalaia
  egress-zones: docker
  services:
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

La restricción aplicada anteriormente para que dockerd no añada reglas de filtrado y de manipulación de paquetes provoca que la zona docker deje de estar activa en Firewalld.

```
# Mostrar zonas activas
firewall-cmd --get-active-zones
redAtalaia
  interfaces: enp0s3
```

- Se incrementa el número máximo de áreas de mapeo de memoria que un proceso puede tener al valor indicado en los requisitos necesarios para la instalación en Docker de la plataforma de seguridad Wazuh [54].

```
# Se comprueba el valor inicial del parámetro del kernel
sysctl vm.max_map_count
vm.max_map_count = 65530

# Se establece el nuevo valor
cat /etc/sysctl.d/wazuh_requirements.conf
# https://documentation.wazuh.com/current/deployment-options/docker/docker-
installation.html#requirements
vm.max_map_count = 262144

# Se aplica el cambio indicado
sysctl --system |grep -i vm.max_map_count
vm.max_map_count = 262144
```

- Se permite la apertura de puertos privilegiados a partir del 80 para los servicios.

```
# Se comprueba el valor inicial del parámetro del kernel
sysctl net.ipv4.ip_unprivileged_port_start
net.ipv4.ip_unprivileged_port_start = 1024

# Se establece el nuevo valor
cat /etc/sysctl.d/privileged-ports.conf
# https://docs.docker.com/engine/security/rootless/#exposing-privileged-ports
net.ipv4.ip_unprivileged_port_start=80

# Se aplica el cambio indicado
sysctl --system |grep -i privileged
net.ipv4.ip_unprivileged_port_start = 80
```

I.2. Instalación de las máquinas virtuales que representan a los endpoints

Se configuran dos máquinas virtuales para simular dos endpoints, uno con sistema operativo Microsoft Windows 10 y otro con Debian GNU/Linux 12. Sus instalaciones serán básicas con gran parte de sus valores por defecto para poder evaluar su configuración de seguridad con la plataforma de seguridad Wazuh.

I.2.1. Máquinas virtuales

La configuración relevante de las máquinas virtuales, que se puede modificar en caso de ser necesario, aparece recogida en la siguiente tabla.

Elemento	Configuración
BIOS EFI	Activada
Secure Boot	Activado
vCPU	2
Memoria RAM	3 GB
Memoria de vídeo	40 MB
Disco duro virtual	60 GB
Audio	Habilitado
Red	Red NAT para la 10.10.10.0/24. DHCP activado.

Tabla 27: Configuración de las máquinas virtuales para los endpoints

I.2.2. Instalación de la máquina virtual con SO Windows

En la siguiente tabla se detallan los datos más importantes acerca de la instalación de Microsoft Windows 10.

Nombre	Configuración
Regionalización	
Idioma	Español (España, internacional)
Región	España
Formato de hora y moneda	Español (España, internacional)
Teclado o método de entrada:	Español
Segunda distribución de teclado	Inglés (Estados Unidos) / Estados Unidos
Particiones	Configuradas automáticamente
Sistema	Unidad 0 Partición 1 - 100 MB
MSR (Microsoft Reserved Partition)	Unidad 0 Partición 2 - 16 MB
Principal	Unidad 0 Partición 3 - 59.9 GB
Sistema	
SO a instalar	Windows 10 Pro
Configuración para uso personal	Si
Cuenta sin conexión	Si
Usuario	boole Definir contraseña y preguntas de seguridad
Importar datos del explorador web	No
Usar ubicación	No
Encontrar dispositivo	No
Enviar datos de diagnóstico	No
Mejorar entradas manuscritas y la escritura	No
Experiencias personalizadas con datos de diagnóstico:	No
Usar el id. de publicidad:	No
Personalizar experiencia	No

Nombre	Configuración
Activar Cortana	No
Configuración de red	
Nombre de equipo	pc033w
IPv4	
Dirección IP	10.10.10.33
Máscara de subred:	255.255.255.0
Pasarela	10.10.10.1
Servidores DNS	10.10.10.1
Anexar sufijos DNS	atalaia.local
IPv6	Deshabilitado

Tabla 28: Opciones de instalación de pc033w

I.2.3. Instalación de la máquina virtual con SO Linux

La ISO de Debian GNU/Linux 12 presenta una incompatibilidad cuando se habilita la BIOS EFI, por lo que es necesario editar la entrada de la instalación experta y deshabilitar el *framebuffer* con el parámetro fb=false para que se haga en modo texto [92].

```
set background_color=black
linux    /install.amd/vmlinuz priority=low vga=788 fb=false ---
initrd   /install.amd/initrd.gz
```

Figura 41: Edición de la entrada de instalación experta en GRUB

De esta manera ya es posible realizar la instalación con los valores más relevantes indicados en la Tabla 29.

Nombre	Configuración
Regionalización	
Idioma	Español
Ubicación	España
Localización	España - es_ES.UTF-8
Localización adicional	en_US.UTF-8
Localización del sistema	es_ES.UTF-8
Teclado	Español
Zona horaria	Península
Reloj en hora UTC	Si
Red	
Nombre de la máquina	pc034l
Nombre de dominio	atalaia.local
IPv4	Configuración manual
Dirección IP	10.10.10.34
Máscara de subred	255.255.255.0
Pasarela	10.10.10.1
Servidores DNS	10.10.10.1
IPv6	Deshabilitado
Servidor NTP	0.debian.pool.ntp.org
Réplica de red	deb.debian.org Protocolo http / Sin uso de proxy Uso de firmware no libre Uso de repositorios de fuentes
Usuarios y contraseñas	
Permitir acceso como root	No
Usuario	boole (Se define su contraseña)

Nombre	Configuración
Particiones	Configuradas automáticamente Guiado – utilizar todo el disco Todos los ficheros en una partición
sda1 (512M)	/boot/efi
sda2 (58,5G)	/
sda3 (976M)	swap
Sistema	
Núcleo a instalar	linux-image-amd64
Controladores en initrd	dirigido (sólo los controladores necesarios)
Tipo de Actualizaciones	de seguridad (security.debian.org) de la distribución
Actualizaciones automáticas	No
Encuesta de paquetes	No
Selección de paquetes	Entorno de escritorio Debian MATE Utilidades estándar del sistema
GRUB	
Forzar instalación en medios removibles EFI	No
Actualizar las variables NVRAM	Si
Detectar y arrancar otros SO	No

Tabla 29: Opciones de instalación de pc034l

Finalmente, se instalan los siguientes paquetes en la máquina virtual pc034l porque son necesarios para el capítulo 5 de casos de uso que se proponen en este Trabajo.

```
root@pc034l:~# apt install ssh firewalld rsyslog
```

Paquete	Descripción y finalidad
ssh	Servidor OpenSSH que permite gestionar la máquina de manera remota.
firewalld	Servicio de cortafuegos que proporciona las herramientas necesarias para gestionar las reglas de seguridad que se aplicarán.
rsyslog	Sistema de registro del sistema basado en mensajes en formato syslog, necesario para que la plataforma de Wazuh obtenga los eventos necesarios que hacen saltar las reglas configuradas de fábrica.

Tabla 30: Paquetes instalados en la máquina virtual pc034l

I.3. Conectividad de red entre las máquinas virtuales del entorno

Se utiliza el fichero local *hosts* de cada SO de la máquina virtual para realizar la resolución de nombres del entorno configurado con Oracle VM VirtualBox.

```
# Entradas añadidas al fichero hosts
# Ruta en Windows: C:\Windows\System32\drivers\etc\hosts
# Ruta en GNU/Linux: /etc/hosts

# Red Atalaia
10.10.10.4  atl004s.atalaia.local indexer.atalaia.local manager.atalaia.local dashboard.atalaia.local
10.10.10.33 pc033w.atalaia.local
10.10.10.34 pc034l.atalaia.local
```

Finalmente, se verifica la conectividad entre los distintos hosts con el comando ping, teniendo en cuenta que ha sido necesario añadir una regla de entrada el cortafuegos de Windows 10 para permitir el tráfico ICMP.

- Se configura una regla de entrada de tipo personalizada en Windows Defender Firewall con seguridad avanzada.

Elemento	Configuración
Nombre	Tráfico ICMP desde la red Atalaia
Programas	Todos los programas
Dirección IP remota	10.10.10.0/24
Dirección IP local	Cualquiera dirección IP
Protocolo	ICMPv4
Acción	Permitir la conexión

Tabla 31: Tráfico ICMP permitido en Windows 10

- Se verifica la conectividad entre hosts.

```
# Conectividad desde atl004s.atalaia.local
[boole@atl004s ~]$ ping -c 1 pc033w.atalaia.local
PING pc033w.atalaia.local (10.10.10.33) 56(84) bytes of data.
64 bytes from pc033w.atalaia.local (10.10.10.33): icmp_seq=1 ttl=128 time=0.328 ms

[boole@atl004s ~]$ ping -c 1 pc034l.atalaia.local
PING pc034l.atalaia.local (10.10.10.34) 56(84) bytes of data.
64 bytes from pc034l.atalaia.local (10.10.10.34): icmp_seq=1 ttl=64 time=0.347 ms

[boole@atl004s ~]$ ping -c 1 www.uoc.edu
PING d3h7m5mv8dd7fj.cloudfront.net (52.84.66.17) 56(84) bytes of data.
64 bytes from server-52-84-66-17.mad51.r.cloudfront.net (52.84.66.17): icmp_seq=1 ttl=244
time=12.0 ms

# Conectividad desde pc033w.atalaia.local
C:\Users\boole>ping -n 1 atl004s.atalaia.local
Haciendo ping a atl004s.atalaia.local [10.10.10.4] con 32 bytes de datos:
Respuesta desde 10.10.10.4: bytes=32 tiempo<1m TTL=64

C:\Users\boole>ping -n 1 pc034l.atalaia.local
Haciendo ping a pc034l.atalaia.local [10.10.10.34] con 32 bytes de datos:
Respuesta desde 10.10.10.34: bytes=32 tiempo<1m TTL=64

C:\Users\boole>ping -n 1 www.uoc.edu
Haciendo ping a d3h7m5mv8dd7fj.cloudfront.net [52.84.66.93] con 32 bytes de datos:
Respuesta desde 52.84.66.93: bytes=32 tiempo=15ms TTL=244

# Conectividad desde pc034l.atalaia.local
boole@pc034l:~$ ping -c 1 atl004s.atalaia.local
PING atl004s.atalaia.local (10.10.10.4) 56(84) bytes of data.
64 bytes from atl004s.atalaia.local (10.10.10.4): icmp_seq=1 ttl=64 time=0.373 ms

boole@pc034l:~$ ping -c 1 pc033w.atalaia.local
PING pc033w.atalaia.local (10.10.10.33) 56(84) bytes of data.
64 bytes from pc033w.atalaia.local (10.10.10.33): icmp_seq=1 ttl=128 time=0.249 ms

boole@pc034l:~$ ping -c 1 www.uoc.edu
PING d3h7m5mv8dd7fj.cloudfront.net (52.84.66.30) 56(84) bytes of data.
64 bytes from server-52-84-66-30.mad51.r.cloudfront.net (52.84.66.30): icmp_seq=1 ttl=244
time=14.0 ms
```

Anexo II: Script para la generación de certificados para la plataforma

El script mk-atalaia-certificates.sh permite generar los distintos certificados necesarios para su uso en la plataforma de seguridad Wazuh, entre los cuales se encuentran el de la Autoridad de Certificación (CA), los necesarios para los servicios de los componentes centrales, los asociados a los endpoints y el certificado admin.atalaia.local adicional para realizar tareas relacionadas con la gestión y la seguridad.

```
#!/bin/bash

# Script mk-atalaia-certificates.sh
# Version 202311211914

# Variables

# Curva ECDSA
ECDSA_CURVE="secp256k1"

# Tamaño de clave RSA en número de bits
RSA_KEY_SIZE="3072"

# Validez del certificado en número de días
VALIDITY_DAYS="365"

# Sufijo DNS
DNS_SUFFIX="atalaia.local"

# Parte común del Nombre Distinguido (DN) del sujeto para todos los certificados
CERTS_SUBJECT="/C=ES/ST=A Coruña/L=Santiago de Compostela/O=Atalaia/OU=ATIC"

# Nombre común (CN) de certificado raíz
ROOT_CA="atalaia-root-ca"

# Ruta en donde se crean los certificados y las claves privadas asociadas
CERTIFICATES_PATH="$(pwd)/backups/certs"

# Ruta con el certificado y clave privada del certificado raíz
ROOT_CERTIFICATES_PATH="$(pwd)/backups/root-certs"

# Ruta con los certificados y claves privadas necesarios para Wazuh
WZH_PATH="$(pwd)/config/certs"

# Nombres comunes (CN) de componentes centrales de la plataforma Wazuh
WZH_HOST=("admin" "indexer" "manager" "dashboard")

# Funciones

# Función de ayuda
function help_command () {
    echo
    echo "Modo de empleo:"
    echo " $(basename $0) {rsa|ecdsa} root_ca"
    echo " $(basename $0) {rsa|ecdsa} file"
    echo " $(basename $0) help"
    echo
    echo "Claves criptográficas:"
    echo " rsa: Usar claves RSA para generar los certificados."
    echo " ecdsa: Usar claves ECDSA para generar los certificados."
    echo
    echo "Fichero de entrada:"
    echo " file: Crear certificados para los hosts a partir del archivo indicado."
    echo "       Debe estar ubicado en la ruta del script $(basename $0)."
    echo "       Cada línea tiene una asignación de nombre de host y de dirección IP"
```

```
    echo "       separados por espacios o tabuladores."
    echo
    echo "Certificado raíz de CA:"
    echo " root_ca: Crear certificado para la autoridad de certificación raíz."
    echo
    echo "Ayuda:"
    echo " help: Solicitar ayuda."
    echo
}

# Crear los directorios necesarios
function create_dirs () {
    if [ "${ROOT_CA_PARAMETER}" == "root_ca" ]; then
        if [ ! -d "${ROOT_CERTIFICATES_PATH}" ]; then
            echo -n "El directorio con el certificado raíz y su clave privada no existe."
            echo " Creándolo..."
            echo "Ruta: ${ROOT_CERTIFICATES_PATH}"
            mkdir -p ${ROOT_CERTIFICATES_PATH}
        else
            echo "El directorio ${ROOT_CERTIFICATES_PATH} existe."
            echo "Creando backup en ${ROOT_CERTIFICATES_PATH}-$(date +%Y%m%d%H%M%S)..."
            mv ${ROOT_CERTIFICATES_PATH} ${ROOT_CERTIFICATES_PATH}-$(date +%Y%m%d%H%M%S)
            mkdir -p ${ROOT_CERTIFICATES_PATH}
        fi
    elif [ -z ${ROOT_CA_PARAMETER} ]; then
        if [ ! -d "${CERTIFICATES_PATH}" ]; then
            echo "Creando directorio ${CERTIFICATES_PATH} para generar certificados..."
            mkdir -p ${CERTIFICATES_PATH}
        else
            echo -n "El directorio ${CERTIFICATES_PATH} existe."
            echo " Creando backup en ${CERTIFICATES_PATH}-$(date +%Y%m%d%H%M%S)"
            mv "${CERTIFICATES_PATH}" "${CERTIFICATES_PATH}-$(date +%Y%m%d%H%M%S)"
            mkdir -p ${CERTIFICATES_PATH}
        fi
    fi

    if [ ! -d "${WZH_PATH}" ]; then
      echo -n "El directorio con certificados para la plataforma de seguridad Wazuh"
      echo " no existe. Creándolo..."
      echo "Ruta: ${WZH_PATH}"
      mkdir -p ${WZH_PATH}
    fi
}

# ROOT CA
function create_root_ca () {
    echo "Creando certificado raíz ${CERTS_SUBJECT}/CN=${ROOT_CA}.${DNS_SUFFIX}..."
    # Clave privada
    if [ ${SELECTED_ALGORITHM} == "ecdsa" ]; then
    openssl ecparam -name ${ECDSA_CURVE} -genkey -noout \
    -out ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.key
    elif [ ${SELECTED_ALGORITHM} == "rsa" ];then
    openssl genrsa -out ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.key ${RSA_KEY_SIZE}
    fi
    # Certificado
    openssl req -x509 -sha512 -utf8 -noenc -days ${VALIDITY_DAYS} \
    -subj "${CERTS_SUBJECT}/CN=${ROOT_CA}.${DNS_SUFFIX}" \
    -key ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.key \
    -out ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.crt 2> /dev/null
    # Permisos
    chmod 400 ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.{crt,key}
}

# Certificados
function create_certificates () {
```

```
    # Acceso a los arrays WZH_HOST y WZH_IP mediante índices
    while read endpointName endpointIP;
    #for CERTIFICATE_INDEX in ${!WZH_HOST[@]}
    do
        CERT_OPTIONS="[CERT_CONFIG]\n\
        subjectAltName = DNS:${endpointName}.${DNS_SUFFIX},\
        IP:${endpointIP}\n\
        basicConstraints = CA:FALSE\n\
        keyUsage = nonRepudiation, digitalSignature, keyEncipherment, dataEncipherment"

        echo -n "Creando certificado ${CERTS_SUBJECT}"
        echo "/CN=${endpointName}.${DNS_SUFFIX}..."
        # Clave privada
        if [ ${SELECTED_ALGORITHM} == "ecdsa" ]; then
        openssl ecparam -name ${ECDSA_CURVE} -genkey -noout \
        -out ${CERTIFICATES_PATH}/${endpointName}-temp.key
        elif [ ${SELECTED_ALGORITHM} == "rsa" ];then
        openssl genrsa \
        -out ${CERTIFICATES_PATH}/${endpointName}-temp.key ${RSA_KEY_SIZE}
        fi
        # Clave privada PKCS#8
        openssl pkcs8 -inform PEM -outform PEM \
        -in ${CERTIFICATES_PATH}/${endpointName}-temp.key \
        -topk8 -nocrypt -v1 PBE-SHA1-3DES \
        -out ${CERTIFICATES_PATH}/${endpointName}.key
        # CSR
        openssl req -new \
        -subj "${CERTS_SUBJECT}/CN=${endpointName}.${DNS_SUFFIX}" \
        -reqexts CERT_CONFIG -config <(echo -e "${CERT_OPTIONS}") \
        -utf8 -key ${CERTIFICATES_PATH}/${endpointName}.key \
        -out ${CERTIFICATES_PATH}/${endpointName}.csr
        # Certificado
        openssl x509 -req -in ${CERTIFICATES_PATH}/${endpointName}.csr \
        -CA ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.crt \
        -CAkey ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.key -copy_extensions copy \
        -CAcreateserial -sha512 -days 365 \
        -out ${CERTIFICATES_PATH}/${endpointName}.crt 2> /dev/null
    done < ${HOSTS_FILE}
}

# Mostar datos del certificado raíz de la Autoridad de Certificción (CA)
function show_root_CA_certificate () {
    echo -e "\n---- Certificado ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.crt----"
    openssl x509 -in ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.crt -noout -subject \
    -ext basicConstraints,subjectAltName,keyUsage -dates -nameopt oneline,-esc_msb
}

# Mostar datos de los certificados
function show_certificates () {
    while read endpointName endpointIP;
    do
        echo -e "\n---- Certificado ${endpointName}.crt----"
        openssl x509 -in ${CERTIFICATES_PATH}/${endpointName}.crt -noout -subject \
        -ext basicConstraints,subjectAltName,keyUsage -dates -nameopt oneline,-esc_msb
    done < ${HOSTS_FILE}
    echo "----"
}

# Copiar certificados y claves privadas a ${WZH_PATH} para despliegue en Docker Compose
# No se copia clave privada del certificado ROOT CA
function copy_certificates () {
    if [ "${ROOT_CA_PARAMETER}" == "root_ca" ] &&
    [ -f ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.crt ]; then
        echo "Copiando certificado raíz de la autoridad de certificación (CA)..."
        echo -e "\tClave privada: ${ROOT_CA}.crt"
```

```
        echo -e "\tDe: ${ROOT_CERTIFICATES_PATH}/"
        echo -e "\tA: ${WZH_PATH}/"
        cp -f ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.crt ${WZH_PATH}/
        chmod 400 ${WZH_PATH}/${ROOT_CA}.crt
    elif [ -z "${ROOT_CA_PARAMETER}" ]; then
        # Certificados y claves privadas de los hosts
        for CERTIFICATE in ${WZH_HOST[@]}
        do
            if [ -f ${CERTIFICATES_PATH}/${CERTIFICATE}.crt ]; then
                echo "Copiando certificado ${CERTIFICATE}.crt:"
                echo -e "\tDe: ${CERTIFICATES_PATH}/"
                echo -e "\tA: ${WZH_PATH}/"
                cp -f ${CERTIFICATES_PATH}/${CERTIFICATE}.crt ${WZH_PATH}/
                chmod 400 ${WZH_PATH}/${CERTIFICATE}.crt
            fi
            if [ -f ${CERTIFICATES_PATH}/${CERTIFICATE}.key ]; then
                echo "Copiando clave privada ${CERTIFICATE}.key:"
                echo -e "\tDe: ${CERTIFICATES_PATH}/"
                echo -e "\tA: ${WZH_PATH}/"
                cp -f ${CERTIFICATES_PATH}/${CERTIFICATE}.key ${WZH_PATH}
                chmod 400 ${WZH_PATH}/${CERTIFICATE}.key
            fi
        done
    fi
    # Permisos
    # chmod 400 ${WZH_PATH}/*
}

# Ejecución del script

umask 7077

# Condiciones de ejecución
if [ $# != 2  ]; then
    help_command
else
    SELECTED_ALGORITHM="$1"
    case "${SELECTED_ALGORITHM}" in
        rsa|ecdsa)
            if [ $2 != "root_ca" ] && [ ! -f $(pwd)/$2 ]; then
                echo
                echo "ERROR: No existe el fichero de hosts indicado: $(pwd)/$2"
                echo
            else
                echo -en "\nUso de claves "
                echo "$(echo "${SELECTED_ALGORITHM}" | tr [:lower:] [:upper:])"
                echo
                if [ $2 == "root_ca" ]; then
                    ROOT_CA_PARAMETER=$2
                    create_dirs
                    create_root_ca
                    copy_certificates
                    show_root_CA_certificate
                elif [ -f $(pwd)/$2 ]; then
                    if [ -f ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.crt ] ||
                    [ -f ${ROOT_CERTIFICATES_PATH}/${ROOT_CA}.key ] ; then
                        HOSTS_FILE="$2"
                        create_dirs
                        create_certificates
                        copy_certificates
                        show_certificates
                    else
                        echo -n "ERROR: No existen el certificados raíz y/o la "
                        echo "clave privada de la autoridad de certificación."
                        echo -n "Se pueden generar con el siguiente comando: "
```

```
                        echo "$(basename $0) {rsa|ecdsa} root_ca"
                        echo
                    fi
                fi
            fi
            ;;
        *) help_command;;
    esac
fi
```

Fichero 30: Script mk-atalaia-certificates.sh

El script se estructura de la siguiente manera:

- La primera línea especifica el intérprete de comandos que ejecuta el script.
- Se definen las variables que permiten alterar configuraciones asociadas con los certificados.
- La función help_command muestra la ayuda cuando se ejecuta incorrectamente el script o se solicita explícitamente.
- La función create_dirs crea los directorios necesarios en caso de que no existan.
- La función create_root_ca genera la autoridad de certificación raíz, a partir de la cual se emitirán el resto de los certificados: primero, se crea la clave privada con el algoritmo de clave pública seleccionado y, a continuación, se obtiene el certificado raíz de CA firmado con una función resumen SHA512 (SHA2-512), con sus campos en formato UTF8 y con un periodo de validez de 365 días desde su creación.
- La función create_certificates lee un fichero externo, indicado en uno de los parámetros de ejecución del script, a través de un bucle while. De cada línea se obtiene el nombre del dispositivo y su dirección IP que se utilizarán para crearle su propio certificado. Esta sección de código destaca por hacer una conversión de las claves privadas al formato PKCS#8 y por la generación de los certificados a través de sus correspondientes CSR (solicitudes de firmas de certificados). Asimismo, cada CSR incluye opciones adicionales de configuración: el nombre DNS, la dirección IP, una restricción básica que indica que no se solicita un certificado de CA y los propósitos para los que se utilizará la clave asociada.
- Las funciones show_root_CA_certificate y show_certificates muestran información relevante de los certificados de CA y de los hosts respectivamente.
- La función copy_certificates realiza copia con ajuste de permisos de los certificados y claves privadas necesarias para su despliegue posterior a través de Docker Compose. Se debe tener en cuenta que la clave privada de la CA no se copia para el despliegue ya que su único fin es la firma de los nuevos certificados que se generen, por lo que se debe mantener a buen recaudo.
- Después de la definición de las funciones, comienza la llamada de las instrucciones en función de su posición en las sentencias de control de flujo.
- Se configura la máscara de los ficheros para restringir los permisos con los que se crean los directorios y los ficheros de los certificados.
- Una sentencia if-else controla el número de parámetros recibidos durante la ejecución del script.

- Si se introducen dos parámetros cuando se ejecuta el script, una sentencia *case* verifica el primer parámetro introducido.
- Finalmente, se anidan sentencias if-else e if-elif dentro del case para verificar si hay algún tipo de error y, en caso afirmativo, se indicará mediante su mensaje de aviso correspondiente.
- El script irá llamando a las funciones definidas previamente si los parámetros introducidos son correctos.
- El script muestra la ayuda cuando el número de parámetros es distinto de dos o si el primer parámetro es diferente de rsa o ecdsa.

Las instrucciones para ejecutar el script pueden consultarse fácilmente con su ejecución sin parámetros o con la opción help_command.

```
[boole@atl004s atalaia]$ ./mk-atalaia-certificates.sh

Modo de empleo:
 mk-atalaia-certificates.sh {rsa|ecdsa} root_ca
 mk-atalaia-certificates.sh {rsa|ecdsa} file
 mk-atalaia-certificates.sh help

Claves criptográficas:
 rsa: Usar claves RSA para generar los certificados.
 ecdsa: Usar claves ECDSA para generar los certificados.

Fichero de entrada:
 file: Crear certificados para los hosts a partir del archivo indicado.
       Debe estar ubicado en la ruta del script mk-atalaia-certificates.sh.
       Cada línea tiene una asignación de nombre de host y de dirección IP
       separados por espacios o tabuladores.

Certificado raíz de CA:
 root_ca: Crear certificado para la autoridad de certificación raíz.

Ayuda:
 help: Solicitar ayuda.
```

Los pasos necesarios para generar tanto los certificados como sus claves privadas asociadas se pueden resumir en:

- Creación del certificado de la autoridad de certificación:

```
[boole@atl004s atalaia]$ ./mk-atalaia-certificates.sh rsa root_ca

Uso de claves RSA

El directorio /opt/atalaia/backups/root-certs existe.
Creando backup en /opt/atalaia/backups/root-certs-20231124210132...
Creando certificado raíz /C=ES/ST=A Coruña/L=Santiago de
Compostela/O=Atalaia/OU=ATIC/CN=atalaia-root-ca.atalaia.local...
Copiando certificado raíz de la autoridad de certificación (CA)...
       Clave privada: atalaia-root-ca.crt
       De: /opt/atalaia/backups/root-certs/
       A: /opt/atalaia/config/certs/

---- Certificado /opt/atalaia/backups/root-certs/atalaia-root-ca.crt----
subject=C = ES, ST = A Coruña, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN =
atalaia-root-ca.atalaia.local
X509v3 Basic Constraints: critical
```

```
    CA:TRUE
notBefore=Nov 24 20:01:34 2023 GMT
notAfter=Nov 23 20:01:34 2024 GMT
```

- Fichero que contiene los datos de nombre común (CN) y de dirección IP que se usan para generar los certificados necesarios.

```
[boole@atl004s atalaia]$ cat certificados_atalaia.txt
admin           10.10.10.4
indexer         10.10.10.4
manager         10.10.10.4
dashboard       10.10.10.4
pc033w          10.10.10.33
pc034l          10.10.10.34
```

- Generación de los certificados indicados en el fichero anterior.

```
[boole@atl004s atalaia]$ ./mk-atalaia-certificates.sh rsa certificados_atalaia.txt

Uso de claves RSA

El directorio /opt/atalaia/backups/certs existe. Creando backup en
/opt/atalaia/backups/certs-20231124210140
Creando certificado /C=ES/ST=A Coruña/L=Santiago de
Compostela/O=Atalaia/OU=ATIC/CN=admin.atalaia.local...
Creando certificado /C=ES/ST=A Coruña/L=Santiago de
Compostela/O=Atalaia/OU=ATIC/CN=indexer.atalaia.local...
Creando certificado /C=ES/ST=A Coruña/L=Santiago de
Compostela/O=Atalaia/OU=ATIC/CN=manager.atalaia.local...
Creando certificado /C=ES/ST=A Coruña/L=Santiago de
Compostela/O=Atalaia/OU=ATIC/CN=dashboard.atalaia.local...
Creando certificado /C=ES/ST=A Coruña/L=Santiago de
Compostela/O=Atalaia/OU=ATIC/CN=pc033w.atalaia.local...
Creando certificado /C=ES/ST=A Coruña/L=Santiago de
Compostela/O=Atalaia/OU=ATIC/CN=pc034l.atalaia.local...
Copiando certificado admin.crt:
        De: /opt/atalaia/backups/certs/
        A: /opt/atalaia/config/certs/
Copiando clave privada admin.key:
        De: /opt/atalaia/backups/certs/
        A: /opt/atalaia/config/certs/
Copiando certificado indexer.crt:
        De: /opt/atalaia/backups/certs/
        A: /opt/atalaia/config/certs/
Copiando clave privada indexer.key:
        De: /opt/atalaia/backups/certs/
        A: /opt/atalaia/config/certs/
Copiando certificado manager.crt:
        De: /opt/atalaia/backups/certs/
        A: /opt/atalaia/config/certs/
Copiando clave privada manager.key:
        De: /opt/atalaia/backups/certs/
        A: /opt/atalaia/config/certs/
Copiando certificado dashboard.crt:
        De: /opt/atalaia/backups/certs/
        A: /opt/atalaia/config/certs/
Copiando clave privada dashboard.key:
        De: /opt/atalaia/backups/certs/
        A: /opt/atalaia/config/certs/

---- Certificado admin.crt----
```

```
subject=C = ES, ST = A Coruña, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN =
admin.atalaia.local
X509v3 Subject Alternative Name:
    DNS:admin.atalaia.local, IP Address:10.10.10.4
X509v3 Basic Constraints:
    CA:FALSE
X509v3 Key Usage:
    Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
notBefore=Nov 24 20:01:40 2023 GMT
notAfter=Nov 23 20:01:40 2024 GMT

---- Certificado indexer.crt----
subject=C = ES, ST = A Coruña, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN =
indexer.atalaia.local
X509v3 Subject Alternative Name:
    DNS:indexer.atalaia.local, IP Address:10.10.10.4
X509v3 Basic Constraints:
    CA:FALSE
X509v3 Key Usage:
    Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
notBefore=Nov 24 20:01:42 2023 GMT
notAfter=Nov 23 20:01:42 2024 GMT

---- Certificado manager.crt----
subject=C = ES, ST = A Coruña, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN =
manager.atalaia.local
X509v3 Subject Alternative Name:
    DNS:manager.atalaia.local, IP Address:10.10.10.4
X509v3 Basic Constraints:
    CA:FALSE
X509v3 Key Usage:
    Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
notBefore=Nov 24 20:01:43 2023 GMT
notAfter=Nov 23 20:01:43 2024 GMT

---- Certificado dashboard.crt----
subject=C = ES, ST = A Coruña, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN =
dashboard.atalaia.local
X509v3 Subject Alternative Name:
    DNS:dashboard.atalaia.local, IP Address:10.10.10.4
X509v3 Basic Constraints:
    CA:FALSE
X509v3 Key Usage:
    Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
notBefore=Nov 24 20:01:43 2023 GMT
notAfter=Nov 23 20:01:43 2024 GMT

---- Certificado pc033w.crt----
subject=C = ES, ST = A Coruña, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN =
pc033w.atalaia.local
X509v3 Subject Alternative Name:
    DNS:pc033w.atalaia.local, IP Address:10.10.10.33
X509v3 Basic Constraints:
    CA:FALSE
X509v3 Key Usage:
    Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
notBefore=Nov 24 20:01:44 2023 GMT
notAfter=Nov 23 20:01:44 2024 GMT

---- Certificado pc034l.crt----
```

```
subject=C = ES, ST = A Coruña, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN =
pc034l.atalaia.local
X509v3 Subject Alternative Name:
    DNS:pc034l.atalaia.local, IP Address:10.10.10.34
X509v3 Basic Constraints:
    CA:FALSE
X509v3 Key Usage:
    Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
notBefore=Nov 24 20:01:45 2023 GMT
notAfter=Nov 23 20:01:45 2024 GMT
----
```

Anexo III: Fichero docker-compose.yml

La configuración del fichero necesario para el despliegue de la plataforma de seguridad de Wazuh a través de Docker Compose consta de los siguientes elementos relevantes.

- Se disponen de dos secciones de nivel superior: *services*, para definir los servicios wazuh-indexer, wazuh-manager y wazuh-dashboard, y *volumes*, para declarar los volúmenes a utilizar junto a sus descripciones. Todos los elementos de configuración que se citan a continuación están asociados a los distintos servicios definidos dentro de services.
- La sección de construcción de cada una de las imágenes, build, dispone del atributo dockerfile_inline con las instrucciones necesarias para la personalización de cada una de las imágenes necesarias para cada uno de los servicios definidos.

Servicio	Definición dockerfile_inline
wazuh-indexer	▪ Se obtiene la imagen base wazuh-indexer. ▪ Se establece el usuario root. ▪ Se instalan los paquetes iproute2, para disponer de la utilidad ss que permita consultar los puertos abiertos, iputils-ping, para disponer del comando ping, y curl, para comprobar el acceso a servicios HTTPS. ▪ Se ajustan los permisos de ficheros y directorios parar reducir las advertencias que se muestran al iniciar el servicio. ▪ Se continúa la ejecución de la imagen como el usuario wazuh-indexer.
wazuh-manager	▪ Se obtiene la imagen base wazuh-manager y se ejecuta su punto de entrada o entrypoint /init. ▪ Se añade el script cp-filebeat-files.sh con los permisos adecuados para copiar los archivos necesarios para el funcionamiento del servicio Filebeat. ▪ Se instalan los paquetes iproute2, para disponer de la utilidad ss que permita consultar los puertos abiertos, y curl, para comprobar el acceso a servicios HTTPS. ▪ Se edita el archivo /etc/cont-init.d/0-wazuh-init para añadir la llamada al script cp-filebeat-files.sh si no existe.
wazuh-dashboard	▪ Se obtiene la imagen base wazuh-dashboard. ▪ Se establece el usuario root. ▪ Se edita el script de punto de entrada /entrypoint.sh para evitar que se actualice el keystore debido a que se proporciona una versión inicial preconfigurada. ▪ Se instalan los paquetes iproute2, para disponer de la utilidad ss que permita consultar los puertos abiertos, y curl, para comprobar el acceso a servicios HTTPS. ▪ Se continúa la ejecución de la imagen como el usuario wazuh-dashboard.

Tabla 32: Configuración Dockerfile para cada uno de los servicios

- Otras propiedades definidas en la subsección build son network y tags, que establecen respectivamente el tipo de red y las etiquetas para la generación de las imágenes. Otra propiedad disponible es context que define la ubicación relativa por defecto para ficheros Dockerfile, aunque no son utilizados en el diseño.
- Las propiedades container_name y hostname definen el nombre del contenedor y el nombre de host utilizado por el servicio respectivamente.
- El nombre del dominio para cada uno de los servicios viene establecido por el elemento domainname, que en este caso obtiene su valor de la variable ${DNS_SUFFIX} definida en el fichero .env.

- El apartado extra_hosts permite realizar la asignación de nombres de hosts a direcciones IP para no depender de un servidor DNS externo.
- El ajuste restart define la política de reinicio a always, de tal forma que es posible detener el servicio manualmente y asegurar su inició automático cuando lo hace el host anfitrión Docker [93].
- La sección ulimits define los límites de recursos para el contenedor, donde un valor establecido de -1 indica que no lo tiene.
- Los elementos environment definen las variables utilizadas en el contenedor de cada uno de los servicios, mientras que en la sección volumes se establecen los volúmenes y montajes de tipo bind.
- El servicio wazuh-manager necesita realizar los montajes de tipo bind en dos directorios específicos, /filebeat-config-mount/ y /wazuh-config-mount/, para que se preconfigure y funcione correctamente.
    - /wazuh-config-mount/: Directorio incluido por defecto por Wazuh para copiar la estructura de directorios que contiene en /var/ossec.
    - /filebeat-config-mount/: Adaptación realizada con la configuración Dockerfile indicada anteriormente para realizar la preconfiguración del servicio Filebeat: se copian sus archivos a los destinos indicados por el script script cp-filebeat-files.sh.
- La sección healthcheck establece el comando necesario para realizar comprobaciones periódicas que permitan determinar el estado operativo (healthy) del servicio, de tal forma que se necesitan 5 comprobaciones fallidas fuera del periodo de inicio definido por la propiedad start_period para considerarlo como no operativo (unhealthy), ya sea por error o falta de respuesta en 10 o más segundos.
- Los servicios wazuh-manager y wazuh-dashboard utilizan depends_on para establecer las dependencias con otros servicios antes de su inicio: wazuh-manager depende de wazuh-indexer mientras que wazuh-dashboard lo hace tanto de wazuh-indexer como de wazuh-manager.
- El elemento labels dentro de cada uno de los servicios establece las etiquetas descriptivas.

```
# Wazuh App Copyright (C) 2017, Wazuh Inc. (License GPLv2)
version: '3.7'

services:
  wazuh-indexer:
    image: wazuh/wazuh-indexer:${IMAGE_VERSION}
    build:
      context: .
      dockerfile_inline: |
        FROM wazuh/wazuh-indexer:${IMAGE_VERSION}
        USER root
        RUN apt-get update ; apt-get -y install iproute2 iputils-ping curl; \
          find /usr/share/wazuh-indexer/ -type d -exec chmod 700 {} \; ; \
          find /usr/share/wazuh-indexer/ \
            -not -path "/usr/share/wazuh-indexer/bin/*" \
            -not -path "/usr/share/wazuh-indexer/jdk/bin/*" \
            -not -path "/usr/share/wazuh-indexer/performance-analyzer-rca/bin/*" \
            -not -path "/usr/share/wazuh-indexer/plugins/opensearch-security/tools/*" \
            -not -path "/usr/share/wazuh-indexer/jdk/lib/jspawnhelper" \
```

```
            -type f -exec chmod 600 {} \;
        USER wazuh-indexer
      network: host
      # tags:
        # BUILD_VERSION="$(date +%Y%m%d%H%M%S)" docker compose build
        # - "wzh-indexer:${BUILD_VERSION:-noVersion}"
    container_name: indexer
    hostname: indexer
    domainname: ${DNS_SUFFIX}
    extra_hosts:
      - indexer.${DNS_SUFFIX}:10.10.10.4
      - manager.${DNS_SUFFIX}:10.10.10.4
      - dashboard.${DNS_SUFFIX}:10.10.10.4
      - pc033w.${DNS_SUFFIX}:10.10.10.33
      - pc034l.${DNS_SUFFIX}:10.10.10.34
    restart: always
    network_mode: "host"
    environment:
      - "OPENSEARCH_JAVA_OPTS=-Xms512m -Xmx512m"
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 65536
        hard: 65536
    volumes:
      # Configuración de zona horaria utilizando la configuración del host local
      - /etc/localtime:/etc/localtime:ro
      - /etc/timezone:/etc/timezone:ro
      # Volúmenes
      - wazuh-indexer-data:/var/lib/wazuh-indexer
      # Certificados
      - ./config/certs/atalaia-root-ca.crt:/usr/share/wazuh-indexer/certs/root-ca.pem:ro
      - ./config/certs/indexer.key:/usr/share/wazuh-indexer/certs/wazuh.indexer.key:ro
      - ./config/certs/indexer.crt:/usr/share/wazuh-indexer/certs/wazuh.indexer.pem:ro
      - ./config/certs/admin.crt:/usr/share/wazuh-indexer/certs/admin.pem:ro
      - ./config/certs/admin.key:/usr/share/wazuh-indexer/certs/admin-key.pem:ro
      # Ficheros de configuración
      - ./config/wazuh_indexer/wazuh.indexer.yml:/usr/share/wazuh-indexer/opensearch.yml
      - ./config/wazuh_indexer/internal_users.yml:/usr/share/wazuh-indexer/opensearch-
security/internal_users.yml
    healthcheck:
      test: curl --cacert /usr/share/wazuh-indexer/certs/root-ca.pem -s
https://indexer.${DNS_SUFFIX}:9200 >/dev/null || exit 1
      interval: 30s # First run after the container is started
      timeout: 10s  # Timeout of every check
      start_period: 15s # Container initialization time
      retries: 5    # Retries os consecutive failure
    labels:
      local.uoc.tfg.description: "Wazuh Indexer"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"

  wazuh-manager:
    image: wazuh/wazuh-manager:${IMAGE_VERSION}
    build:
      context: .
      dockerfile_inline: |
        FROM wazuh/wazuh-manager:${IMAGE_VERSION}
        COPY --chown=0:0 --chmod=700 ./config/wazuh_manager/cp-filebeat-files.sh /bin/cp-
filebeat-files.sh
        RUN apt-get update; apt-get -y install iproute2 curl; \
        if [[ -z "$(sed -n "/bin\/cp-filebeat-files.sh/p" /etc/cont-init.d/0-wazuh-init)" ]]; \
        then echo "/bin/cp-filebeat-files.sh" >> /etc/cont-init.d/0-wazuh-init; fi
```

```
      network: host
      # tags:
        # BUILD_VERSION="$(date +%Y%m%d%H%M%S)" docker compose build
        # - "wzh-manager:${BUILD_VERSION:-noVersion}"
    container_name: manager
    hostname: manager
    domainname: ${DNS_SUFFIX}
    restart: always
    network_mode: "host"
    extra_hosts:
      - indexer.${DNS_SUFFIX}:10.10.10.4
      - manager.${DNS_SUFFIX}:10.10.10.4
      - dashboard.${DNS_SUFFIX}:10.10.10.4
      - pc033w.${DNS_SUFFIX}:10.10.10.33
      - pc034l.${DNS_SUFFIX}:10.10.10.34
    environment:
      - INDEXER_URL=https://indexer.${DNS_SUFFIX}:9200
      - FILEBEAT_SSL_VERIFICATION_MODE=full
      - SSL_CERTIFICATE_AUTHORITIES=/etc/ssl/root-ca.pem
      - SSL_CERTIFICATE=/etc/ssl/filebeat.pem
      - SSL_KEY=/etc/ssl/filebeat.key
      - API_USERNAME=${MANAGER_API_USERNAME}
      - API_PASSWORD=${MANAGER_API_PASSWORD}
    volumes:
      # Configuración de zona horaria utilizando la configuración del host local
      - /etc/localtime:/etc/localtime:ro
      - /etc/timezone:/etc/timezone:ro
      # Volúmenes
      - wazuh_api_configuration:/var/ossec/api/configuration
      - wazuh_etc:/var/ossec/etc
      - wazuh_logs:/var/ossec/logs
      - wazuh_queue:/var/ossec/queue
      - wazuh_var_multigroups:/var/ossec/var/multigroups
      - wazuh_integrations:/var/ossec/integrations
      - wazuh_active_response:/var/ossec/active-response/bin
      - wazuh_agentless:/var/ossec/agentless
      - wazuh_wodles:/var/ossec/wodles
      - filebeat_etc:/etc/filebeat
      - filebeat_var:/var/lib/filebeat
      # Keystore
      - ./config/wazuh_manager/filebeat.keystore:/filebeat-config-
mount/var/lib/filebeat/filebeat.keystore
      # Certificados
      - ./config/certs/atalaia-root-ca.crt:/filebeat-config-mount/etc/ssl/root-ca.pem:ro
      - ./config/certs/manager.crt:/filebeat-config-mount/etc/ssl/filebeat.pem:ro
      - ./config/certs/manager.key:/filebeat-config-mount/etc/ssl/filebeat.key:ro
      - ./config/certs/atalaia-root-ca.crt:/wazuh-config-mount/api/configuration/ssl/root-
ca.pem:ro
      - ./config/certs/manager.crt:/wazuh-config-mount/api/configuration/ssl/api.pem:ro
      - ./config/certs/manager.key:/wazuh-config-mount/api/configuration/ssl/api.key:ro
      # Ficheros de configuración
      - ./config/wazuh_manager/wazuh_manager.conf:/wazuh-config-mount/etc/ossec.conf
      - ./config/wazuh_manager/api.yaml:/wazuh-config-mount/api/configuration/api.yaml
      - ./config/wazuh_manager/authd.pass:/wazuh-config-mount/etc/authd.pass
      - ./config/wazuh_manager/filebeat.yml:/filebeat-config-mount/etc/filebeat/filebeat.yml
    healthcheck:
      test: curl --cacert /wazuh-config-mount/api/configuration/ssl/root-ca.pem -s
https://manager.${DNS_SUFFIX}:55000 >/dev/null || exit 1
      interval: 30s # First run after the container is started
      timeout: 10s  # Timeout of every check
      start_period: 15s # Container initialization time
      retries: 5    # Retries os consecutive failure
    depends_on:
      wazuh-indexer:
        condition: service_healthy
```

```
    labels:
      local.uoc.tfg.description: "Wazuh Manager"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"

  wazuh-dashboard:
    image: wazuh/wazuh-dashboard:${IMAGE_VERSION}
    build:
      context: .
      dockerfile_inline: |
        FROM wazuh/wazuh-dashboard:${IMAGE_VERSION}
        USER root
        RUN sed -i 's/yes |/yes N |/' /entrypoint.sh ; apt-get update ; apt-get -y install
iproute2 curl
        USER wazuh-dashboard
      network: host
      # tags:
        # BUILD_VERSION="$(date +%Y%m%d%H%M%S)" docker compose build
        # - "wzh-dashboard:${BUILD_VERSION:-noVersion}"
    container_name: dashboard
    hostname: dashboard
    domainname: ${DNS_SUFFIX}
    extra_hosts:
      - indexer.${DNS_SUFFIX}:10.10.10.4
      - manager.${DNS_SUFFIX}:10.10.10.4
      - dashboard.${DNS_SUFFIX}:10.10.10.4
      - pc033w.${DNS_SUFFIX}:10.10.10.33
      - pc034l.${DNS_SUFFIX}:10.10.10.34
    restart: always
    network_mode: "host"
    environment:
      - WAZUH_API_URL=https://manager.${DNS_SUFFIX}
      - API_USERNAME=${MANAGER_API_USERNAME}
      - API_PASSWORD=${MANAGER_API_PASSWORD}
    volumes:
      # Configuración de zona horaria utilizando la configuración del host local
      - /etc/localtime:/etc/localtime:ro
      - /etc/timezone:/etc/timezone:ro
      # Certificados
      - ./config/certs/dashboard.crt:/usr/share/wazuh-dashboard/certs/wazuh-dashboard.pem:ro
      - ./config/certs/dashboard.key:/usr/share/wazuh-dashboard/certs/wazuh-dashboard-key.pem:ro
      - ./config/certs/atalaia-root-ca.crt:/usr/share/wazuh-dashboard/certs/root-ca.pem:ro
      # Keystore
      - ./config/wazuh_dashboard/opensearch_dashboards.keystore:/usr/share/wazuh-
dashboard/config/opensearch_dashboards.keystore:ro
      # Ficheros de configuración
      - ./config/wazuh_dashboard/opensearch_dashboards.yml:/usr/share/wazuh-
dashboard/config/opensearch_dashboards.yml
      - ./config/wazuh_dashboard/wazuh.yml:/usr/share/wazuh-
dashboard/data/wazuh/config/wazuh.yml
    healthcheck:
      test: curl --cacert /usr/share/wazuh-dashboard/certs/root-ca.pem -s
https://dashboard.${DNS_SUFFIX} >/dev/null || exit 1
      interval: 30s # First run after the container is started
      timeout: 10s  # Timeout of every check
      start_period: 15s # Container initialization time
      retries: 5    # Retries os consecutive failure
    depends_on:
      wazuh-indexer:
        condition: service_healthy
      wazuh-manager:
        condition: service_healthy
    labels:
      local.uoc.tfg.description: "Wazuh Dashboard"
      local.uoc.tfg.department: "ATIC"
```

```
      local.uoc.tfg.organization: "Atalaia"

volumes:
  wazuh_api_configuration:
    labels:
      local.uoc.tfg.description: "/var/ossec/api/configuration"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  wazuh_etc:
    labels:
      local.uoc.tfg.description: "/var/ossec/etc"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  wazuh_logs:
    labels:
      local.uoc.tfg.description: "/var/ossec/logs"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  wazuh_queue:
    labels:
      local.uoc.tfg.description: "/var/ossec/queue"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  wazuh_var_multigroups:
    labels:
      local.uoc.tfg.description: "/var/ossec/var/multigroups"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  wazuh_integrations:
    labels:
      local.uoc.tfg.description: "/var/ossec/integrations"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  wazuh_active_response:
    labels:
      local.uoc.tfg.description: "/var/ossec/active-response/bin"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  wazuh_agentless:
    labels:
      local.uoc.tfg.description: "/var/ossec/agentless"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  wazuh_wodles:
    labels:
      local.uoc.tfg.description: "/var/ossec/wodles"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  filebeat_etc:
    labels:
      local.uoc.tfg.description: "/etc/filebeat"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  filebeat_var:
    labels:
      local.uoc.tfg.description: "/var/lib/filebeat"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
  wazuh-indexer-data:
    labels:
      local.uoc.tfg.description: "/var/lib/wazuh-indexer"
      local.uoc.tfg.department: "ATIC"
      local.uoc.tfg.organization: "Atalaia"
```

Fichero 31: Fichero docker-compose.yml

Anexo IV: Ficheros para el despliegue del indexador

La configuración del indexador de Wazuh se realiza a partir de los siguientes dos ficheros dentro del directorio /opt/atalaia/config/ [94–97].

```
./config/wazuh_indexer/
├── internal_users.yml
└── wazuh.indexer.yml
```

El fichero wazuh.indexer.yml define la configuración necesaria para el indexador.

```
network.host: _site_
cluster.name: atalaia
node.name: "indexer"
path.data: /var/lib/wazuh-indexer
path.logs: /var/log/wazuh-indexer
discovery.type: single-node
http.port: 9200
transport.tcp.port: 9300
compatibility.override_main_response_version: true
plugins.security.ssl.http.pemcert_filepath: /usr/share/wazuh-indexer/certs/wazuh.indexer.pem
plugins.security.ssl.http.pemkey_filepath: /usr/share/wazuh-indexer/certs/wazuh.indexer.key
plugins.security.ssl.http.pemtrustedcas_filepath: /usr/share/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.transport.pemcert_filepath: /usr/share/wazuh-
indexer/certs/wazuh.indexer.pem
plugins.security.ssl.transport.pemkey_filepath: /usr/share/wazuh-indexer/certs/wazuh.indexer.key
plugins.security.ssl.transport.pemtrustedcas_filepath: /usr/share/wazuh-indexer/certs/root-
ca.pem
plugins.security.ssl.http.enabled: true
plugins.security.ssl.transport.enabled: true
plugins.security.ssl.transport.enforce_hostname_verification: true
plugins.security.ssl.transport.resolve_hostname: true
plugins.security.authcz.admin_dn:
  - CN=admin.atalaia.local,OU=TFG,O=UOC,L=Santiago de Compostela,ST=A Coruña,C=ES
plugins.security.restapi.password_validation_regex: '(?=.*[A-Z])(?=.*[^a-zA-Z\d])(?=.*[0-
9])(?=.*[a-z]).{8,}'
plugins.security.restapi.password_validation_error_message: "Password must be minimum 8
characters long and must contain at least one uppercase letter, one lowercase letter, one digit,
and one special character."
plugins.security.check_snapshot_restore_write_privileges: true
plugins.security.enable_snapshot_restore_privilege: true
plugins.security.nodes_dn:
  - CN=indexer.atalaia.local,OU=TFG,O=UOC,L=Santiago de Compostela,ST=A Coruña,C=ES
plugins.security.restapi.roles_enabled:
- "all_access"
- "security_rest_api_access"
plugins.security.system_indices.enabled: true
plugins.security.system_indices.indices: [".opendistro-alerting-config", ".opendistro-alerting-
alert*", ".opendistro-anomaly-results*", ".opendistro-anomaly-detector*", ".opendistro-anomaly-
checkpoints", ".opendistro-anomaly-detection-state", ".opendistro-reports-*", ".opendistro-
notifications-*", ".opendistro-notebooks", ".opensearch-observability", ".opendistro-
asynchronous-search-response*", ".replication-metadata-store"]
plugins.security.allow_default_init_securityindex: true
cluster.routing.allocation.disk.threshold_enabled: true
cluster.routing.allocation.disk.watermark.low: 3gb
cluster.routing.allocation.disk.watermark.high: 2gb
cluster.routing.allocation.disk.watermark.flood_stage: 1gb
```

Fichero 32: wazuh.indexer.yml

Esta configuración destaca por definir los siguientes ajustes.

- Se utiliza la dirección IP asignada al contenedor utilizando el valor _site_ en network.host.

- Se establecen los nombres del clúster y del nodo junto con las rutas de almacenamiento de datos y logs.
- Se define un clúster formado por un único nodo con el valor single-node en la propiedad discovery.type.
- La propiedad http.port define el puerto 9200 para la comunicación HTTP externa con la API RESTful del indexador, mientras que la configuración de transport.tcp.port define el puerto 9300 para la comunicación interna entre los nodos del clúster, aunque en este caso no se aplica al disponer de un único nodo.
- El ajuste de compatibility.override_main_response_version establecido a true es para mantener la compatibilidad con clientes antiguos que verifican la existencia de una versión concreta en el indexador.
- El conjunto de propiedades asociadas a plugins.security.ssl.http y plugins.security.ssl.transport establecen las rutas de los certificados y claves privadas que cifrarán el tráfico en las distintas comunicaciones a través de los puertos establecidos anteriormente.
- Se habilita el tráfico HTTPS tanto para la API RESTful como para la comunicación entre nodos del clúster a través de las propiedades plugins.security.ssl.http.enabled y plugins.security.ssl.transport.enabled.
- Las propiedades enforce_hostname_verification y resolve_hostname asociadas con plugins.security.ssl.transport están habilitadas para comprobar la resolución del nombre de los nodos del clúster y verificar que coincide con el que aparece en sus certificados.
- Se ajusta el nombre distinguido del certificado (DN) que tendrá asignados privilegios de administración a través del valor definido en plugins.security.authcz.admin_dn.
- A través de la opción password_validation_regex de plugins.security.restapi se verifica la política de contraseñas, su complejidad y número mínimo de caracteres, cuando se crea o modifica a través de la API REST del indexador. En el caso que no cumplirse, se mostrará el error definido en la propiedad plugins.security.restapi.password_validation_error_message.
- La propiedad plugins.security.check_snapshot_restore_write_privileges definida a true comprueba si el usuario dispone de los privilegios de crear índices o de añadir documentos JSON al índice antes de proceder a la restauración de la instantánea.
- El ajuste plugins.security.enable_snapshot_restore_privilege establecido a true indica que los usuarios normales podrán restaurar instantáneas que no contengan un estado global del sistema, que no afecten al índice .opendistro_security y siempre que tengan los permisos adecuados de restauración de instantáneas, crear índices y añadir documentos JSON al índice.
- Se define el nombre común o CN del único nodo que conforma el clúster en la propiedad plugins.security.nodes_dn.
- Los roles que son necesarios para acceder a la API RESTful se establecen en la propiedad plugins.security.restapi.roles_enabled.

- Las dos propiedades asociadas a plugins.security.system_indices habilitan y definen los índices de sistema.
- Se habilita plugins.security.allow_default_init_securityindex para obtener los valores por defecto de los ajustes de seguridad si falla la creación del índice asociado al iniciar OpenSearch.
- El ajuste cluster.routing.allocation.disk.threshold_enabled está deshabilitado por defecto en la configuración proporcionada por Wazuh, pero es útil para prevenir la falta de espacio en los discos y presenta la ventaja, en entornos multimodo, de recolocar fragmentos (shards) en *hosts* con mayor capacidad de almacenamiento disponible.
- Se definen los tres niveles para controlar el bajo espacio en el disco a los valores de 3GB, 2GB y 1GB respectivamente, aunque en un entorno mononodo sólo tendrá efecto el último al no poder moverse fragmentos a otros *hosts*.
- La propiedad cluster.routing.allocation.disk.watermark.flood_stage habilita el bloqueo de escritura en los índices cuando se dispone de 1GB o menos en el disco y lo libera cuando se dispone de al menos de 2GB, valor definido en el umbral alto cluster.routing.allocation.disk.watermark.high.

El archivo internal_users.yml define los usuarios internos y los hashes de sus contraseñas, los cuales se han modificado para evitar que el indexador de Wazuh arranque con las credenciales predeterminadas.

```
---
# This is the internal user database
# The hash value is a bcrypt hash and can be generated with plugin/tools/hash.sh

_meta:
  type: "internalusers"
  config_version: 2

# Define your internal users here

## Demo users

admin:
  hash: "$2y$12$yJYt50yYA/BKrbh7VQAX3u1YmJyynvTioRdpaPAsdLhXe.s0kALMW"
  reserved: true
  backend_roles:
  - "admin"
  description: "Demo admin user"

kibanaserver:
  hash: "$2y$12$yJYt50yYA/BKrbh7VQAX3u1YmJyynvTioRdpaPAsdLhXe.s0kALMW"
  reserved: true
  description: "Demo kibanaserver user"

kibanaro:
  hash: "$2y$12$yJYt50yYA/BKrbh7VQAX3u1YmJyynvTioRdpaPAsdLhXe.s0kALMW"
  reserved: false
  backend_roles:
  - "kibanauser"
  - "readall"
  attributes:
    attribute1: "value1"
    attribute2: "value2"
    attribute3: "value3"
```

```
  description: "Demo kibanaro user"

logstash:
  hash: "$2y$12$yJYt50yYA/BKrbh7VQAX3u1YmJyynvTioRdpaPAsdLhXe.s0kALMW"
  reserved: false
  backend_roles:
  - "logstash"
  description: "Demo logstash user"

readall:
  hash: "$2y$12$yJYt50yYA/BKrbh7VQAX3u1YmJyynvTioRdpaPAsdLhXe.s0kALMW"
  reserved: false
  backend_roles:
  - "readall"
  description: "Demo readall user"

snapshotrestore:
  hash: "$2y$12$yJYt50yYA/BKrbh7VQAX3u1YmJyynvTioRdpaPAsdLhXe.s0kALMW"
  reserved: false
  backend_roles:
  - "snapshotrestore"
  description: "Demo snapshotrestore user"
```

Fichero 33: internal_users.yml

Tal y como se indica en el encabezado del fichero anterior, la generación de hashes de contraseñas se realiza con la utilidad hash.sh, incluida con OpenSearch, y en este caso se ha utilizado el mismo para todos los usuarios, aunque en un entorno en producción cada usuario del sistema debe tener sus propias credenciales.

```
[boole@atl004s atalaia]$ docker compose exec wazuh-indexer /bin/bash

wazuh-indexer@indexer:~$ OPENSEARCH_JAVA_HOME=/usr/share/wazuh-indexer/jdk bash
/usr/share/wazuh-indexer/plugins/opensearch-security/tools/hash.sh
************************************************************************
** This tool will be deprecated in the next major release of OpenSearch **
** https://github.com/opensearch-project/security/issues/1755           **
************************************************************************
[Password:]
$2y$12$yJYt50yYA/BKrbh7VQAX3u1YmJyynvTioRdpaPAsdLhXe.s0kALMW
```

Anexo V: Ficheros para el despliegue del gestor

Los ficheros necesarios para el despliegue de Wazuh manager son los siguientes:

```
[boole@atl004s atalaia]$ tree ./config/wazuh_manager/
./config/wazuh_manager/
├── api.yaml
├── authd.pass
├── cp-filebeat-files.sh
├── filebeat.keystore
├── filebeat.yml
└── wazuh_manager.conf
```

La configuración de la API RESTful de Wazuh Manager se realiza a través del fichero api.yaml, donde destacan los siguientes ajustes [98].

- Se define la dirección IP (0.0.0.0) y el puerto de escucha (55000/tcp).
- Se ejecuta el servicio wazuh-api con el usuario wazuh del contenedor.

```
USER         PID COMMAND
wazuh        431 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-
apid.py
wazuh        432 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-
apid.py
wazuh        435 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-
apid.py
wazuh        438 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-
apid.py
```

- Se realiza la configuración HTTPS.
- Se ajustan los parámetros de acceso para evitar ataques de fuerza bruta o la sobrecarga del servicio.
- Se permite la carga de configuraciones de comandos remotos mediante las etiquetas wodle_command y localfile.

```
# https://documentation.wazuh.com/current/user-manual/api/configuration.html#api-configuration-
file

host: 0.0.0.0
port: 55000

drop_privileges: yes
experimental_features: no
max_upload_size: 10485760

intervals:
   request_timeout: 10

# Configuración HTTPS
https:
   enabled: yes
   # Certificados y claves privadas en /var/ossec/api/configuration/ssl
   key: "api.key"
   cert: "api.pem"
   use_ca: False
   ca: "root-ca.pem"
   ssl_protocol: "TLSv1.2"
   ssl_ciphers: ""

logs:
   level: "info"
```

```
   format: "plain"

cors:
   enabled: no
   source_route: "*"
   expose_headers: "*"
   allow_headers: "*"
   allow_credentials: no

cache:
   enabled: yes
   time: 0.750

access:
   max_login_attempts: 15
   block_time: 300
   max_request_per_minute: 300

upload_configuration:
   remote_commands:
      localfile:
         allow: yes
         exceptions: []
      wodle_command:
         allow: yes
         exceptions: []
   limits:
      eps:
         allow: yes
```

Fichero 34: api.yaml

El script cp-filebeat-files.sh se ejecuta al iniciar el contenedor para proporcionarle los archivos definidos en los montajes bind.

```
#!/bin/bash

################################################################################
# Copy all files from $FILEBEAT_CONFIG_MOUNT to their corresponding directory
# and respect destination files permissions.
################################################################################

FILEBEAT_CONFIG_MOUNT=/filebeat-config-mount
CERTIFICATES_INSTALL_PATH=/etc/ssl
FILEBEAT_CONFIG_INSTALL_PATH=/etc/filebeat
FILEBEAT_VAR_INSTALL_PATH=/var/lib/filebeat

if [ -e "${FILEBEAT_CONFIG_MOUNT}" ]; then
  print "Identified Filebeat files to copy..."
  cp -r ${FILEBEAT_CONFIG_MOUNT}/${CERTIFICATES_INSTALL_PATH}/* ${CERTIFICATES_INSTALL_PATH}/
  cp -r ${FILEBEAT_CONFIG_MOUNT}/${FILEBEAT_CONFIG_INSTALL_PATH}/*
${FILEBEAT_CONFIG_INSTALL_PATH}/
  cp -r ${FILEBEAT_CONFIG_MOUNT}/${FILEBEAT_VAR_INSTALL_PATH}/* ${FILEBEAT_VAR_INSTALL_PATH}/
else
  print "No Filebeat files to mount..."
fi
```

Fichero 35: cp-filebeat-files.sh

El fichero filebeat.keystore almacena de manera segura las credenciales necesarias para que el servicio Filebeat pueda enviar las alertas generadas y/o los eventos recopilados al indexador.

```
[boole@atl004s atalaia]$ docker compose exec wazuh-manager /bin/bash
root@manager:/# ls -la /var/lib/filebeat/filebeat.keystore
```

```
-rw-------. 1 root root 214 Nov  3 11:53 /var/lib/filebeat/filebeat.keystore
root@manager:/# filebeat keystore list
password
username
root@manager:/# filebeat keystore remove username
successfully removed key: username
root@manager:/# filebeat keystore remove password
successfully removed key: password
root@manager:/# filebeat keystore add indexer.username
Enter value for indexer.username:
Successfully updated the keystore
root@manager:/# filebeat keystore add indexer.password
Enter value for indexer.password:
Successfully updated the keystore
root@manager:/# filebeat keystore list
indexer.password
indexer.username
root@manager:/# exit
exit
[boole@atl004s atalaia]$ docker compose cp wazuh-
manager:/var/lib/filebeat/filebeat.keystore ./config/wazuh_manager/
[+] Copying 1/0
 ✓ manager copy manager:/var/lib/filebeat/filebeat.keystore to ./config/wazuh_manager/
Copied
```

El funcionamiento del agente Filebeat se ajusta a través del fichero filebeat.yml y destaca por tener configurado solamente el reenvío de las alertas generadas por HTTPS cara el puerto 9200/tcp de indexer.atalaia.local. Los ajustes de usuario y contraseña hacen referencia a las entradas disponible en el repositorio de claves filebeat.keystore.

```
# Wazuh - Filebeat configuration file

filebeat.modules:
  - module: wazuh
    alerts:
      enabled: true
    archives:
      enabled: false

setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.template.overwrite: true
setup.ilm.enabled: false
output.elasticsearch:
  hosts: ['https://indexer.atalaia.local:9200']
  username: ${indexer.username}
  password: ${indexer.password}
  ssl.verification_mode: 'full'
  ssl.certificate_authorities: ['/etc/ssl/root-ca.pem']
  ssl.certificate: '/etc/ssl/filebeat.pem'
  ssl.key: '/etc/ssl/filebeat.key'

logging.metrics.enabled: false

seccomp:
  default_action: allow
  syscalls:
  - action: allow
    names:
    - rseq
```

Fichero 36: filebeat.yml

Finalmente, el fichero XML wazuh_manager.conf se monta en la ruta /var/ossec/etc/ossec.conf del contenedor correspondiente y permite la definición de una amplia variedad de parámetros relacionados con el funcionamiento de Wazuh Manager [59].

```
<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
  </global>

  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

  <!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->
  <logging>
    <log_format>plain</log_format>
  </logging>

  <remote>
    <connection>secure</connection>
    <port>1514</port>
    <protocol>tcp</protocol>
    <queue_size>131072</queue_size>
  </remote>

  <!-- Policy monitoring -->
  <rootcheck>
    <disabled>no</disabled>
    <check_files>yes</check_files>
    <check_trojans>yes</check_trojans>
    <check_dev>yes</check_dev>
    <check_sys>yes</check_sys>
    <check_pids>yes</check_pids>
    <check_ports>yes</check_ports>
    <check_if>yes</check_if>

    <!-- Frequency that rootcheck is executed - every 12 hours -->
    <frequency>43200</frequency>

    <rootkit_files>etc/rootcheck/rootkit_files.txt</rootkit_files>
    <rootkit_trojans>etc/rootcheck/rootkit_trojans.txt</rootkit_trojans>

    <skip_nfs>yes</skip_nfs>
  </rootcheck>

  <wodle name="cis-cat">
    <disabled>yes</disabled>
    <timeout>1800</timeout>
    <interval>1d</interval>
    <scan-on-start>yes</scan-on-start>
```

```
  <java_path>wodles/java</java_path>
  <ciscat_path>wodles/ciscat</ciscat_path>
</wodle>

<!-- Osquery integration -->
<wodle name="osquery">
  <disabled>yes</disabled>
  <run_daemon>yes</run_daemon>
  <log_path>/var/log/osquery/osqueryd.results.log</log_path>
  <config_path>/etc/osquery/osquery.conf</config_path>
  <add_labels>yes</add_labels>
</wodle>

<!-- System inventory -->
<wodle name="syscollector">
  <disabled>no</disabled>
  <interval>1h</interval>
  <scan_on_start>yes</scan_on_start>
  <hardware>yes</hardware>
  <os>yes</os>
  <network>yes</network>
  <packages>yes</packages>
  <ports all="no">yes</ports>
  <processes>yes</processes>

  <!-- Database synchronization settings -->
  <synchronization>
    <max_eps>10</max_eps>
  </synchronization>
</wodle>

<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>

<vulnerability-detector>
  <enabled>no</enabled>
  <interval>5m</interval>
  <min_full_scan_interval>6h</min_full_scan_interval>
  <run_on_start>yes</run_on_start>

  <!-- Ubuntu OS vulnerabilities -->
  <provider name="canonical">
    <enabled>no</enabled>
    <os>trusty</os>
    <os>xenial</os>
    <os>bionic</os>
    <os>focal</os>
    <os>jammy</os>
    <update_interval>1h</update_interval>
  </provider>

  <!-- Debian OS vulnerabilities -->
  <provider name="debian">
    <enabled>no</enabled>
    <os>buster</os>
    <os>bullseye</os>
    <os>bookworm</os>
    <update_interval>1h</update_interval>
  </provider>

  <!-- RedHat OS vulnerabilities -->
```

```
    <provider name="redhat">
      <enabled>no</enabled>
      <os>5</os>
      <os>6</os>
      <os>7</os>
      <os>8</os>
      <os>9</os>
      <update_interval>1h</update_interval>
    </provider>

    <!-- Amazon Linux OS vulnerabilities -->
    <provider name="alas">
      <enabled>no</enabled>
      <os>amazon-linux</os>
      <os>amazon-linux-2</os>
      <os>amazon-linux-2023</os>
      <update_interval>1h</update_interval>
    </provider>

    <!-- SUSE Linux Enterprise OS vulnerabilities -->
    <provider name="suse">
      <enabled>no</enabled>
      <os>11-server</os>
      <os>11-desktop</os>
      <os>12-server</os>
      <os>12-desktop</os>
      <os>15-server</os>
      <os>15-desktop</os>
      <update_interval>1h</update_interval>
    </provider>

    <!-- Arch OS vulnerabilities -->
    <provider name="arch">
      <enabled>no</enabled>
      <update_interval>1h</update_interval>
    </provider>

    <!-- Alma Linux OS vulnerabilities -->
    <provider name="almalinux">
      <enabled>no</enabled>
      <os>8</os>
      <os>9</os>
      <update_interval>1h</update_interval>
    </provider>

    <!-- Windows OS vulnerabilities -->
    <provider name="msu">
      <enabled>yes</enabled>
      <update_interval>1h</update_interval>
    </provider>

    <!-- Aggregate vulnerabilities -->
    <provider name="nvd">
      <enabled>yes</enabled>
      <update_interval>1h</update_interval>
    </provider>

  </vulnerability-detector>

  <!-- File integrity monitoring -->
  <syscheck>
    <disabled>no</disabled>

    <!-- Frequency that syscheck is executed default every 12 hours -->
    <frequency>43200</frequency>
```

```
  <scan_on_start>yes</scan_on_start>

  <!-- Generate alert when new file detected -->
  <alert_new_files>yes</alert_new_files>

  <!-- Don't ignore files that change more than 'frequency' times -->
  <auto_ignore frequency="10" timeframe="3600">no</auto_ignore>

  <!-- Directories to check  (perform all possible verifications) -->
  <directories>/etc,/usr/bin,/usr/sbin</directories>
  <directories>/bin,/sbin,/boot</directories>

  <!-- Files/directories to ignore -->
  <ignore>/etc/mtab</ignore>
  <ignore>/etc/hosts.deny</ignore>
  <ignore>/etc/mail/statistics</ignore>
  <ignore>/etc/random-seed</ignore>
  <ignore>/etc/random.seed</ignore>
  <ignore>/etc/adjtime</ignore>
  <ignore>/etc/httpd/logs</ignore>
  <ignore>/etc/utmpx</ignore>
  <ignore>/etc/wtmpx</ignore>
  <ignore>/etc/cups/certs</ignore>
  <ignore>/etc/dumpdates</ignore>
  <ignore>/etc/svc/volatile</ignore>

  <!-- File types to ignore -->
  <ignore type="sregex">.log$|.swp$</ignore>

  <!-- Check the file, but never compute the diff -->
  <nodiff>/etc/ssl/private.key</nodiff>

  <skip_nfs>yes</skip_nfs>
  <skip_dev>yes</skip_dev>
  <skip_proc>yes</skip_proc>
  <skip_sys>yes</skip_sys>

  <!-- Nice value for Syscheck process -->
  <process_priority>10</process_priority>

  <!-- Maximum output throughput -->
  <max_eps>100</max_eps>

  <!-- Database synchronization settings -->
  <synchronization>
    <enabled>yes</enabled>
    <interval>5m</interval>
    <max_interval>1h</max_interval>
    <max_eps>10</max_eps>
  </synchronization>
</syscheck>

<!-- Active response -->
<global>
  <white_list>127.0.0.1</white_list>
  <white_list>^localhost.localdomain$</white_list>
</global>

<command>
  <name>disable-account</name>
  <executable>disable-account</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>
```

```
  <command>
    <name>restart-wazuh</name>
    <executable>restart-wazuh</executable>
  </command>

  <command>
    <name>firewall-drop</name>
    <executable>firewall-drop</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>

  <command>
    <name>host-deny</name>
    <executable>host-deny</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>

  <command>
    <name>route-null</name>
    <executable>route-null</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>

  <command>
    <name>win_route-null</name>
    <executable>route-null.exe</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>

  <command>
    <name>netsh</name>
    <executable>netsh.exe</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>

  <!--
  <active-response>
    active-response options here
  </active-response>
  -->

  <!-- Log analysis -->
  <localfile>
    <log_format>command</log_format>
    <command>df -P</command>
    <frequency>360</frequency>
  </localfile>

  <localfile>
    <log_format>full_command</log_format>
    <command>netstat -tulpn | sed 's/\([[:alnum:]]\+\)\ \+[[:digit:]]\+\ \+[[:digit:]]\+\
\+\(.*\):\([[:digit:]]*\)\ \+\([0-9\.\:\*]\+\).\+\ \([[:digit:]]*\/[[:alnum:]\-]*\).*/\1 \2 ==
\3 == \4 \5/' | sort -k 4 -g | sed 's/ == \(.*\) ==/:\1/' | sed 1,2d</command>
    <alias>netstat listening ports</alias>
    <frequency>360</frequency>
  </localfile>

  <localfile>
    <log_format>full_command</log_format>
    <command>last -n 20</command>
    <frequency>360</frequency>
  </localfile>

  <ruleset>
    <!-- Default ruleset -->
```

```
    <decoder_dir>ruleset/decoders</decoder_dir>
    <rule_dir>ruleset/rules</rule_dir>
    <rule_exclude>0215-policy_rules.xml</rule_exclude>
    <list>etc/lists/audit-keys</list>
    <list>etc/lists/amazon/aws-eventnames</list>
    <list>etc/lists/security-eventchannel</list>

    <!-- User-defined ruleset -->
    <decoder_dir>etc/decoders</decoder_dir>
    <rule_dir>etc/rules</rule_dir>
  </ruleset>

  <rule_test>
    <enabled>yes</enabled>
    <threads>1</threads>
    <max_sessions>64</max_sessions>
    <session_timeout>15m</session_timeout>
  </rule_test>

  <!-- Configuration for wazuh-authd -->
  <auth>
    <disabled>no</disabled>
    <port>1515</port>
    <use_source_ip>no</use_source_ip>
    <purge>yes</purge>
    <use_password>yes</use_password>
    <ciphers>HIGH:!ADH:!EXP:!MD5:!RC4:!3DES:!CAMELLIA:@STRENGTH</ciphers>
    <!-- <ssl_agent_ca></ssl_agent_ca> -->
    <ssl_agent_ca>/etc/ssl/root-ca.pem</ssl_agent_ca>
    <ssl_verify_host>yes</ssl_verify_host>
    <ssl_manager_cert>/etc/ssl/filebeat.pem</ssl_manager_cert>
    <ssl_manager_key>/etc/ssl/filebeat.key</ssl_manager_key>
    <ssl_auto_negotiate>no</ssl_auto_negotiate>
  </auth>

  <cluster>
    <name>wazuh</name>
    <node_name>manager</node_name>
    <node_type>master</node_type>
    <key>aa093264ef885029653eea20dfcf51ae</key>
    <port>1516</port>
    <bind_addr>0.0.0.0</bind_addr>
    <nodes>
        <node>wazuh.manager</node>
    </nodes>
    <hidden>no</hidden>
    <disabled>yes</disabled>
  </cluster>

</ossec_config>

<ossec_config>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/ossec/logs/active-responses.log</location>
  </localfile>

</ossec_config>
```

Fichero 37: wazuh_manager.conf

Anexo VI: Ficheros para el despliegue del panel de control

Los ficheros necesarios para el despliegue del panel de control son los que se muestran a continuación.

```
[boole@atl004s atalaia]$ tree ./config/wazuh_dashboard/
./config/wazuh_dashboard/
├── opensearch_dashboards.keystore
├── opensearch_dashboards.yml
└── wazuh.yml
```

La configuración del panel de control o dashboard se realiza a través del fichero opensearch_dashboards.yml [99–101].

```
server.host: 0.0.0.0
# server.port: 5601
server.port: 443
opensearch.hosts: https://indexer.atalaia.local:9200
opensearch.ssl.verificationMode: full
opensearch.requestHeadersWhitelist: ["securitytenant","Authorization"]
opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ["kibana_read_only"]
server.ssl.enabled: true
server.ssl.key: "/usr/share/wazuh-dashboard/certs/wazuh-dashboard-key.pem"
server.ssl.certificate: "/usr/share/wazuh-dashboard/certs/wazuh-dashboard.pem"
opensearch.ssl.certificateAuthorities: ["/usr/share/wazuh-dashboard/certs/root-ca.pem"]
uiSettings.overrides.defaultRoute: /app/wazuh
```

Fichero 38: opensearch_dashboards.yml

- La dirección IP y el puerto a través del cual se accede al servicio vienen definidos por los ajustes en server.host y server.port.
- Se define la URL de acceso al indexador a través de la propiedad opensearch.hosts.
- El tráfico HTTPS por TLS está habilitado con la propiedad server.ssl.enabled y los certificados necesarios se definen en los ajustes:
    - server.ssl.certificate: Certificado del servicio.
    - server.ssl.key: Clave privada asociada al certificado.
    - opensearch.ssl.certificateAuthorities: Certificado de la Autoridad de Certificación que emitió el certificado del servicio.
- El ajuste opensearch.ssl.verificationMode establece una verificación completa, nombre de host y del certificado, para la conexión TLS desde el panel de control al indexador.
- Las propiedades siguientes están pensadas para entornos multiarrendatario y no aplican para la implementación propuesta:
    - opensearch.requestHeadersWhitelist: Lista blanca de encabezados necesarios para un entorno multitenancy.
    - opensearch_security.multitenancy.enabled: Deshabilitado al no ser necesario.
- Se establece el rol de sólo lectura kibana_read_only a través de la propiedad opensearch_security.readonly_mode.roles.

- Se utiliza el ajuste uiSettings.overrides.defaultRoute para definir la ruta por defecto del panel de control.

Es posible realizar ajustes relacionados con varios aspectos de la configuración del dashboard a través del fichero wazuh.yml [102] o mediante el apartado de ajustes (Settings) de su interfaz gráfica y, en este caso, contiene la configuración necesaria para conectarse a la API RESTful del gestor de Wazuh, donde run_as deshabilitado indica que la autorización para acceder a los recursos se basará exclusivamente en los roles asignados.

```
hosts:
  - 1513629884013:
      url: https://manager.atalaia.local
      port: 55000
      username: wazuh-wui
      password: risc-23*C1sC
      run_as: false
```

Fichero 39: wazuh.yml

El fichero opensearch_dashboards.keystore contiene las credenciales necesarias para acceder al indexador.

```
[boole@atl004s atalaia]$ docker compose exec wazuh-dashboard /bin/bash
wazuh-dashboard@dashboard:~$ /usr/share/wazuh-dashboard/bin/opensearch-dashboards-keystore
list
v16.20.0
opensearch.username
opensearch.password

wazuh-dashboard@dashboard:~$ /usr/share/wazuh-dashboard/bin/opensearch-dashboards-keystore
add -f opensearch.username
v16.20.0
Enter value for opensearch.username: kibanaserver

wazuh-dashboard@dashboard:~$ /usr/share/wazuh-dashboard/bin/opensearch-dashboards-keystore
add -f opensearch.password
v16.20.0
Enter value for opensearch.password: ********
```

Anexo VII: Verificaciones de los componentes centrales de la plataforma Wazuh

En este anexo se enumeran verificaciones que permiten comprobar el correcto funcionamiento de los componentes centrales de la plataforma de seguridad Wazuh:

- Se comprueba que se han abierto los puertos necesarios en la pila de red del host anfitrión Docker.

```
[boole@atl004s atalaia]$ ss -patnl |grep -i "State\|443\|1514\|1515\|9200\|9300\|55000"
State  Recv-Q Send-Q       Local Address:Port  Process
LISTEN 0      128               0.0.0.0:1515
LISTEN 0      128               0.0.0.0:1514
LISTEN 0      128               0.0.0.0:55000
LISTEN 0      511               0.0.0.0:443   users:(("node",pid=321244,fd=18))
LISTEN 0      4096   [::ffff:10.10.10.4]:9300  users:(("java",pid=319201,fd=555))
LISTEN 0      4096   [::ffff:10.10.10.4]:9200  users:(("java",pid=319201,fd=557))

Aviso: Se ha eliminado el contenido de la columna “Peer Address:Port”
```

- Se verifican los certificados de cada uno de los servicios

```
[boole@atl004s atalaia]$ openssl s_client -CAfile ./config/certs/atalaia-root-ca.crt -
connect indexer.atalaia.local:9200 | openssl x509 -noout -subject -ext subjectAltName -
nameopt oneline,-esc_msb
depth=1 C = ES, ST = A Coru\C3\B1a, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN
= atalaia-root-ca.atalaia.local
verify return:1
depth=0 C = ES, ST = A Coru\C3\B1a, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN
= indexer.atalaia.local
verify return:1
subject=C = ES, ST = A Coruña, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN =
indexer.atalaia.local
X509v3 Subject Alternative Name:
    DNS:indexer.atalaia.local, IP Address:10.10.10.4

[boole@atl004s atalaia]$ openssl s_client -CAfile ./config/certs/atalaia-root-ca.crt -
connect manager.atalaia.local:55000 | openssl x509 -noout -subject -ext subjectAltName -
nameopt oneline,-esc_msb
depth=1 C = ES, ST = A Coru\C3\B1a, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN
= atalaia-root-ca.atalaia.local
verify return:1
depth=0 C = ES, ST = A Coru\C3\B1a, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN
= manager.atalaia.local
verify return:1
subject=C = ES, ST = A Coruña, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN =
manager.atalaia.local
X509v3 Subject Alternative Name:
    DNS:manager.atalaia.local, IP Address:10.10.10.4

[boole@atl004s atalaia]$ openssl s_client -CAfile ./config/certs/atalaia-root-ca.crt -
connect dashboard.atalaia.local:443 | openssl x509 -noout -subject -ext subjectAltName -
nameopt oneline,-esc_msb
depth=1 C = ES, ST = A Coru\C3\B1a, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN
= atalaia-root-ca.atalaia.local
verify return:1
depth=0 C = ES, ST = A Coru\C3\B1a, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN
= dashboard.atalaia.local
verify return:1
subject=C = ES, ST = A Coruña, L = Santiago de Compostela, O = Atalaia, OU = ATIC, CN =
dashboard.atalaia.local
X509v3 Subject Alternative Name:
    DNS:dashboard.atalaia.local, IP Address:10.10.10.4
```

- Se verifica que se puede acceder correctamente al servicio del indexador de Wazuh, el cual consiste en el motor de búsqueda y análisis distribuido OpenSearch.

```
[boole@atl004s atalaia]$ curl --cacert ./config/certs/atalaia-root-ca.crt
https://indexer.atalaia.local:9200/ -u admin -s |jq
Enter host password for user 'admin':
{
  "name": "indexer",
  "cluster_name": "atalaia",
  "cluster_uuid": "f9AKtSlbToeoyLjFN17nkg",
  "version": {
    "number": "7.10.2",
    "build_type": "rpm",
    "build_hash": "db90a415ff2fd428b4f7b3f800a51dc229287cb4",
    "build_date": "2023-06-03T06:24:25.112415503Z",
    "build_snapshot": false,
    "lucene_version": "9.6.0",
    "minimum_wire_compatibility_version": "7.10.0",
    "minimum_index_compatibility_version": "7.0.0"
  },
  "tagline": "The OpenSearch Project: https://opensearch.org/"
}
```

- Se realiza una prueba de conectividad HTTPS desde el componente Filebeat del gestor Wazuh.

```
[boole@atl004s atalaia]$ docker compose exec wazuh-manager filebeat test output
elasticsearch: https://indexer.atalaia.local:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 10.10.10.4
    dial up... OK
  TLS...
    security: server's certificate chain verification is enabled
    handshake... OK
    TLS version: TLSv1.3
    dial up... OK
  talk to server... OK
  version: 7.10.2
```

- La API RESTFul disponible en el puerto 55000/tcp del gestor Wazuh se prueba solicitando un token de autorización para utilizar en peticiones API.

```
[boole@atl004s atalaia]$ TOKEN=$(curl -s --cacert ./config/certs/atalaia-root-ca.crt -u
wazuh-wui -X POST
"https://manager.atalaia.local:55000/security/user/authenticate?raw=true")
Enter host password for user 'wazuh-wui':

[boole@atl004s atalaia]$ curl -s --cacert ./config/certs/atalaia-root-ca.crt -X GET
"https://manager.atalaia.local:55000/" -H "Authorization: Bearer $TOKEN" |jq
{
  "data": {
    "title": "Wazuh API REST",
    "api_version": "4.7.0",
    "revision": 40704,
    "license_name": "GPL 2.0",
    "license_url": "https://github.com/wazuh/wazuh/blob/v4.7.0/LICENSE",
    "hostname": "manager",
    "timestamp": "2023-12-06T19:27:34Z"
  },
  "error": 0
}
```

- Se verifica el acceso web al panel de control de Wazuh y que está operativa la conexión con la API RESTful de Wazuh Manager.

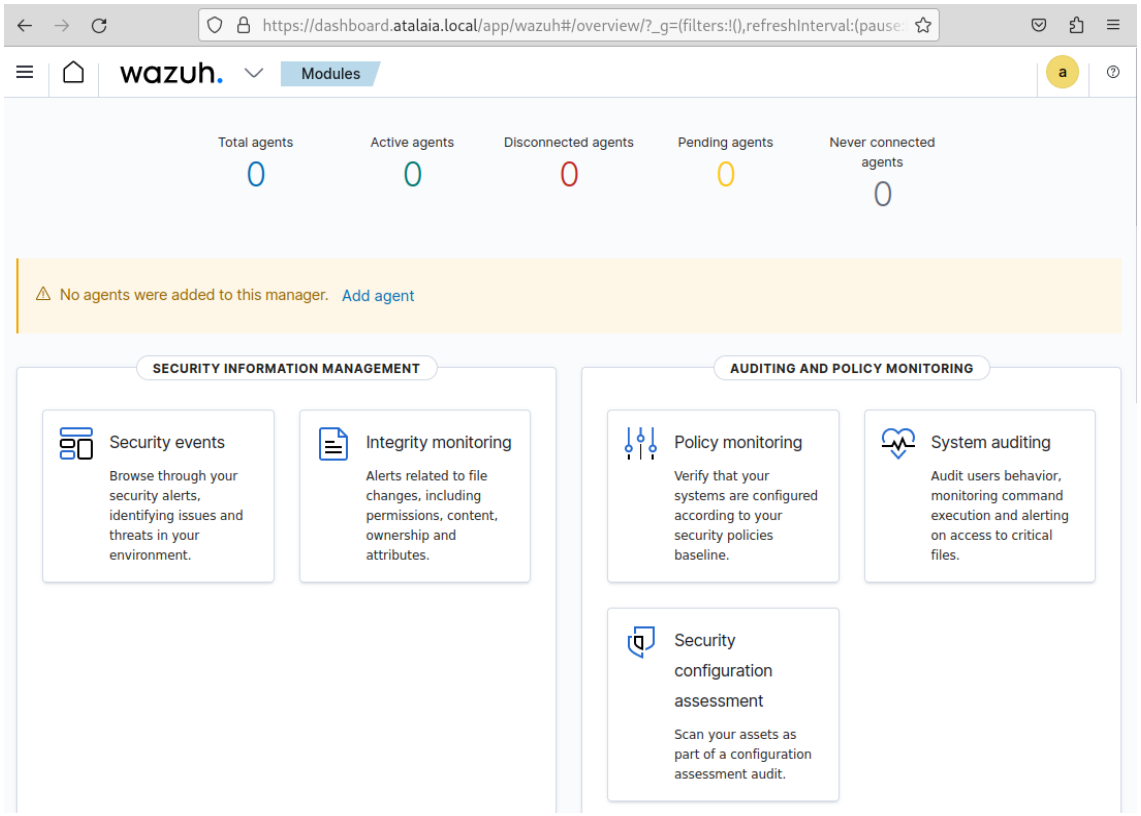


Figura 42: Interfaz web del panel de control de Wazuh

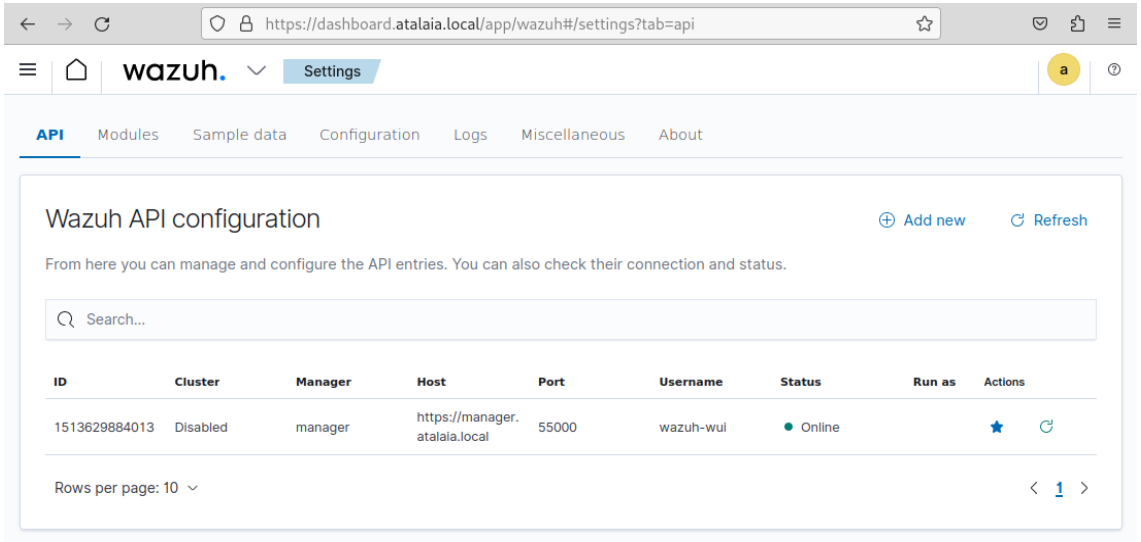


Figura 43: Conexión con la API RESTful del gestor Wazuh