

# ***Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat***

---

LAVEREDA

ESO  
BATXILLERAT  
CICLES

**Héctor Rafael Ruiz Freitas - 2ºASIR - 2025**  
**Administración de Sistemas Informáticos en**  
**Red**  
**IES La Vereda - Víctor Castro Sancho**

Proyecto de creación de una infraestructura de monitorización y detección de comportamientos sospechosos mediante la pila ELK y Filebeat. El objetivo es recolectar, centralizar y analizar los logs de múltiples equipos dentro de una red utilizando inteligencia artificial para identificar accesos indebidos, cambios no autorizados y actividades anómalas. El entorno está compuesto por un servidor Ubuntu 22.04 con ELK Stack instalado y configurado para recibir logs desde clientes con Windows 10 mediante Filebeat. Los clientes tienen instalado Filebeat versión 8.17.1 que está enviando logs correctamente al servidor. La red incluye también un firewall OpnSense que proporciona conectividad y separación de segmentos. Kibana está disponible para visualizar los logs y crear dashboards personalizados. A futuro se planea habilitar módulos adicionales en Filebeat como system o windows y migrar del input log a filestream para mayor compatibilidad. Se contempla además añadir capacidades de análisis mediante inteligencia artificial para potenciar la detección de amenazas y elevar el nivel de seguridad de la red

# ***Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat***

***Héctor Rafael Ruiz Freitas***

**Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades  
Sospechosas con ELK + Filebeat**

**Índice:**

<b>Resumen:</b>	<b>6</b>
Objetivo general:	6
<b>Entorno:</b>	<b>7</b>
Entorno de Virtualización:	7
Sistema Operativo en Cada Máquina Virtual:	7
Software Utilizado:	7
<b>Infraestructura de Red:</b>	<b>8</b>
Topología de Red:	8
Funciones de Cada Nodo:	8
Direcciones IP de cada máquina:	9
<b>Instalacion y Configuración de Maquinas Virtuales:</b>	<b>9</b>
<b>Servidor de Monitorización (Wazuh):</b>	<b>11</b>
<b>Instalación de Clientes:</b>	<b>12</b>
Descripción de las máquinas cliente:	12
Instalación y configuración del agente Wazuh:	13
<b>Paneles de Control (Dashboards):</b>	<b>13</b>
Estado de los agentes:	14
Alertas por severidad:	14
Escaladas de privilegios:	14
Cambios en archivos sensibles:	14
Conexiones sospechosas:	15
Accesos fallidos o repetidos:	15
Nuevos procesos ejecutados:	15
<b>Análisis de Logs y Eventos:</b>	<b>15</b>
Tipos de logs recogidos:	16

**Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades  
Sospechosas con ELK + Filebeat**

Ejemplos de alertas importantes.....	16
Reacción ante incidentes simulados.....	17
<b>Seguridad.....</b>	<b>17</b>
Prácticas de endurecimiento aplicadas (Hardening).....	18
Seguridad de la red virtual.....	18
Gestión de usuarios y privilegios.....	19
<b>Pruebas Realizadas.....</b>	<b>20</b>
Simulación de un ataque.....	20
Detección desde Wazuh.....	20
Respuesta en el firewall.....	21
<b>Resultados Obtenidos.....</b>	<b>22</b>
Valoración de la efectividad.....	22
Métricas y gráficas destacadas.....	22
Limitaciones detectadas.....	23
<b>Conclusiones y Trabajo Futuro.....</b>	<b>24</b>
Lecciones aprendidas.....	24
Posibles mejoras.....	24
Aplicabilidad en entornos reales e integración de Inteligencia Artificial.....	25
<b>Implementación del Sistema de Mensajería: Telegram y Gmail.....</b>	<b>26</b>
Telegram como canal de alerta inmediata.....	26
Gmail como sistema complementario de notificación.....	26

**Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades  
Sospechosas con ELK + Filebeat**

**Resumen:**

**Objetivo general:**

El principal objetivo de este proyecto es crear una infraestructura virtualizada que integre las herramientas necesarias para monitorizar, detectar y responder a posibles incidentes de seguridad en una red de ordenadores. Para lograrlo, se utilizarán tecnologías como OPNsense, para el control del tráfico y la gestión de la red, y Wazuh, como plataforma de monitorización, correlación de eventos y detección de intrusiones.

Entre los objetivos específicos destacan:

- **Desplegar una infraestructura de red virtual segura y funcional**, utilizando máquinas virtuales configuradas para simular distintos equipos y roles de una red empresarial.
- **Implementar un sistema de monitorización y detección de intrusos con Wazuh**, capaz de registrar eventos relevantes, generar alertas y detectar comportamientos anómalos.
- **Incorporar sistemas de notificación automática mediante servicios de mensajería como Telegram y Gmail**, de modo que los administradores reciban alertas críticas en tiempo real.
- **Crear dashboards personalizados en Kibana** que permitan visualizar de forma clara y centralizada el estado de los sistemas, los agentes conectados y los eventos críticos.

## **Entorno:**

### **Entorno de Virtualización:**

Para el desarrollo de este proyecto se ha utilizado **VirtualBox**, una herramienta de virtualización de código abierto desarrollada por Oracle. Esta elección se debe a su facilidad de uso, compatibilidad multiplataforma y la posibilidad de crear entornos virtuales de red complejos, sin necesidad de depender de un hipervisor de pago. VirtualBox permite una administración eficiente de recursos, la creación de snapshots y una configuración detallada de las redes virtuales, lo que ha sido fundamental para simular un entorno empresarial seguro y controlado.

Dentro de VirtualBox se han desplegado varias máquinas virtuales interconectadas en una red interna, simulando un entorno corporativo que incluye estaciones de trabajo, un servidor de monitorización y una puerta de enlace de red. Esta arquitectura ha permitido realizar pruebas controladas sobre la detección de amenazas, monitorización de tráfico y respuesta ante incidentes de seguridad.

### **Sistema Operativo en Cada Máquina Virtual**

Se han utilizado distintos sistemas operativos según el rol que desempeña cada máquina:

- **Servidor de Monitorización (Wazuh):** Ubuntu 22.04.5 LTS. Este sistema se ha seleccionado por su estabilidad, compatibilidad con herramientas de seguridad y facilidad de automatización mediante scripts.
- **Clientes de Red:** Windows 10 Pro. Estas máquinas simulan los equipos de los usuarios dentro de una empresa y generan eventos de seguridad como inicios de sesión, ejecuciones de procesos o cambios en archivos.
- **Firewall/Gateway:** OPNsense. Este sistema basado en FreeBSD actúa como cortafuegos y router entre la red interna y el exterior. Se ha configurado para ofrecer servicios de NAT, DHCP y DNS, así como reglas de cortafuegos para simular escenarios de seguridad reales.

### **Software Utilizado**

El ecosistema de seguridad y monitorización del proyecto se ha construido mediante el uso de las siguientes herramientas:

- **OPNsense:** Distribución open-source basada en FreeBSD, especializada en cortafuegos y ruteo. Su panel web intuitivo y sus capacidades avanzadas han permitido implementar políticas de seguridad y monitorizar el tráfico de red.

## **Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat**

- **Wazuh:** Plataforma de seguridad y análisis que permite la recolección, procesamiento y visualización de eventos de seguridad desde múltiples puntos de la red. Incluye funcionalidades como detección de intrusiones (IDS), integridad de archivos, control de acceso, y correlación de logs.
- **Elasticsearch:** Motor de búsqueda y análisis que forma parte del stack ELK. Su papel en el proyecto es almacenar grandes volúmenes de eventos estructurados para permitir búsquedas rápidas y análisis de datos históricos.
- **Kibana:** Herramienta de visualización que se conecta a Elasticsearch. En este proyecto se ha utilizado para construir dashboards personalizados que muestran el estado de los agentes, alertas por severidad, eventos sospechosos y mucho más.
- **Filebeat:** Agente ligero que se instala en los sistemas cliente para recolectar logs y enviarlos a Wazuh/Elasticsearch. Filebeat permite una recolección en tiempo real, facilitando la detección temprana de incidentes.

### **Infraestructura de Red:**

#### **Topología de Red:**

La topología diseñada para este proyecto simula una pequeña red corporativa en un entorno virtual controlado. Se ha utilizado una **topología en estrella**, donde todos los dispositivos están conectados a través de un firewall centralizado (OPNsense), que actúa como puerta de enlace y controlador de acceso a Internet. Este diseño permite una gestión centralizada del tráfico y un control granular de las reglas de seguridad.

Todos los dispositivos están conectados a una red interna gestionada por VirtualBox. La interfaz LAN de OPNsense enruta el tráfico entre los clientes y el servidor de monitorización, y la interfaz WAN conecta con Internet (si es necesario para actualizaciones o mensajería externa).

#### **Funciones de Cada Nodo**

- **OPNsense (Firewall/Gateway):**
  - Asigna direcciones IP mediante DHCP o estáticamente.
  - Filtra el tráfico de red mediante reglas de firewall.
  - Gestiona el NAT para acceso a Internet.
  - Controla el flujo de datos entre la red interna y externa.
  - Actúa como punto central de la topología de red.



## Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat

- **Ubuntu 22.04.5 (Servidor Wazuh):**
  - Instancia principal de Wazuh.
  - Recibe y analiza logs de seguridad desde los clientes.
  - Aloja los componentes de Wazuh: API, Manager, Dashboard, Elasticsearch, Filebeat y Kibana.
  - Provee dashboards visuales y alertas personalizadas.
- **Windows 10 1, 2 y 3 (Clientes):**
  - Generan eventos de sistema reales como inicios de sesión, ejecución de procesos, acceso a archivos.
  - Tienen instalado el **agente Wazuh** para reportar eventos al servidor.
  - Simulan usuarios de red con actividad diaria.

### Direcciones IP de cada máquina:

Dispositivo	Rol	Direccion Ip
Opsense	firewall LAN	192.168.10.1(24)
Ubuntu	servidor de monitoreo	192.168.10.20(24)
Windows 10	cliente 1	192.168.10.21(24)
Windows 10	cliente 2	192.168.10.22(24)
Windows 10	cliente 3	192.168.10.23(24)
Kali		

### Adaptador de red Firewall/Gateway(OpenSense):

```
*** OPNsense.localdomain: OPNsense 25.1 (amd64) ***  
  
LAN (em1)      -> v4: 192.168.10.10/24  
WAN (em0)      -> v4/DHCP4: 10.0.2.15/24  
                v6: fd17:625c:f037:2:a00:27ff:fe29:d078/64
```

Red Opsense

Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades  
Sospechosas con ELK + Filebeat

**Adaptador de red Servidor (Ubuntu):**

IPv4 Method

☐ Automatic (DHCP) ☐ Link-Local Only

☒ Manual ☐ Disable

☐ Shared to other computers

Addresses

Address	Netmask	Gateway	
192.168.10.20	255.255.255.0	192.168.10.10	

DNS

Automatic ☒

8.8.8.8

Separate IP addresses with commas

Red Servidor

**Adaptador de red Cliente 1 (Windows 10):**

☒ Usar la siguiente dirección IP:

Dirección IP: 192 . 168 . 10 . 21

Máscara de subred: 255 . 255 . 255 . 0

Puerta de enlace predeterminada: 192 . 168 . 10 . 10

☐ Obtener la dirección del servidor DNS automáticamente

☒ Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido: 8 . 8 . 8 . 8

Servidor DNS alternativo: 8 . 8 . 4 . 4

☐ Validar configuración al salir

Opciones avanzadas...

Aceptar Cancelar

Red cliente 1€

**Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades  
Sospechosas con ELK + Filebeat**

**Adaptador de red Cliente 2 (Windows 10):**

☒ Usar la siguiente dirección IP:

Dirección IP: 192 . 168 . 10 . 22

Máscara de subred: 255 . 255 . 255 . 0

Puerta de enlace predeterminada: 192 . 168 . 10 . 10

☐ Obtener la dirección del servidor DNS automáticamente

☒ Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido: 8 . 8 . 8 . 8

Servidor DNS alternativo: 8 . 8 . 4 . 4

☐ Validar configuración al salir

Opciones avanzadas...

Aceptar Cancelar

Red cliente 2

**Adaptador de red Cliente 3 (Windows 10):**

☒ Usar la siguiente dirección IP:

Dirección IP: 192 . 168 . 10 . 23

Máscara de subred: 255 . 255 . 255 . 0

Puerta de enlace predeterminada: 192 . 168 . 10 . 10

☐ Obtener la dirección del servidor DNS automáticamente

☒ Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido: 8 . 8 . 8 . 8

Servidor DNS alternativo: 8 . 8 . 4 . 4

☐ Validar configuración al salir

Opciones avanzadas...

Aceptar Cancelar

Red cliente 3

## **Instalacion y Configuracion de Maquinas Virtuales:**

OPNsense se desplegó en una máquina virtual dentro del entorno VirtualBox, configurada con dos interfaces de red: una conectada al exterior mediante NAT para garantizar acceso a Internet, y otra que simula la red interna donde residen tanto el servidor de seguridad como los clientes monitorizados. Esta configuración permite que OPNsense actúe como punto de entrada y salida de todo el tráfico que circula entre las distintas máquinas virtuales, permitiendo una supervisión y control exhaustivo.

Una vez operativo, OPNsense fue configurado para asumir el rol de firewall, permitiendo definir políticas de filtrado de tráfico precisas basadas en interfaces, protocolos, puertos y direcciones IP. A través de su cortafuegos de estado, se establecieron reglas que permiten el tráfico saliente desde la red interna hacia Internet, mientras se bloquea por defecto cualquier intento de acceso externo no autorizado. Esta estrategia de seguridad perimetral asegura que solo el tráfico legítimo tenga permitido el paso, reduciendo así el riesgo de intrusiones o accesos indebidos.

Paralelamente, OPNsense fue configurado como **servidor DHCP**, encargándose de asignar dinámicamente direcciones IP a los dispositivos conectados a la red interna. Este servicio simplifica la gestión de red, eliminando la necesidad de configurar manualmente cada dispositivo. El rango de direcciones IP definido asegura una adecuada distribución sin conflictos, mientras que las reservas por dirección MAC, en algunos casos, garantizan la asignación fija para equipos críticos como el servidor Wazuh.

En lo relativo a la resolución de nombres, se hizo uso del DNS Resolver integrado en OPNsense. Este componente actúa como intermediario entre los clientes internos y los servidores DNS públicos (como los de Google), permitiendo la resolución eficiente y segura de nombres de dominio. Al mantener el control de DNS dentro del entorno, también se pueden bloquear o redirigir dominios específicos si se detecta comportamiento sospechoso o malicioso en los clientes.

El componente de NAT de OPNsense es esencial en esta arquitectura. Gracias a él, los dispositivos de la red interna pueden acceder a recursos externos utilizando una única IP pública, lo que no solo ahorra direcciones IP, sino que también oculta la estructura interna de la red ante el exterior, añadiendo una capa adicional de seguridad.

Toda esta configuración dota a la infraestructura de una base sólida para el monitoreo y la seguridad. Al centralizar las funciones de red en OPNsense, se facilita la inspección del tráfico, el control de acceso y la detección de anomalías. Su papel como intermediario permite, además, registrar intentos de conexión, movimientos entre segmentos de red y cualquier patrón que pueda sugerir una actividad maliciosa.

Esta instalación y configuración no solo proporciona conectividad entre los distintos componentes del sistema, sino que también establece el marco de seguridad fundamental sobre el que se apoya la recolección y análisis de eventos mediante herramientas como Wazuh y ELK. OPNsense no es solo un firewall, sino el guardián que protege el perímetro

## **Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat**

de una red virtual diseñada para entrenar y demostrar capacidades avanzadas de ciberdefensa.

### **Servidor de Monitorización (Wazuh)**

Dentro del esquema de monitorización y seguridad de esta infraestructura, el componente central es el servidor Wazuh, cuya función principal es recolectar, analizar y correlacionar datos provenientes de los diferentes equipos de la red. Esta solución, de código abierto, ha sido seleccionada por su capacidad para ofrecer una visión detallada del estado de seguridad de los sistemas monitorizados, su escalabilidad y su integración nativa con la pila ELK (Elasticsearch, Logstash y Kibana).

El servidor Wazuh fue desplegado en una máquina virtual basada en Ubuntu 22.04 LTS, elegida por su estabilidad, compatibilidad y bajo consumo de recursos. La instalación se realizó empleando el script oficial del proyecto, el cual automatiza la implementación de los tres componentes fundamentales: Wazuh Manager, Wazuh Indexer y Wazuh Dashboard.

El Wazuh Manager es el cerebro del sistema. Es el encargado de recibir y analizar los datos enviados por los agentes desplegados en los distintos equipos cliente. Estos datos abarcan desde logs del sistema operativo hasta eventos relacionados con integridad de archivos, ejecución de comandos, intentos de acceso, cambios en privilegios, y mucho más. A través de su motor de correlación, el manager es capaz de detectar patrones sospechosos y generar alertas en tiempo real.

Para el almacenamiento y consulta eficiente de los eventos, se utilizó Wazuh Indexer, una variante personalizada de Elasticsearch optimizada para la integración con Wazuh. Este componente se encarga de indexar todos los logs y eventos recibidos del manager, permitiendo una consulta rápida y estructurada mediante búsquedas personalizadas y dashboards. Su rol es esencial para poder navegar grandes volúmenes de datos de manera fluida y efectiva.

El tercer componente instalado fue el Wazuh Dashboard, que ofrece una interfaz web amigable para consultar los datos almacenados. Basado en Kibana, pero personalizado por el equipo de Wazuh, este panel permite visualizar gráficas, alertas, mapas de amenazas y múltiples módulos especializados, como la vigilancia de integridad de archivos, análisis de vulnerabilidades, y actividad de los usuarios. Su diseño modular facilita la creación de dashboards específicos para distintos contextos, lo que fue especialmente útil en este proyecto para mostrar información clave como agentes conectados, alertas críticas, y comportamiento anómalo en la red.

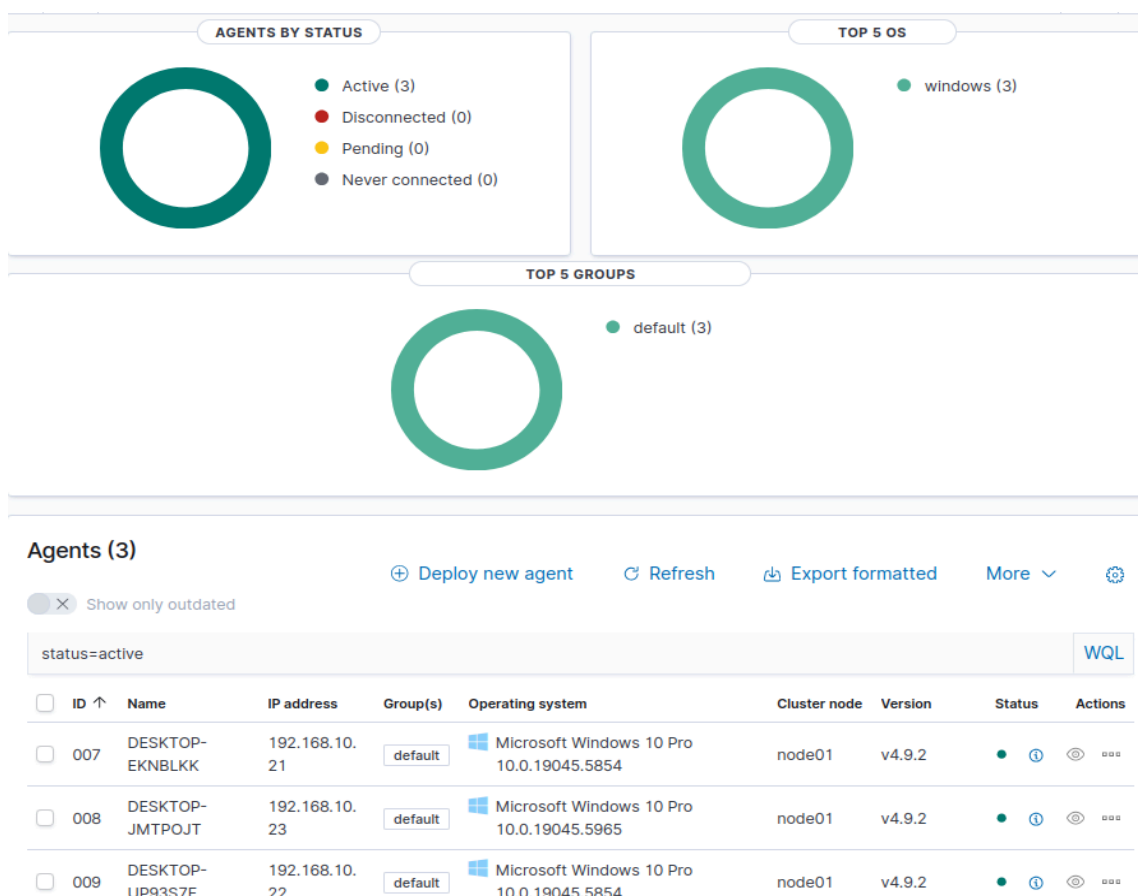
La instalación y configuración de los agentes de Wazuh en las máquinas cliente fue igualmente una parte fundamental del proceso. Cada equipo Windows fue configurado con su correspondiente agente, el cual establece comunicación segura con el Wazuh Manager a través de certificados. Estos agentes recolectan eventos del sistema, como inicios de sesión, ejecución de procesos, cambios en archivos sensibles o intentos fallidos de

## Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat

autenticación. A través de una configuración personalizada, se habilitaron módulos de auditoría avanzados y reglas específicas para capturar comportamientos potencialmente maliciosos o sospechosos.

Finalmente, se integró el entorno con Kibana mediante el Wazuh Dashboard, aprovechando la potencia de visualización que ofrece esta herramienta. Aunque Kibana forma parte del stack ELK, la versión modificada por Wazuh proporciona plantillas prediseñadas para los distintos tipos de alertas, lo que facilitó en gran medida la interpretación de la información recolectada. Además, se configuraron dashboards personalizados para mostrar datos alineados con los objetivos del proyecto, tales como conexiones de red inusuales, accesos no autorizados, escaladas de privilegios y cambios en archivos críticos.

Gracias a esta arquitectura de monitorización, se logró construir un sistema de vigilancia centralizado, capaz de detectar eventos clave de seguridad en tiempo real, visualizar patrones de riesgo y mantener un control detallado sobre la actividad de los sistemas. La instalación de Wazuh no solo aportó visibilidad, sino también capacidades de alerta proactiva, fundamentales para prevenir incidentes o actuar rápidamente ante cualquier anomalía.



El Dashboard de Wazuh con los agentes y su información.

## **Instalación de Clientes**

En el entorno diseñado para este proyecto, se incluyeron varias máquinas cliente con el objetivo de simular un entorno corporativo realista en el que existiera una variedad de dispositivos finales que generarían eventos susceptibles de ser monitorizados. Estas máquinas cliente fueron desplegadas dentro de la plataforma de virtualización **VirtualBox**, seleccionada por su compatibilidad, facilidad de uso y recursos disponibles. Las máquinas se integraron en la misma red virtual administrada por **OPNsense**, el cual actúa como firewall y gateway.

### **Descripción de las máquinas cliente**

Se utilizaron dos tipos principales de sistemas operativos en las máquinas cliente:

- **Windows 10 Pro:** Se implementaron tres máquinas con este sistema operativo. Fueron configuradas con roles típicos de un entorno corporativo, como estaciones de trabajo de usuario final. Se habilitaron servicios como registro de eventos de seguridad y autenticación para permitir la recolección de información relevante.

Todas las máquinas cliente se unieron a la red interna virtual, con direcciones IP estáticas dentro del rango definido por el firewall, y con conectividad directa al servidor de monitorización Wazuh.

### **Instalación y configuración del agente Wazuh**

Para permitir la monitorización desde el servidor Wazuh, fue necesario instalar el **agente Wazuh** en cada una de las máquinas cliente. Este agente se encarga de recolectar información local del sistema (logs, alertas, actividad de procesos, cambios en archivos, etc.) y enviarla al servidor Wazuh para su análisis y correlación.

En los sistemas **Windows**, la instalación del agente Wazuh se realizó utilizando el instalador oficial proporcionado por Wazuh. Una vez instalado, se modificó el archivo de configuración `ossec.conf` para establecer la dirección IP del servidor Wazuh (192.168.10.20 en este caso), garantizando así la correcta comunicación. Se aseguraron también los servicios necesarios para la recolección de logs de seguridad de Windows Event Channel, lo que permite detectar eventos como accesos fallidos, inicios de sesión, elevaciones de privilegios y cambios en políticas de seguridad.

En el caso de Windows, la configuración fue realizada desde el asistente gráfico, estableciendo igualmente la dirección IP del servidor y probando la conectividad antes de registrar el agente.

## Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat

Además, se añadieron reglas personalizadas para ampliar la detección en ciertas áreas como intentos de fuerza bruta o cambios en archivos sensibles. También se probó la recolección de logs específicos (como syslog o eventos de procesos) para validar que la instalación estaba completamente funcional.

### **Paneles de Control (Dashboards)**

Una de las funcionalidades más destacadas del sistema de monitorización Wazuh, en conjunto con **Kibana**, es la posibilidad de construir paneles de control visuales que permiten representar de manera gráfica, clara y dinámica toda la información recogida por los agentes desplegados. Estos **dashboards personalizados** han sido fundamentales en este proyecto para presentar los principales indicadores de seguridad y facilitar el análisis de eventos relevantes.


Los paneles fueron contruidos utilizando el módulo de visualización de Kibana, que permite añadir distintas visualizaciones (gráficas, tablas, indicadores, etc.) a un tablero común. Cada visualización fue diseñada para representar un aspecto específico de la actividad de la red y de los clientes monitorizados.

### **Estado de los agentes**

Este panel proporciona una visión general de la conectividad y el estado de todos los agentes Wazuh desplegados en la red. Se muestran los agentes clasificados por estado: **activos**, **desconectados** y **pendientes**. Esta información permite verificar en tiempo real si hay agentes que han perdido la comunicación con el servidor, lo cual podría indicar un problema de red o un fallo en el sistema monitorizado.

La visualización fue construida a partir del campo data.status (o agent.status en versiones más recientes), y filtrada por valores "active", "disconnected" y "pending".

#### **Agentes Activos**



status: Descending	Count
disconnected	107
active	11
pending	2

Visualización de agentes activos.




## Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat

### Alertas por severidad

Este panel agrupa las alertas generadas por el sistema según su nivel de severidad: **Critical**, **High**, **Medium**, y **Low**. Gracias a esta clasificación, el usuario puede enfocar su atención en los eventos más críticos que requieran una respuesta inmediata. Esta visualización también permite observar si se está produciendo un aumento inusual en ciertas categorías de riesgo.

El panel se basó en el campo rule.level, aplicando rangos de valores para agrupar los eventos por severidad. Se utilizaron filtros y buckets personalizados para cada intervalo.

#### Alertas de Nivel de seguridad



filters	Count
Bajo	0
Medio	0
Alto	0
Critico	0


Visualización de alertas de seguridad.

### Escaladas de privilegios

El objetivo de este panel es detectar intentos de escalada de privilegios, es decir, situaciones en las que un usuario normal intenta obtener permisos elevados en el sistema. Estos intentos pueden indicar acciones maliciosas o malas prácticas de administración.

La visualización se creó filtrando por rule.groups: "privilege\_escalation" o por descripciones específicas como "Windows Logon Success" en contextos anómalos. El resultado se representa en formato tabla con el nombre del agente, la IP y la descripción del evento.

#### Deteccion de escalas de privilegios



filters	Count
rule.groups: "privilege_escalation" OR rule.description: "Windows Logon Success"	0

Visualización de escalada de privilegios con el filtro y la cuenta.

### **Cambios en archivos sensibles**

Este panel recoge las modificaciones, eliminaciones o creaciones de archivos críticos en los sistemas cliente. Se trata de una funcionalidad integrada en Wazuh mediante el módulo **FIM (File Integrity Monitoring)**, que permite hacer seguimiento a cualquier cambio en rutas configuradas como sensibles.

La visualización está basada en los eventos generados por syscheck, agrupando por tipo de evento (modified, deleted, added) y la ruta del archivo. Esto ayuda a detectar cambios no autorizados en archivos como /etc/passwd o C:\Windows\System32\drivers\etc\hosts.

### **Conexiones sospechosas**

En este panel se recogen alertas relacionadas con conexiones de red inusuales o sospechosas, como por ejemplo el uso de puertos no estándar, accesos desde direcciones IP externas o patrones de escaneo. Se utiliza información de logs del sistema operativo y de eventos generados por Wazuh en sus reglas de detección.

Los datos se agrupan por dirección IP de origen (data.srcip), puerto (data.dstport) y nombre del agente. Esto permite correlacionar qué equipos han sido objetivo o han iniciado conexiones inusuales.

### **Accesos fallidos o repetidos**

Esta visualización tiene como objetivo identificar intentos de inicio de sesión fallidos, ya sea por errores de credenciales o por intentos de fuerza bruta. Se filtran eventos como "authentication\_fail" o descripciones que contengan "logon failure".

Se muestran gráficos de barras agrupados por usuario, equipo y número de intentos, lo que permite identificar ataques automatizados o intentos de acceso no autorizados.

### **Nuevos procesos ejecutados**

Este panel muestra la actividad relacionada con la ejecución de procesos en los equipos cliente. Resulta muy útil para detectar la ejecución de scripts o binarios que no deberían estar presentes, así como para identificar posibles infecciones por malware.

Se utilizan los campos data.process.name y data.process.cmd para mostrar tanto el nombre del proceso como la línea de comandos asociada. También se incluye información del usuario que lo ejecutó y del equipo origen.

## **Análisis de Logs y Eventos**

Una de las funcionalidades clave en el despliegue del sistema de monitorización ha sido la **recogida, análisis y correlación de logs** procedentes de las máquinas cliente. Gracias al uso de **Wazuh** junto con **Filebeat** y **Elasticsearch**, se ha conseguido centralizar la información generada por los equipos y convertirla en eventos comprensibles, accionables y visualizables mediante paneles.

### **Tipos de logs recogidos**

Los logs recogidos durante el proyecto proceden de diferentes fuentes y abarcan múltiples capas del sistema. A continuación se detallan los principales tipos de logs utilizados:

- **Logs del sistema operativo (Windows/Linux):** Información sobre inicio y cierre de sesiones, errores del sistema, bloqueos, cambios en usuarios, etc.
- **Eventos de seguridad de Windows:** Registro de autenticaciones, fallos de acceso, ejecución de procesos, etc., principalmente del canal Security.
- **Syslog (Linux):** Eventos del sistema relacionados con servicios, errores del kernel, autenticaciones y comandos ejecutados con sudo.
- **Logs de integridad (FIM):** Cambios detectados en archivos monitorizados por el módulo Syscheck de Wazuh.

## Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat

- **Logs de red:** Información relacionada con conexiones, puertos abiertos, tráfico inusual o escaneos.
- **Alertas de Wazuh:** Generadas por reglas correladas a eventos detectados, clasificadas por nivel de severidad y grupo.

Todos estos registros fueron recogidos por **Filebeat**, enviados a **Elasticsearch** y analizados en tiempo real por Wazuh mediante reglas personalizables.

### **Ejemplos de alertas importantes**

A lo largo de las pruebas del proyecto se identificaron y analizaron diversos eventos que pueden suponer un riesgo para la seguridad. Algunos ejemplos significativos fueron:

- **Escalada de privilegios:** Una alerta fue generada cuando se ejecutó un comando sudo en una máquina Linux. La alerta indicaba "Successful sudo to ROOT executed", con detalles del usuario y el comando lanzado.
- **Inicio de sesión exitoso con token elevado:** Wazuh detectó un inicio de sesión en Windows con un token elevado (indicando permisos de administrador), lo que fue registrado como "Windows Logon Success" con privilegios.
- **Acceso fallido repetido:** Varias alertas indicaron intentos fallidos de acceso desde la misma IP hacia un equipo Windows, lo que podría corresponderse con un intento de ataque por fuerza bruta.
- **Modificación de archivos sensibles:** Se registró una alerta cuando se editó el archivo /etc/hosts, considerado un fichero crítico por el módulo de FIM.
- **Conexiones internas inesperadas:** Se detectó tráfico entre equipos cliente que no debería haber existido según el diseño lógico de la red, lo que generó una alerta de conexión inesperada entre IPs.

### **Reacción ante incidentes simulados**

Para evaluar la eficacia del sistema se simularon varios incidentes, midiendo la capacidad de Wazuh para detectarlos y alertar adecuadamente:

- Se generaron múltiples accesos fallidos en un cliente Windows, lo cual fue detectado por Wazuh, y mediante las reglas configuradas se generó una alerta crítica que se reflejó en el panel de accesos.

## **Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat**

- Se realizó un intento de escalada de privilegios mediante ejecución de comandos como sudo su en una máquina Linux. Inmediatamente se registraron alertas en el dashboard de privilegios.
- Se probó el sistema de mensajería automática mediante Telegram y Gmail, enviando notificaciones en tiempo real cuando una alerta crítica se activaba. Esta funcionalidad demostró ser útil para respuestas tempranas ante eventos de seguridad.
- Se simuló cambios en archivos del sistema para comprobar la funcionalidad del módulo FIM. Se observaron alertas detalladas indicando qué archivo fue alterado, por quién y en qué momento.

Estas pruebas demostraron que el sistema puede actuar como una **primera línea de defensa**, permitiendo identificar eventos sospechosos, prevenir daños mayores y proporcionar evidencia para análisis forenses.

## **Seguridad**

La seguridad ha sido un eje fundamental en el desarrollo e implementación de este proyecto. Dado que el objetivo principal era monitorizar y proteger la infraestructura virtualizada, se implementaron múltiples medidas orientadas al endurecimiento del sistema (hardening), la protección de la red y una correcta gestión de usuarios y privilegios.

### **Prácticas de endurecimiento aplicadas (Hardening)**

El **hardening del sistema operativo** y de los servicios expuestos ha sido una prioridad desde las fases iniciales del proyecto. Las principales medidas aplicadas incluyen:

- **Desactivación de servicios innecesarios:** En las máquinas Linux y Windows se eliminaron o detuvieron servicios no esenciales para reducir la superficie de ataque.
- **Actualización del sistema:** Todos los sistemas se actualizaron antes de la implementación, minimizando vulnerabilidades conocidas.
- **Configuración de permisos de archivos y directorios:** Se revisaron los permisos en rutas críticas como /etc, /var/ossec, C:\Windows\System32, entre otras.
- **Acceso restringido por SSH y RDP:** Se limitó el acceso mediante firewall y se impusieron medidas como autenticación por clave pública en SSH.
- **Auditoría y monitorización:** A través de Wazuh se activaron módulos de FIM, rootcheck, SCA y auditoría de comandos sudo, permitiendo alertar sobre

## Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat

configuraciones inseguras y comportamientos sospechosos.

- **Bloqueo de cuentas tras múltiples fallos de acceso:** En Linux se utilizaron módulos como pam\_tally2 y en Windows se ajustaron directivas de seguridad local.
- **Integración de antivirus (opcional):** Se contempló la posibilidad de usar ClamAV o Windows Defender, aunque no fue el foco principal.

Estas acciones básicas pero efectivas ayudan a garantizar una configuración más segura y resistente ante ataques comunes.

### **Seguridad de la red virtual**

El entorno de red fue cuidadosamente diseñado para **minimizar la exposición** de los servicios críticos y garantizar la segmentación adecuada:

- **Uso de OPNsense como firewall:** Se configuraron reglas para restringir tráfico, permitir solo comunicaciones necesarias y bloquear accesos externos no autorizados.
- **Separación de redes:** Se utilizaron interfaces y subredes independientes para clientes, servidor y acceso a Internet.
- **Activación de NAT y DHCP controlado:** A través de OPNsense se gestionó el direccionamiento IP y el reenvío de paquetes con políticas específicas.
- **Bloqueo de puertos innecesarios:** Solo se mantuvieron abiertos puertos como 5601 (Kibana), 9200 (Elasticsearch) y 1514/1515 (Wazuh agent).
- **Revisión de logs de red:** Se supervisan logs del firewall y conexiones entre clientes para detectar posibles comportamientos anómalos.
- **Simulación de ataque de red:** Durante las pruebas se realizaron escaneos con herramientas como nmap desde clientes comprometidos para validar la eficacia del firewall.

Estas configuraciones buscan **prevenir accesos no deseados** y asegurar que cada nodo sólo tenga visibilidad de lo estrictamente necesario.

### **Gestión de usuarios y privilegios**

Una gestión correcta de los usuarios fue crucial para garantizar que cada componente del sistema actuase bajo el principio de **mínimos privilegios**:

- **Usuarios limitados en las máquinas cliente:** Se crearon cuentas de usuario estándar sin privilegios administrativos en los clientes Windows y Linux.
- **Registros de inicio de sesión y uso de sudo:** Cualquier uso de comandos administrativos era registrado y alertado a través de Wazuh.
- **Control del grupo sudoers en Linux:** Solo el usuario de administración tenía acceso para ejecutar comandos como root.
- **Monitorización de eventos de logon en Windows:** A través del Event ID 4624 y 4625 se detectaron accesos válidos y fallidos, asociados a usuarios reales.
- **Notificaciones por Telegram:** En los eventos críticos de elevación de privilegios o accesos anómalos, se enviaban alertas automáticamente a un canal seguro.

Gracias a esta política de control y supervisión de cuentas, se pudo identificar con precisión quién realizó cada acción importante en el sistema..

### **Implementación del Sistema de Mensajería: Telegram y Gmail**

Con el objetivo de reforzar la capacidad de respuesta ante eventos de seguridad, se ha integrado un sistema de **notificaciones automáticas** basado en los canales de **Telegram y Gmail**. Estas herramientas permiten alertar en tiempo real a los administradores del sistema ante la detección de comportamientos sospechosos, accesos fallidos o escaladas de privilegios, mejorando significativamente el tiempo de reacción.

#### **Telegram como canal de alerta inmediata**

La integración con Telegram se ha llevado a cabo mediante la creación de un **script en Bash** que utiliza la API de bots de Telegram. Este script se ejecuta como respuesta activa (active-response) desde Wazuh cuando se dispara una alerta crítica.

#### **Funcionamiento resumido:**

- El script extrae los campos más relevantes de la alerta generada (nivel, agente, descripción, etc.).

## **Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat**

- Construye un mensaje formateado.
- Lo envía mediante curl a través de la API HTTPS de Telegram, utilizando el token del bot y el identificador del chat.
- El mensaje aparece instantáneamente en el chat de Telegram definido.

Esto permite al administrador recibir alertas en su dispositivo móvil o escritorio sin necesidad de acceder al panel de control de Wazuh o Kibana.

### **Gmail como sistema complementario de notificación**

Además de Telegram, se ha configurado el sistema para enviar alertas a través del **correo electrónico usando Gmail**. Para ello, se ha utilizado msmtplib, una herramienta ligera de envío de correo desde la línea de comandos.

#### **Pasos realizados:**

- Configuración del fichero .msmtplib con los datos del servidor SMTP de Gmail, usuario, puerto y autenticación TLS.
- Creación de un script de respuesta activa (gmail-alert.sh) que construye un mensaje basado en la alerta recibida.
- Integración del script en el sistema de alertas de Wazuh mediante el archivo ossec.conf.
- Definición de los niveles de severidad que activan el envío del correo.

Con esta integración, cada vez que se produce una alerta con nivel crítico o superior, se envía un correo al administrador, conteniendo el mensaje, el nombre del agente afectado y detalles del evento.

### **Ventajas del sistema dual de mensajería:**

- **Redundancia:** Si un canal falla (por ejemplo, la API de Telegram), el correo electrónico sigue funcionando.
- **Versatilidad:** Los usuarios pueden consultar los mensajes en diferentes dispositivos y aplicaciones.



## Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat

- **Auditoría:** Los correos quedan registrados como evidencia adicional de las alertas ocurridas.

Ambos sistemas se complementan, ofreciendo **una solución de notificación robusta, multiplataforma y en tiempo real**, alineada con los objetivos del proyecto: mejorar la monitorización, respuesta y gestión de incidentes.

### **Problemas detectados durante la integración**

Durante el desarrollo del proyecto y la integración entre los distintos componentes (clientes, servidor Wazuh, Filebeat, Elasticsearch y sistemas de mensajería), se identificaron diversos problemas que dificultaron el correcto funcionamiento del sistema. A continuación, se detallan los principales:

### **Comunicación entre los agentes y el servidor Wazuh**

#### **Estado actual**

Los agentes de Wazuh se registran correctamente en el servidor, y este los reconoce con sus respectivos IDs. Sin embargo, no se reciben logs de estos agentes en el servidor y, por tanto, no aparecen eventos en el panel de control de Wazuh.

```
root@servidor-wazuh1:/home/servidor# /var/ossec/bin/agent_control -l

Wazuh agent_control. List of available agents:
  ID: 000, Name: servidor-wazuh1 (server), IP: 127.0.0.1, Active/Local
  ID: 007, Name: DESKTOP-EKNBLKK, IP: any, Active
  ID: 009, Name: DESKTOP-UP9357F, IP: any, Active
  ID: 008, Name: DESKTOP-JMTPOJT, IP: any, Active

List of agentless devices:
```

Listado de los agentes activos

#### **Síntomas observados**

- El archivo alerts.json sí registra eventos generados localmente en el servidor, pero no aparecen logs provenientes de agentes remotos.
- Los índices de las alertas en ElasticSearch están vacíos.
- En la interfaz de Kibana (Wazuh UI), no aparecen eventos, lo que impide generar alertas o paneles útiles.

#### **Posibles causas**

## Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat

- Fallos en el envío de datos desde Filebeat a Elasticsearch.
- El índice wazuh-alerts-\* no está siendo creado o poblado correctamente.

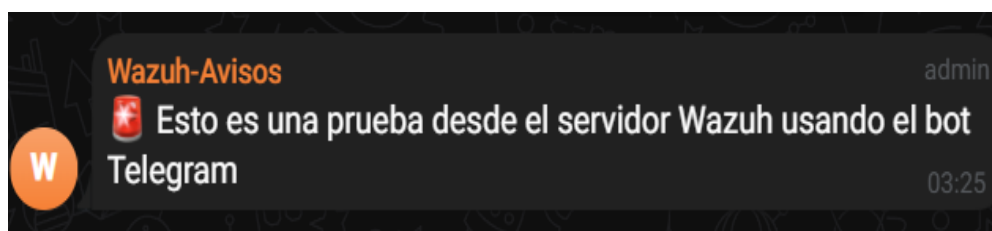
### **Automatización de alertas por mensajería (Telegram y Gmail)**

#### **Estado actual**

Tanto el script de Telegram como el de Gmail funcionan correctamente al ejecutarse manualmente desde terminal. Sin embargo, cuando se intenta activar estos scripts como respuestas automáticas a alertas de seguridad mediante el sistema de “Active Response”, no se ejecutan.



Mail recibido de prueba.



Mensaje recibido vía telegram de prueba.

#### **Síntomas observados**

- Los Scripts manuales (telegram.sh y gmail.sh) funcionan perfectamente y envían mensajes.
- No se observan ejecuciones automáticas tras generar una alerta real.
- En el archivo active-responses.log no siempre se registran intentos de ejecución.

### **Problemas con el dashboard (Wazuh UI)**

#### **Estado actual**

El dashboard muestra datos, por que si que se han llegado a recibir un mínimo pero a partir de un punto ya sin recibir lo datos el dashboard queda completamente vacío, esto también es por un problema en los índices de Wazuh.

#### **Síntomas observados**

- La consulta al índice de Elasticsearch indica que wazuh-alerts está vacío o no existe.
- No se muestran gráficos ni alertas en el dashboard.

#### **Posibles causas**

- Filebeat no está recogiendo correctamente los logs de Wazuh.

### **Conclusión de las incidencias**

Estas incidencias se deben principalmente a la **complejidad de la integración entre múltiples servicios**, cada uno con configuraciones sensibles (certificados, rutas, permisos, etc.). Las pruebas demostraron que los componentes por separado funcionan, pero el desafío principal fue **lograr una integración estable** que permita que las alertas generadas por los agentes desencadenen acciones automáticas y que se visualice en tiempo real en el dashboard.

### **Pruebas Realizadas**

Para validar la eficacia del sistema de monitorización y seguridad desplegado, se realizaron una serie de **pruebas controladas** que simulaban diferentes tipos de incidentes de seguridad. El objetivo fue observar cómo reaccionaba **Wazuh, Filebeat, Elasticsearch y OPNsense** ante situaciones reales, y comprobar si eran capaces de detectar y alertar correctamente sobre actividades sospechosas o maliciosas. Dado a los problemas que hemos hablado anteriormente estas pruebas se han realizado pero la mayoría de ellas los datos obtenidos no se han podido visualizar desde el Dashboard de Wazuh, pero si se han comprobado que los logs han llegado.

### **Simulación de un ataque**

Durante las pruebas, se llevaron a cabo diferentes tipos de ataques simulados desde las máquinas cliente. Algunos de los más relevantes fueron:

## **Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat**

- **Modificación de archivos sensibles:** Se editó manualmente el archivo /etc/shadow en una máquina Linux. El módulo de FIM (File Integrity Monitoring) de Wazuh detectó el cambio y emitió una alerta con la ruta exacta y el usuario que lo realizó.
- **Escalada de privilegios:** En una máquina Windows se intentó acceder como root mediante sudo su. Wazuh generó una alerta inmediata, indicando que un usuario había ejecutado un comando con privilegios de administrador, categorizado como "Successful sudo to ROOT executed".

### **Detección desde Wazuh**

La función de Wazuh aquí es la de en base a las reglas que creamos, los ataques se agrupan por severidad (baja, media, alta, crítica) y por categoría (autenticación, escalada, red, integridad, etc.).

Además, las visualizaciones creadas en los dashboards de Kibana deben permitir ver de forma clara y gráfica cuándo y dónde se habían producido estas actividades, y qué agente estaba implicado.

### **Respuesta en el firewall**

Aunque Wazuh actuó como sistema de detección, el firewall OPNsense se configuró para dar respuesta activa y protección perimetral. Algunas de las medidas tomadas fueron:

- **Bloqueo de IPs sospechosas:** Al detectar múltiples accesos fallidos, se aplicaron reglas manuales en OPNsense para bloquear temporalmente la IP atacante.
- **Restricción de puertos:** Se limitaron los puertos abiertos a servicios necesarios, denegando por defecto conexiones entrantes no autorizadas.
- **NAT y segmentación de red:** Se utilizaron reglas de traducción de direcciones y aislamiento por subred para evitar que una máquina comprometida pudiera acceder a otras.
- **Monitoreo del tráfico:** Se habilitó la inspección del tráfico desde la interfaz LAN, registrando intentos de escaneo o conexiones extrañas entre hosts.

### **Resultados Obtenidos**

Durante el desarrollo del proyecto se ha conseguido desplegar con éxito la infraestructura base formada por una red virtualizada compuesta por una máquina OPNsense como cortafuegos, un servidor con Wazuh Manager y varias máquinas cliente con el agente de Wazuh instalado. El dashboard de Wazuh ha sido accesible y funcional tras la instalación, y se verificó la correcta autenticación de los agentes desde el servidor.

## **Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat**

### **Logros alcanzados:**

- Se logró establecer comunicación entre el servidor Wazuh y los agentes cliente. Los agentes se registraron correctamente en el servidor y su estado fue visible desde el dashboard.
- Se configuraron múltiples fuentes de logs, incluyendo dpkg.log, journald, comandos periódicos y archivos personalizados.
- Se creó una integración de mensajería mediante scripts personalizados para enviar alertas a Telegram y Gmail, funcionando correctamente en pruebas manuales.
- Se generaron alertas localmente y se comprobaron registros en alerts.json, confirmando que el sistema de detección del manager estaba activo.

### **Problemas encontrados:**

- A pesar de que los agentes se conectan correctamente al servidor, los eventos generados no llegaban al dashboard. Tras múltiples pruebas se identificó que las alertas sí estaban siendo generadas y registradas en alerts.json, pero no estaban siendo indexadas correctamente en Elasticsearch, impidiendo que se visualizarán en la interfaz web.
- Las consultas a Elasticsearch no devolvían resultados, lo cual confirmó que los índices wazuh-alerts no estaban siendo creados.
- En cuanto a la mensajería automatizada, los scripts para enviar notificaciones funcionaban correctamente si se ejecutaban manualmente desde la terminal. Sin embargo, al integrarlos en el sistema de respuestas activas, no se ejecutaban al generar una alerta.

### **Conclusión de esta sección:**

Los resultados muestran que la detección de eventos en el host funciona correctamente, y que los scripts de notificación son válidos, pero existe una desconexión crítica entre el módulo de análisis y el dashboard. Esto refleja la necesidad de verificar a fondo la integración del indexer y asegurarse de que los datos fluyan correctamente desde alerts.json hasta Elasticsearch. Estos fallos, aunque frustrantes, permiten identificar los puntos clave que hay que revisar en un despliegue real y mejoran el entendimiento profundo del funcionamiento interno de Wazuh.

**Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades  
Sospechosas con ELK + Filebeat**

**Limitaciones detectadas**

A pesar del éxito global del sistema, se detectaron algunos aspectos mejorables:

- **Uso elevado de recursos:** Elasticsearch requiere una cantidad significativa de memoria RAM. En entornos con recursos limitados, puede provocar cuelgues si no se gestiona adecuadamente el heap o el índice.
- **Problemas con campos .keyword:** Algunas visualizaciones no funcionaron de forma óptima por la falta de campos tipo keyword, lo que obligó a adaptar filtros de manera menos precisa.
- **Configuración inicial compleja:** La integración entre Filebeat, Wazuh y Kibana, junto con la necesidad de certificados y plantillas personalizadas, puede dificultar la puesta en marcha si no se sigue un procedimiento claro.
- **Telegram sin soporte oficial:** Aunque funcional, el sistema de alertas vía Telegram no es una característica nativa de Wazuh, por lo que requiere scripts personalizados y pruebas manuales para asegurar su operatividad.

A pesar de estas limitaciones, el sistema ofreció **gran valor a nivel educativo y funcional**, cumpliendo con los objetivos del proyecto y permitiendo su aplicación real en entornos de pequeña y mediana empresa.

**Conclusiones y Trabajo Futuro**

**Lecciones aprendidas**

Durante el desarrollo de este proyecto se han adquirido conocimientos esenciales sobre **infraestructura de red segura, virtualización, sistemas de monitorización centralizada**, y especialmente sobre la **integración de herramientas complejas** como OPNsense, Wazuh, Filebeat, Elasticsearch y Kibana. Entre los aprendizajes más relevantes destacan:

- La **configuración precisa de agentes y rutas de datos** es clave para asegurar que la monitorización sea efectiva.
- La **automatización de respuestas** mediante scripts personalizados (como el sistema de alertas por Telegram) puede mejorar significativamente la capacidad de reacción ante incidentes.

## Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat

- La correcta **visualización de eventos** mediante dashboards no solo aporta control, sino que permite detectar patrones ocultos.
- Es fundamental contar con **una planificación detallada de recursos**, ya que herramientas como Elasticsearch tienen un consumo elevado si no se dimensionan correctamente.

Además, se ha evidenciado la importancia de una **documentación clara y secuencial**, ya que muchas de las configuraciones no están completamente automatizadas y requieren intervención manual.

### **Posibles mejoras**

Aunque el sistema ha demostrado ser eficaz, existen varias mejoras que podrían implementarse para optimizar y escalar la solución:

- **Automatización del despliegue** mediante herramientas como Ansible o scripts en Bash que faciliten la instalación y configuración de Filebeat, certificados y plantillas.
- **Creación de plantillas personalizadas optimizadas**, para evitar errores relacionados con tipos de datos y mejorar el soporte de campos .keyword.
- **Alertas multi-canal**, integrando además de Telegram, correo electrónico (Gmail) o plataformas como Slack o Microsoft Teams.
- **Dashboards avanzados** con visualizaciones que correlacionen múltiples eventos o comportamientos anómalos por usuario, equipo o rango horario.

### **Aplicabilidad en entornos reales e integración de Inteligencia Artificial**

Una de las ideas más destacadas y de gran proyección es la **integración de inteligencia artificial (IA)** dentro del sistema de monitorización. Este proyecto abre la puerta a **aplicaciones reales en entornos productivos**, como por ejemplo:

- **Análisis automatizado de logs mediante IA**: Se podrían usar modelos de machine learning o redes neuronales ligeras para detectar patrones inusuales o sospechosos que escapen a las reglas tradicionales.
- **Clasificación inteligente de alertas**: La IA puede priorizar eventos no solo por nivel de severidad, sino también por correlación con otros eventos históricos, aprendizaje del comportamiento del usuario o anomalías de red.

## Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades Sospechosas con ELK + Filebeat

- **Respuestas automáticas adaptativas:** Mediante IA, el sistema podría determinar si un evento requiere bloqueo inmediato, monitoreo pasivo o generación de informe para revisión humana.
- **Predicción de incidentes:** Con suficiente historial de alertas, sería posible anticipar posibles vectores de ataque o vulnerabilidades recurrentes.

Estos elementos, combinados con el uso actual de Telegram y el potencial añadido de Gmail, constituyen una evolución hacia un **SIEM** más avanzado, inteligente y autónomo.

La aplicación en entornos reales no solo es viable, sino deseable, ya que proporciona un **entorno seguro, reactivo y escalable**, adaptado a las necesidades actuales de ciberseguridad en redes empresariales.

### **Índice De Imágenes:**

Red Opnsense.....	9
Red Servidor.....	10
Red cliente 1€.....	10
Red cliente 2.....	11
Red cliente 3.....	11
El Dashboard de Wazuh con los agentes y su información.....	14
Visualización de agentes activos.....	16
Visualización de alertas de seguridad.....	17
Visualización de escalada de privilegios con el filtro y la cuenta.....	17
Listado de los agentes activos.....	25
Mail recibido de prueba.....	26
Mensaje recibido vía telegram de prueba.....	26



**Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades  
Sospechosas con ELK + Filebeat**

**Enlace GitHub:**

<https://github.com/Freitas04/proyecto-asir-wazuh-HectorRuiz.git>

En el enlace de github no encontraremos estos archivos:

- `ossec.conf`: Configuración principal de Wazuh Manager.
- `local\_rules.xml`: Reglas personalizadas.
- `local\_decoder.xml`: Decodificadores adicionales.
- `telegram.sh`: Script para notificaciones por Telegram.
- `gmail.sh`: Script para notificaciones por correo.
- `msmtprc`: Configuración SMTP para Gmail.
- `test log`: Archivo de pruebas de eventos manuales.
- `ossec.log`: Log principal de Wazuh.
- `alerts.json`: Registro de alertas.

También encontraremos este documento y las diapositivas de la presentación (es posible que cambien de el día de entrega al día de presentacion).

**Proyecto 2ºASIR - Curso 24/25 - Monitorización y Detección de Actividades  
Sospechosas con ELK + Filebeat**

***Bibliografía***

Alertas Gmail:

[https://www.youtube.com/watch?v=dSHJ\\_u02qGc&t=304s](https://www.youtube.com/watch?v=dSHJ_u02qGc&t=304s)

<https://www.youtube.com/watch?v=kH8tfraVzFk&t=52s>

Creación Dashboard Wazuh:

<https://www.youtube.com/watch?v=D8jZRM962Zk&t=218s>

<https://documentation.wazuh.com/current/user-manual/wazuh-dashboard/creating-custom-dashboards.html>

Instalación ELK:

[https://www.youtube.com/watch?v=roJQ-7F\\_Vgg](https://www.youtube.com/watch?v=roJQ-7F_Vgg)

Instalación Wazuh:

<https://www.youtube.com/watch?v=Ag6lIMxBFZ4&t=224s>

Descarga Opnsense:

<https://opnsense.org/>

Descarga Wazuh:

<https://documentation.wazuh.com/current/installation-guide/wazuh-agent/wazuh-agent-package-linux.html>

<https://documentation.wazuh.com/current/installation-guide/wazuh-agent/index.html>

Descarga de Ubuntu:

[Download Ubuntu Desktop | Ubuntu](#)