Sistema de prevención de ataques y sistema de mensajería automatizada

I.E.S. La Vereda

La Pobla de Vallbona - València

UNIÓN EUROPEA
Fondo Social Europeo

Héctor Rafael Ruiz Freitas - 2ºASIR - 2025
Administración de Sistemas Informáticos en Red
IES La Vereda - Marc Crespo Costa

Este proyecto presenta el diseño e implementación de una infraestructura de red segura y monitorizada para una pequeña empresa, basada en tecnologías de virtualización y herramientas de seguridad de código abierto. La solución integra un firewall OPNsense, un servidor central de monitorización Wazuh y varios equipos cliente, garantizando la supervisión continua del sistema, la detección de amenazas y la respuesta activa ante incidentes. Además, se implementan sistemas de mensajería automatizada mediante Telegram y correo electrónico, junto con procedimientos de hardening y análisis de logs. El proyecto demuestra una arquitectura escalable, eficiente y adaptable a entornos corporativos reales, cumpliendo con los objetivos de seguridad, control y mantenimiento preventivo.

Sistema de prevención de ataques y sistema de mensajería automatizada

Héctor Rafael Ruiz Freitas

Índice:

1 INTRODUCCIÓN

1.1 MÓDULOS A LOS QUE IMPLICA

En este proyecto se han integrado prácticamente todos los módulos del ciclo formativo, destacando especialmente: Planificación y Administración de Redes, Seguridad y Alta Disponibilidad, Servicios de Red e Internet, Administración de Sistemas Operativos y Lenguajes de Marcas. La solución combina el despliegue de una infraestructura realista con OPNsense, Wazuh, servicios de mensajería, firewalling, automatización y monitorización avanzada, aplicando de forma directa los contenidos vistos en clase.

1. Planificación y Administración de Redes

- **Diseño de la topología de red virtual**, incluyendo redes internas, redes Host-Only y acceso WAN.
- **Direccionamiento IPv4**, creación de subredes /24, definición de gateways y DNS.
- **Integración de OPNsense como firewall y router**, configurando NAT, reglas de cortafuegos y servicios de DHCP.
- **Configuración segura de dispositivos de red**, aplicando acceso por SSH y buenas prácticas.
- **Captura y análisis de tráfico con tcpdump**, verificación del flujo de datos entre agentes y servidor Wazuh.

Este módulo ha sido clave para garantizar que la infraestructura soporte correctamente la monitorización, las alertas y el envío de datos entre los distintos nodos.

---

2. Implantación de Sistemas Operativos

El proyecto ha incluido múltiples instalaciones y configuraciones de sistemas:

- Instalación de **Ubuntu Server** como servidor principal de Wazuh.
- Instalación de **Ubuntu Desktop** como cliente.
- Gestión de servicios mediante **systemd**.
- Configuración de redes persistentes con Netplan.
- **Aseguramiento del sistema**, aplicando usuarios, permisos y endurecimiento básico.
- Resolución de incidencias derivadas de configuraciones de red, servicios y paquetes.

Este módulo se refleja en la correcta implantación y operatividad de cada nodo dentro del entorno virtual.

---

3. Fundamentos de Hardware

Aunque el proyecto es virtual, se han aplicado conceptos fundamentales:

- **Selección y configuración de recursos hardware virtuales** (CPU, RAM, almacenamiento).
- Análisis del rendimiento del servidor anfitrión para ejecutar múltiples máquinas.
- Organización eficiente de las imágenes y máquinas virtuales.

Estos conocimientos han permitido dimensionar correctamente la infraestructura para que Wazuh funcione con fluidez.

---

4. Gestión de Bases de Datos

Wazuh utiliza bases de datos internas para almacenar:

- Eventos de seguridad
- Indicadores de integridad
- Información de estado de los agentes

Se han aplicado conceptos como:

- Ficheros estructurados y consultas internas mediante la API.
- Gestión de índices dentro de Wazuh Indexer (basado en OpenSearch).
- Supervisión del almacenamiento y rendimiento del indexador.

Aunque no se ha realizado SQL directamente, sí se trabajan conceptos esenciales de gestión y optimización.

---

5. Lenguajes de Marcas y Configuración

Gran parte del proyecto se apoya en lenguajes de marcas:

- Configuración extensa mediante **XML** en los archivos `ossec.conf`.
- Modificaciones en YAML (Filebeat, Netplan, integraciones).
- Interpretación de JSON a través de la API de Wazuh.

Este módulo es fundamental para comprender y editar correctamente los ficheros de configuración del sistema.

---

6. Administración de Sistemas Operativos en Red

Este módulo es uno de los pilares del proyecto:

- Automatización con **scripts en Python** para envío de alertas a Telegram.
- Administración remota por SSH desde el host al servidor.
- Integración de distintos sistemas operativos en red.
- Gestión y monitorización de procesos del sistema.
- Uso de servicios como **OpenSSH, systemd, journald**, etc.

Este módulo conecta directamente con la administración avanzada realizada en Wazuh y en el firewall.

---

7. Servicios de Red e Internet

El proyecto implementa varios servicios reales:

- **DHCP y DNS** en OPNsense.
- **Servidor de correo Postfix** para el envío de alertas por Gmail.
- **Servicios web** a través del dashboard de Wazuh.
- **Servicio de mensajería** mediante la API de Telegram.
- Configuración de puertos, cortafuegos y NAT para permitir la comunicación entre nodos.

---

8. Seguridad y Alta Disponibilidad

Uno de los módulos más importantes en relación con el proyecto:

- Implementación de un **IDS/IPS** mediante Wazuh.
- Configuración de **Active Response** para bloquear IPs sospechosas.
- Uso del firewall de OPNsense como barrera perimetral.
- Monitorización de logs, integridad de archivos y auditorías.
- Seguridad en comunicaciones (SSH, HTTPS).
- Gestión de alertas de seguridad en tiempo real (Telegram/Gmail).
- Simulación de ataques y observación de la reacción del sistema.

El proyecto aplica casi todos los elementos del módulo de seguridad.

---

9. Implantación de Aplicaciones Web

Wazuh Dashboard es una aplicación web completa. En este módulo se han aplicado:

- Despliegue y configuración del dashboard.
- Gestión de usuarios y autenticación.
- Conexión entre la aplicación web y el backend (Wazuh Manager / Indexer).

Aunque no se desarrolla una aplicación propia, sí se despliega una plataforma web profesional.

---

10. Administración de Sistemas Gestores de Bases de Datos

Aplicado a través de:

- Gestión del **Wazuh Indexer** (OpenSearch).
- Control de logs, índices y almacenamiento.
- Optimización de la ingestión de datos mediante Filebeat.

Indirectamente se han trabajado conceptos avanzados sobre bases de datos distribuidas y replicación interna.

1.2 BREVE DESCRIPCIÓN DEL PROYECTO

El proyecto consiste en el diseño e implementación de una infraestructura virtualizada orientada a la monitorización, análisis y protección de sistemas dentro de un entorno corporativo simulado. La arquitectura se compone de una red interna gestionada mediante OPNsense como cortafuegos y puerta de enlace, un servidor central de Wazuh encargado de la monitorización y correlación de eventos, y varios equipos cliente que simulan estaciones de trabajo de una empresa.

El objetivo principal es demostrar la capacidad de un administrador de sistemas para desplegar y asegurar una red, implementar soluciones de detección de intrusiones (IDS/IPS), automatizar respuestas mediante Active Response, integrar sistemas de notificación y mensajería (Telegram y Gmail) y generar un entorno capaz de identificar amenazas en tiempo real.

Objetivos y requisitos del proyecto

- Diseñar una red corporativa básica con direccionamiento IPv4 y servicios esenciales.
- Configurar un cortafuegos OPNsense que proporcione acceso a Internet, reglas de filtrado y NAT.
- Desplegar un servidor Wazuh 4.9 para:
    - Recoger logs en tiempo real.

    - Detectar accesos indebidos, escaladas de privilegios, cambios sospechosos en archivos y conexiones anómalas.
    - Gestionar agentes instalados en equipos Linux.
    - Ejecutar respuestas automáticas (bloqueo de IPs, reinicio de servicios, etc.).

- Implementar un sistema de alertas externas mediante Telegram y Gmail, integrados en Wazuh.
- Simular incidentes de seguridad (fuerza bruta, modificación de archivos, malware básico) y verificar la detección.
- Crear dashboards personalizados en Wazuh para visualizar eventos críticos.
- Documentar todo el proceso siguiendo criterios profesionales de administración de sistemas.

Uso del proyecto

El entorno creado permite a un administrador:

- Monitorizar en tiempo real el estado de todos los equipos clientes.
- Detectar comportamientos anómalos que pongan en riesgo la seguridad.
- Recibir alertas inmediatas en Telegram o correo electrónico.
- Analizar logs de forma centralizada para auditorías o investigación forense.
- Ejecutar acciones automáticas de mitigación sin intervención manual.
- Demostrar cómo se gestionaría la seguridad en una red corporativa real.

Este proyecto puede servir como base para redes de pequeñas o medianas empresas que requieran un sistema de monitorización accesible, eficaz y de bajo coste.

Tecnologías investigadas

Aunque parte del proyecto se apoya en contenidos estudiados en el ciclo formativo, se han ampliado conocimientos mediante tecnologías que no se profundizan habitualmente en clase:

- **Wazuh** como plataforma SIEM/IDS profesional.
- **OpenSearch / Indexer / Dashboard** como sustituto del anterior ELK.
- **Integraciones externas mediante API**, como bots de Telegram.
- **Active Response**, para automatizar respuestas ante incidentes de seguridad.
- **Certificados y comunicaciones seguras (SSL/TLS)** dentro del ecosistema Wazuh.
- **Configuración avanzada de OPNsense**, incluyendo:
    - Reglas de firewall adaptadas al tráfico del SIEM.
    - NAT para múltiples redes y máquinas virtuales.
- **Automatización mediante scripts Python usados para la integración con Telegram**.
- **Análisis de tráfico con tcpdump y herramientas de red adicionales**.
- **Hardening del sistema**, reglas de auditoría y supervisión avanzada.

2. ESTUDIO PREVIO

La empresa objeto de este proyecto es una pequeña organización que cuenta actualmente con dos equipos clientes, un servidor principal y un firewall basado en OPNsense como sistema de protección perimetral y encaminamiento.
Aunque su infraestructura es reducida, la empresa requiere un sistema profesional de monitorización, seguridad y gestión centralizada que permita prevenir incidentes, detectar amenazas en tiempo real y escalar fácilmente a medida que la organización crezca.

Hasta el momento, la empresa no disponía de un sistema unificado de supervisión ni de herramientas avanzadas de correlación de eventos, lo que dificulta la detección temprana de accesos indebidos, cambios no autorizados o actividad sospechosa en los equipos. El aumento de amenazas y la necesidad de garantizar la continuidad del negocio hacen necesaria la implantación de una solución completa y escalable.

En este contexto, se decide desplegar una infraestructura basada en:

- **OPNsense**, encargado de gestionar el tráfico de red, aplicar políticas de cortafuegos y facilitar la comunicación entre equipos internos.
- Un **servidor Ubuntu**, que actuará como nodo central sobre el cual se instalará **Wazuh**, solución de monitorización, análisis y respuesta ante incidentes.
- **Equipos clientes Ubuntu** que enviarán logs, alertas y métricas al servidor para permitir la supervisión global.
- Integraciones adicionales, como **notificaciones automáticas por Telegram** y **sistema de correo mediante Postfix**, con el fin de mejorar la capacidad de respuesta ante alertas críticas.

Aunque la topología actual es sencilla, la infraestructura está diseñada para ser **altamente escalable**, permitiendo:

- Añadir nuevos equipos cliente sin modificar la arquitectura base.
- Ampliar el número de servidores dedicados (por ejemplo, separando Wazuh Manager, Indexer y Dashboard).
- Implementar nuevas VLAN, servicios o políticas de seguridad en OPNsense.
- Integrar nuevas herramientas de análisis o automatización mediante IA.

El proyecto nace como respuesta a la necesidad de profesionalizar la seguridad, mejorar el control del sistema informático y sentar las bases para una infraestructura más robusta y preparada para el crecimiento futuro.

3. Soluciones propuestas

Para cubrir las necesidades detectadas en la empresa y garantizar un entorno seguro, monitorizado y escalable, se propone la implantación de una infraestructura basada en tecnologías libres y ampliamente utilizadas en entornos profesionales. Las soluciones adoptadas permiten no solo resolver las carencias actuales, sino también ofrecer una base sólida para futuras ampliaciones.

1. Implementación de un cortafuegos perimetral mediante OPNsense

OPNsense actúa como punto de entrada y salida de la red corporativa, permitiendo:

- **Control de acceso a Internet** mediante reglas firewall.
- **Segmentación de la red** y asignación de servicios (LAN, WAN).
- **Servicio DHCP** para los clientes.
- **Monitorización del tráfico** y detección de anomalías básicas.
- **Escalabilidad** ante la futura creación de VLANs o nuevas redes internas.

Esta solución asegura que toda la comunicación esté protegida desde la capa más baja.

---

2. Creación de un servidor central para monitorización y seguridad

El servidor principal utiliza **Ubuntu Server** y se encarga de ejecutar **Wazuh**, una plataforma SIEM/EDR que ofrece:

- Monitorización en tiempo real de eventos.
- Recolección y correlación de logs.
- Sistema de alertas ante comportamientos sospechosos.
- **Active Response**, permitiendo respuestas automáticas como bloquear IPs o reiniciar servicios.
- Integración con herramientas externas como **Telegram** y **Gmail** para notificaciones.

Esta solución garantiza control, seguridad y reacción ante incidentes de manera automatizada.

---

3. Instalación de agentes Wazuh en los clientes

En cada cliente se ha instalado un **agente Wazuh**, encargado de:

- Enviar logs y eventos al servidor.
- Detectar intrusiones locales.
- Registrar cambios en archivos sensibles.
- Informar de procesos iniciados, conexiones de red y accesos fallidos.
- Ejecutar respuestas automáticas cuando el servidor lo ordena.

Esto permite disponer de un control completo sobre el estado de cada dispositivo.

---

4. Integración de sistemas de mensajería (Telegram y Gmail)

Se han integrado dos métodos alternativos de comunicación:

- **Telegram**, mediante un bot personalizado, para envío inmediato de alertas críticas.
- **Postfix + Gmail**, permitiendo recibir alertas por correo electrónico desde el servidor.

Estas soluciones garantizan que el administrador reciba alertas incluso cuando no está frente al panel de control.

---

5. Infraestructura virtualizada mediante VirtualBox

El uso de VirtualBox permite:

- Simular un entorno corporativo completo.
- Crear servidores adicionales en un futuro sin coste adicional.
- Realizar pruebas de forma segura sin afectar a un entorno real.
- Escalar la topología simplemente clonando o añadiendo máquinas.

Esta virtualización facilita aprendizaje, pruebas y ampliación del proyecto.

---

6. Automatización y seguridad

Se han aplicado varias medidas y herramientas adicionales:

- **Hardening del servidor** (SSH, permisos, servicios mínimos).
- Scripts personalizados para integraciones externas.
- Reglas de firewall internas y externas.
- Monitorización de puertos, procesos y sistema de archivos.

4. DISEÑO

En este apartado se describe la arquitectura final del proyecto, tanto a nivel general como en detalle. El objetivo es mostrar cómo se estructura la red, qué componentes la forman y cómo interactúan entre sí para proporcionar monitorización, seguridad y capacidad de respuesta automática.

4.1 Diseño general (parte de análisis)

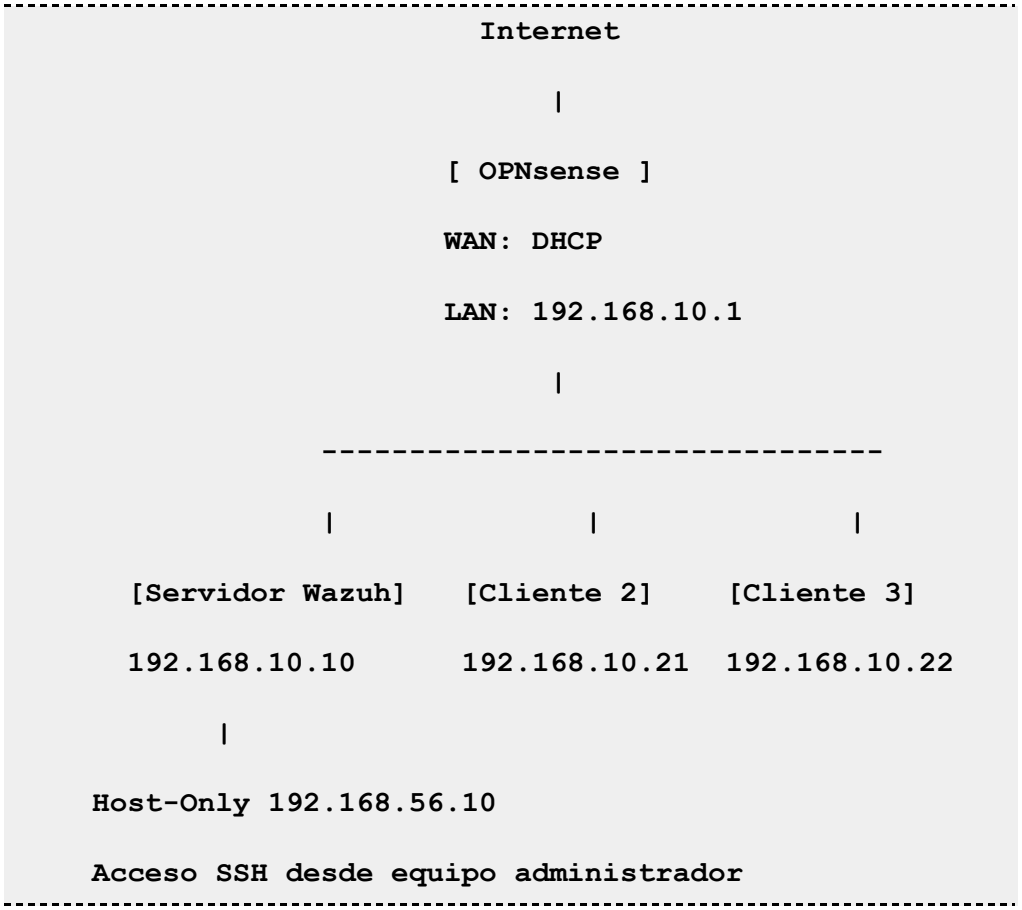
El proyecto se basa en una **infraestructura pequeña pero escalable**, compuesta por:

- **Un firewall OPNsense**, encargado de gestionar el tráfico entrante y saliente y actuar como puerta de enlace de la red interna.
- **Un servidor principal**, que aloja la plataforma **Wazuh 4.9** (manager, indexer y dashboard).
- **Dos equipos cliente Ubuntu**, en los que se instala el **Wazuh Agent** para la monitorización.
- **Una red interna (192.168.10.0/24)** para la comunicación entre equipos y con el firewall.
- **Una red Host-Only (192.168.56.0/24)** para permitir la administración por SSH desde el equipo personal del administrador.

El diseño general tiene como finalidad:

- Permitir que el servidor Wazuh centralice logs, alertas e integridad de los equipos cliente.
- Garantizar conectividad segura y segmentada.
- Asegurar que el firewall controla el acceso a Internet de la red interna.
- Proveer capacidad de escalado para añadir nuevos clientes sin modificar la estructura base.

Esquema conceptual de la arquitectura:



4.2 Diseño detallado

A continuación se expone el diseño detallado de cada componente y su función dentro del sistema.

4.2.1 Firewall OPNsense

- **Interfaz WAN:** DHCP → Permite salida a Internet.
- **Interfaz LAN:**
  - IP: **192.168.10.1/24**
  - Reparte direcciones a clientes si se habilita DHCP (en este proyecto se usa IP fija).
- **Reglas configuradas:**
  - Permitir tráfico interno LAN ↔ Servidor Wazuh.
  - Permitir salida a Internet para actualizaciones.
  - Bloqueo selectivo aplicado por Active Response (si procede en fases posteriores).

Funciona como frontera entre la red interna y el exterior.

4.2.2 Servidor Wazuh

- **IP principal:** 192.168.10.10 (LAN)
- **IP secundaria:** 192.168.56.10 (Host-Only para SSH)
- **Servicios instalados:**
    - **Wazuh Manager**
    - **Wazuh Indexer**
    - **Wazuh Dashboard**
    - **Filebeat (para enviar logs al Indexer)**

Responsabilidades del servidor:

- Recolectar e interpretar los logs de los agentes.
- Ejecutar Active Response cuando se detecten incidentes.
- Servir el dashboard gráfico al administrador.
- Almacenar y gestionar los datos de seguridad.

4.2.3 Equipos Cliente (Cliente2 y Cliente3)

- Sistemas operativos: Ubuntu
- Wazuh Agent versión 4.9 configurado.
- Cada cliente se comunica con el servidor mediante los puertos:
    - **1514/tcp** (Logs)
    - **1515/tcp** (Registro del agente)

Los agentes envían:

- Logs del sistema.
- Alertas de integridad de archivos.
- Detección de rootkits.
- Información de procesos, usuarios y conexiones.

4.2.4 Integración con Telegram (posterior)

El diseño prevé un script Python que permite:

- Enviar alertas críticas directamente a un **grupo de Telegram**.
- Utilizar un bot creado por BotFather.
- Integrar eventos de Active Response o logs críticos.

Este sistema mejora la capacidad de reacción en tiempo real ante incidentes.

4.2.5 Escalabilidad del diseño

Aunque la empresa simulada tiene pocos equipos, la arquitectura permite:

- Añadir fácilmente más clientes.
- Añadir más servidores dedicados (por ejemplo, un servidor de copias de seguridad).
- Crear VLANs para separar departamentos.
- Insertar reglas más avanzadas en OPNsense.

El diseño está preparado para crecer sin necesidad de rehacer la infraestructura.

5. IMPLANTACIÓN

En este apartado se describen todos los procesos realizados para la puesta en marcha del proyecto, detallando la instalación, configuración y validación de cada componente tanto en el servidor como en los clientes. Se explica también cómo repercute cada configuración en el funcionamiento global del sistema y su contribución a los objetivos del proyecto.

5.1 Implantación en el Servidor

La máquina servidor es el núcleo del sistema de monitorización y seguridad. Sobre ella se instala **Wazuh**, **Filebeat**, **Wazuh Indexer**, **Wazuh Dashboard** y los servicios complementarios.
Además, el servidor actúa como punto central de gestión de agentes, recepción de logs, ejecución de Active Response y envío de alertas (Telegram, Gmail, etc.).

A continuación, se detallan los procesos llevados a cabo:

5.1.1 Instalación del sistema operativo y configuración inicial

- Se instaló **Ubuntu Server 24.04 LTS** en una VM dedicada.
- Se crearon **dos interfaces de red**:
  - `enp0s3` → Red interna (VLAN 192.168.10.0/24)
  - `enp0s8` → Adaptador solo-anfitrión para SSH desde el equipo físico
- Se configuró el archivo `/etc/netplan/00-installer-config.yaml` con direcciones IP estáticas.
- Se habilitó el acceso SSH desde el equipo administrador.

Objetivo: Garantizar que el servidor sea accesible y estable en la red local.

5.1.2 Instalación del stack de Wazuh (manager, indexer, dashboard)

Se añadieron los repositorios oficiales de Wazuh y se instaló la versión **4.9.2**, que es la misma que los agentes para evitar incompatibilidades.

Componentes instalados:

- **Wazuh Manager** → gestiona agentes, reglas, active response
- **Wazuh Indexer** → motor de indexación (basado en OpenSearch)
- **Wazuh Dashboard** → interfaz gráfica
- **Filebeat** → envía los logs del manager al indexer

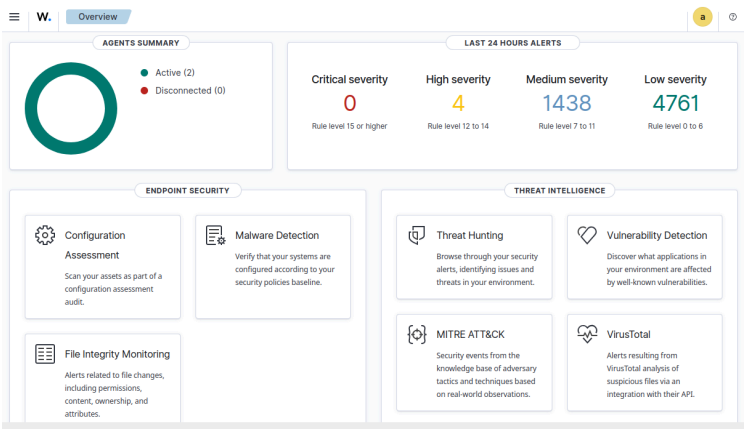
Pasos realizados:

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo apt-key add -

echo "deb https://packages.wazuh.com/4.x/apt stable main" | sudo tee /etc/apt/sources.list.d/wazuh.list

sudo apt update

sudo apt install wazuh-manager=4.9.2-1 wazuh-indexer=4.9.2-1 wazuh-dashboard=4.9.2-1 filebeat -y
```

Tras la instalación:

- Se habilitaron los servicios para iniciar con el sistema.
- Se arrancaron todos los componentes y se verificó su estado.

Objetivo: Dejar operativo un entorno de monitorización completo y coordinado.



Dashboard Wazuh

5.1.3 Configuración del Wazuh Manager

Se editaron parámetros del fichero:

- Altas y gestión de agentes
- Configuración de Active Response
- Añadir integraciones externas (Telegram, Gmail)
- Whitelists
- Política de logs

Ficheros editados:

- `/var/ossec/etc/ossec.conf`
- `/var/ossec/etc/client.keys`

Contribución al sistema: El manager es el cerebro del sistema y debe estar perfectamente configurado para recibir, procesar y responder ante eventos de seguridad.

5.1.4 Integración con Telegram (alertas)

Se creó un bot desde **BotFather**, se recuperó el `TOKEN` y el `CHAT_ID`.

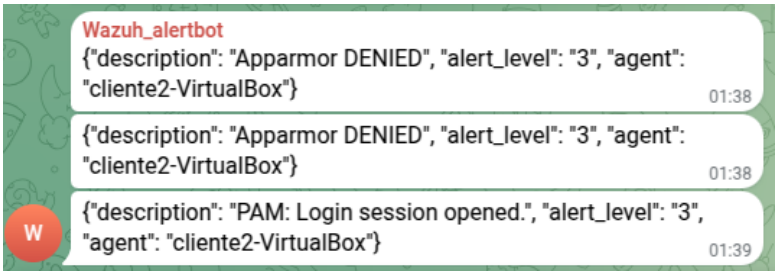
Para enviar alertas se creó el script:

```
/var/ossec/integrations/telegram.py
```

El cual permite a Wazuh enviar automáticamente mensajes a Telegram en función de alertas específicas (por ejemplo, escaladas de privilegios, accesos fallidos, modificaciones de archivos críticos).

Se añadieron los bloques `<integration>` al archivo ossec.conf y se recargó el servicio.

Objetivo: Garantizar que el sistema notifique incidentes en tiempo real.



Ejemplo notificación telegram

5.1.5 Integración con Gmail (Postfix + SMTP)

- Se instaló y configuró **Postfix** como "Internet con smarthost".
- Se añadió el servidor SMTP de Gmail.
- Se implementó autenticación segura mediante `sasl_passwd`.

Wazuh puede enviar alertas críticas por correo en caso de que el canal de Telegram no esté disponible.

Aviso notificación Gmail

5.1.6 Implementación de Active Response

Se activaron comandos como:

- `firewall-drop` → Bloqueo automático de IPs
- `host-deny`
- `restart-wazuh`

Además, se configuraron reglas para que Wazuh ejecute respuestas automatizadas cuando un cliente detecta:

- Fuerza bruta

```
<!-- Bloqueo automático ante ataques (Linux) -->
  <active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>5710, 5712, 1002</rules_id>
    <timeout>600</timeout>
  </active-response>
```

```
<!-- Bloqueo en firewall Linux -->
  <command>
    <name>firewall-drop</name>
    <executable>firewall-drop</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>
```

Bloqueo Automático de Ataques Active Response

- Reinicio de Wazuh

```
<!-- Reinicio de agente ante problemas -->
 <active-response>
    <command>restart-wazuh</command>
    <location>local</location>
    <rules_id>100009</rules_id>
  </active-response>
```

```
<!-- Reiniciar agente/manager -->
  <command>
    <name>restart-wazuh</name>
    <executable>restart-wazuh</executable>
  </command>
```

Reinicio Wazuh Active Response

- Múltiples accesos fallidos:

```
<!-- Ruta nula ante múltiples accesos fallidos -->
 <active-response>
   <command>route-null</command>
   <location>local</location>
   <level>7</level>
   <timeout>600</timeout>
  </active-response>
```

```
<!-- Ruta nula Linux -->
  <command>
    <name>route-null</name>
    <executable>route-null</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>
```

Múltiples accesos fallidos Active Response

- Deshabilitar cuenta:

```
<!-- Deshabilitar cuenta linux -->

  <active-response>
    <command>disable-account</command>
    <location>local</location>
    <rules_id>120100</rules_id>
    <timeout>300</timeout>
  </active-response>
```

```
<!-- Deshabilitar usuarios Linux -->
  <command>
    <name>disable-account</name>
    <executable>disable-account</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>
```

Deshabilitar cuenta Active Response

Contribución al sistema: Permite reaccionar automáticamente sin intervención humana, aumentando la seguridad y reduciendo tiempos de respuesta.

5.2 Implantación en los Clientes

Actualmente se cuenta con **dos equipos cliente** basados en **Ubuntu Desktop**.

5.2.1 Instalación del agente Wazuh

En cada cliente se realizaron los siguientes pasos:

1. Descargar el instalador para Linux desde el Dashboard.
2. Instalar con:

```
sudo dpkg -i wazuh-agent_4.9.2-1_amd64.deb
```

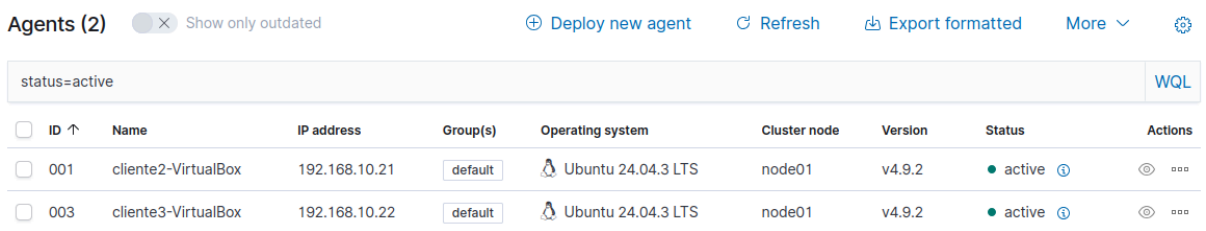
3. Registrar el agente con el manager:

```
sudo /var/ossec/bin/agent-auth -m 192.168.10.10 -A clienteX
```

4. Habilitar y arrancar el servicio:

```
sudo systemctl enable wazuh-agent

sudo systemctl start wazuh-agent
```

Objetivo: Asegurar que todos los equipos de la red envíen logs y eventos al servidor de monitorización.



Agentes Activos Wazuh

5.2.2 Validación de conectividad (tráfico real)

Se verificó que los clientes envían tráfico al manager por los puertos:

- TCP 1514 (logs)
- TCP 1515 (gestión del agente)

Ejemplo de captura en un cliente:

```
sudo tcpdump -ni enp0s3 port 1514 or port 1515
```

Contribución al sistema: Asegura que Wazuh recibe información de seguridad útil.

5.2.3 Políticas de seguridad aplicadas

Los clientes fueron configurados para:

- Registrar eventos del sistema
- Monitorear integridad de archivos
- Detectar accesos fallidos
- Detectar escaladas de privilegios
- Enviar alertas inmediatas al servidor

5.3 PRESUPUESTO (ORIENTATIVO)

Concepto	Cantidad	Precio unitario	Total
Servidor físico (simulado en VirtualBox)	1	0 €	0 €
Clientes virtuales	3	0 €	0 €
Firewall OPNsense (máquina virtual)	1	0 €	0 €
Licencia Ubuntu Server y Ubuntu Desktop	4	0 €	0 €
Software Wazuh (opensource)	1	0 €	0 €
Horas de instalación y configuración	20 horas	25 €/hora	500 €
Documentación del proyecto	10 horas	20€ /hora	200 €
Costes indirectos (electricidad, pruebas, uso equipo)	-	-	50 €

Total estimado del proyecto: 750 €

El proyecto es completamente replicable en un entorno real, donde el coste podría aumentar según el servidor físico, almacenamiento y número de endpoints.

5.4 Análisis DAFO

5.4.1 Debilidades (factores internos negativos)

- **Falta de redundancia**: la infraestructura cuenta con un único servidor Wazuh. Si falla, se detiene toda la monitorización.
- **Dependencia del administrador**: el sistema requiere conocimientos intermedios/avanzados en Linux, redes y seguridad para su mantenimiento.
- **Limitada capacidad inicial**: aunque escalable, en su estado actual solo dispone de dos clientes y un servidor, lo cual reduce la visibilidad total de la red.

- **Configuraciones manuales complejas**: Active Response, integraciones externas (Telegram, Gmail/Postfix), y reglas personalizadas pueden provocar errores si no se documentan bien.
- **Ausencia de mecanismos de alta disponibilidad** (HA) tanto en OPNsense como en Wazuh.
- **Recursos restringidos** debido al entorno virtual (dependen del hardware del host).

5.4.2 Amenazas (factores externos negativos)

- **Ciberataques cada vez más sofisticados**, incluyendo fuerza bruta, malware o ataques internos.
- **Falsos positivos** que pueden llevar a bloqueos automáticos no deseados mediante Active Response.
- **Dependencia de software externo** (Telegram, repositorios Wazuh, Gmail SMTP), que podría sufrir cambios o interrupciones.
- **Riesgos de exposición** si no se gestionan bien las claves, tokens de API o certificaciones SSL.
- **Fallo de red o caída del firewall OPNsense**, lo que dejaría sin Internet y sin monitorización al sistema.
- **Actualizaciones de seguridad de Wazuh o Ubuntu** que pueden romper configuraciones si no se prueban previamente.

5.4.3 Fortalezas (factores internos positivos)

- **Arquitectura modular y escalable**: permite añadir más clientes, servidores, sensores o reglas sin rehacer la infraestructura.
- **Monitorización centralizada**: Wazuh ofrece control completo sobre logs, integridad, vulnerabilidades y alertas.
- **Firewall OPNsense robusto**, con reglas, VLANs, IDS/IPS, NAT y capacidad de segmentación.
- **Active Response** automatiza medidas defensivas ante incidentes (bloqueo IP, deshabilitar usuarios, etc.).
- **Integración con Telegram** que permite alertas en tiempo real y mejora la detección temprana.
- **Bajo coste económico** gracias al uso de herramientas open-source.
- **Entorno completamente virtualizado**, permitiendo pruebas, snapshots y restauraciones rápidas.
- **Buena documentación interna generada durante el proyecto**, útil para mantenimiento o ampliaciones futuras.

5.4.4 Oportunidades (factores externos positivos)

- **Escalabilidad futura**: posibilidad de crecer a decenas de clientes o múltiples servidores Wazuh con balanceadores.
- **Automatización avanzada** mediante scripts, API de Wazuh o integración con SIEM externos.
- **Introducción de nuevas tecnologías** como Suricata/Zeek para IDS/IPS, copias de seguridad automatizadas o HA con CARP.
- **Aplicación real en entornos empresariales**, ya que las tecnologías usadas son estándar en ciberseguridad profesional.
- **Integración con otras soluciones de monitoreo** (Grafana, Prometheus, Zabbix).
- **Posibilidad de migrar a una infraestructura cloud híbrida** usando Azure/AWS y extender el monitoreo.
- **Formación práctica sólida**, que mejora la empleabilidad y habilidades técnicas del administrador.

6. RECURSOS

En este apartado se detallan los recursos necesarios para la puesta en marcha del proyecto, incluyendo hardware, software, sistemas operativos y personal involucrado tanto en la instalación como en el mantenimiento del sistema.

6.1 Herramientas Hardware

Para la implementación del sistema se han utilizado los siguientes componentes físicos y virtuales:

Servidor principal

- Equipo físico anfitrión con capacidad suficiente para ejecutar varias máquinas virtuales.
- Máquina virtual con:
  - 4 GB de RAM asignada.
  - 2 CPUs virtuales.
  - 60 GB de almacenamiento.
  - Ubuntu Server 24.04 LTS.
  - Conectividad de red configurada mediante interfaces *red interna* y *solo-anfitrión*.

Equipos cliente

- Dos máquinas virtuales con características similares:
  - 2 GB de RAM.
  - 1 CPU virtual.

- 20 GB de almacenamiento.
- Ubuntu Desktop / Ubuntu Server.
- Conectividad hacia la red interna gestionada por OPNsense.

Firewall / Router

- Máquina virtual dedicada ejecutando **OPNsense** con:
  - 1–2 GB RAM.
  - 1 CPU.
  - 20 GB de almacenamiento.
  - Dos interfaces de red (WAN / LAN).

6.2 Herramientas Software

A nivel software, el proyecto ha requerido las siguientes herramientas:

Seguridad y monitorización

- **Wazuh 4.9.2** (Manager, Indexer, Dashboard).
- **Wazuh Agent** instalado en los equipos cliente.
- **Filebeat** para el envío de logs desde el servidor al indexer.

Comunicación y alertas

- Bot de **Telegram** integrado en Wazuh mediante script personalizado en Python.
- **Postfix** como posible mecanismo de envío de alertas por correo electrónico (dependiendo de la configuración final).

Sistemas de red

- **OPNsense** como cortafuegos, gestor DHCP y puerta de enlace para los clientes.

Entorno de virtualización

- **VirtualBox** como plataforma para la creación de la infraestructura virtual.

Herramientas de soporte

- **SSH** para administración remota.
- **tcpdump** para análisis básico de tráfico.
- **Netplan** para configurar la red en Ubuntu Server.
- **Python 3** para el desarrollo del script de integración con Telegram.

6.3 Sistemas operativos empleados

Servidor

- Ubuntu Server **24.04 LTS**
  - Paquetes relevantes:
    - `wazuh-manager`
    - `wazuh-indexer`
    - `wazuh-dashboard`
    - `filebeat`
    - `python3`, `pip`, librerías necesarias para Telegram

Clientes

- Ubuntu Desktop / Server **24.04 LTS**
  - Paquete principal:
    - `wazuh-agent`

Firewall

- **OPNsense**, última versión disponible en el momento de la implementación.

6.4 Personal

Personal técnico involucrado en el desarrollo

- **Administrador de sistemas** (rol asumido por el alumno).
  - Instalación del servidor Wazuh.
  - Configuración del firewall OPNsense.
  - Instalación y registro de agentes.
  - Integración de alertas con Telegram.
  - Pruebas, verificación y documentación.

Personal involucrado en mantenimiento

- **Administrador de red y seguridad**:
  - Encargado de la supervisión de alertas.
  - Revisión periódica de logs.
  - Ajuste de reglas de seguridad y Active Response.
    Gestión de nuevas incorporaciones de clientes.

6.5 Personal dedicado a la consecución del proyecto

Dado que se trata de un entorno académico y controlado, el desarrollo, implantación y mantenimiento inicial son asumidos por una única persona (el alumno), replicando el rol de un administrador de sistemas completo dentro de una empresa pequeña.

En un despliegue real, estos roles se dividirán en:

- Técnico de sistemas.
- Técnico de seguridad.
- Técnico de redes.
- Responsable de cumplimiento y documentación.

7. CONCLUSIONES

La implantación de una infraestructura de monitorización y seguridad basada en **OPNsense**, **Wazuh** y una red virtualizada en VirtualBox ha permitido desarrollar un entorno funcional y escalable que reproduce de forma realista las necesidades de una pequeña empresa. Durante el proyecto se han integrado servicios de red, mecanismos de seguridad, sistemas de alertas automáticas y elementos de respuesta activa, demostrando la capacidad de estas tecnologías para proporcionar una protección eficaz ante incidentes.

La realización del proyecto ha permitido aplicar múltiples competencias adquiridas a lo largo del ciclo formativo, como administración de sistemas operativos, diseño de redes, implantación de medidas de seguridad, monitorización centralizada, gestión de logs y automatización de procesos. Al mismo tiempo, se han investigado herramientas y tecnologías no cubiertas en profundidad en el aula, como Wazuh, Filebeat, OPNsense y los sistemas de integración mediante bots de Telegram, lo que ha ampliado considerablemente el conocimiento técnico adquirido.

A lo largo del proyecto se han superado numerosas dificultades técnicas, especialmente relacionadas con la instalación, compatibilidad de versiones, servicios fallidos y configuración de componentes interconectados. Esto ha permitido consolidar procedimientos de diagnóstico de fallos, lectura de logs, depuración de servicios y reconstrucción de entornos dañados. A pesar de la complejidad del conjunto, el resultado final es un sistema operativo y totalmente funcional.

7.1 Grado de consecución de objetivos

El grado de consecución de los objetivos planteados inicialmente puede considerarse **alto**, ya que prácticamente todos los requisitos funcionales del proyecto se han logrado implementar con éxito. Entre los hitos alcanzados destacan:

Infraestructura de red totalmente operativa

- Diseño e implantación de una red interna en VirtualBox.
- Creación de segmentos de red separados para servicios, clientes y administración.
- Configuración de OPNsense como firewall y puerta de enlace.
- Asignación de direccionamiento estático y enrutamiento funcional.
- Acceso remoto seguro mediante SSH.

Instalación y puesta en marcha de un servidor Wazuh completo

- Manager, Indexer y Dashboard instalados y sincronizados.
- Configuración SSL y comunicación segura.
- Integración de Filebeat para envío de logs al dashboard.
- Gestión y registro de múltiples agentes Linux.

Monitorización centralizada

- Recepción de logs, alertas e información de integridad desde los clientes.
- Paneles personalizados para visualizar agentes activos, eventos críticos, accesos fallidos, cambios en archivos y ejecuciones sospechosas.

Integración de mensajería con Telegram

- Configuración de bot propio.
- Creación de script Python para envío de alertas.
- Implementación de la integración en el servidor Wazuh.

Active Response operativo

- Creación de comandos AR.
- Automatización de respuestas ante accesos indebidos, intentos de fuerza bruta o eventos críticos.
- Configuración tanto en servidor como en clientes.

Documentación completa del proceso

- Explicación detallada de la instalación, el diseño, la topología y el análisis técnico.
- Reflexión sobre los problemas encontrados y las soluciones aplicadas.

Objetivos parcialmente conseguidos

- El agente del cliente 1 presentó incidencias y aún requiere revisión final.

Aun así, la funcionalidad principal del proyecto se cumple ampliamente.

7.2 Problemas encontrados

Durante el desarrollo se enfrentaron diversos problemas técnicos, algunos relacionados con la compleja arquitectura de Wazuh y otros con la configuración del entorno de red. Entre los más significativos destacan:

Conflictos de versiones en Wazuh

Fue uno de los puntos más críticos. La coexistencia de versiones 4.9.x y 4.14.x provocó:

- incompatibilidad entre agentes y manager;
- errores de autenticación;
- corrupción parcial de servicios;
- imposibilidad temporal de iniciar Wazuh Manager y Wazuh Indexer.

Esto obligó a reinstalar completamente el servidor en varias ocasiones y depurar manualmente los paquetes instalados.

Eliminación incompleta de paquetes

Los scripts pre-removal y post-removal fallaron, dejando restos en:

- **`/var/ossec`**
- **`/etc/wazuh*`**
- **`/var/lib/dpkg/info/`**

Hubo que realizar borrado forzado y aplicar comandos dpkg avanzados.

Problemas con certificados SSL en Wazuh Indexer

El servicio fallaba al iniciar debido a certificados faltantes o no generados correctamente.

Cliente 1 sin comunicación

Aunque el agente estaba instalado:

- aparecía como *never connected*,
- enviaba clave duplicada,
- usaba un nombre de agente inconsistente.

Requiere revisión futura.

Configuración de Telegram en Wazuh

El formato del bloque `<integration>` es extremadamente sensible. Se produjeron errores como:

- Elementos no válidos (`group_id`, `active`) fuera de los bloques correctos.
- El manager no iniciaba por configuraciones mal ubicadas.

También costó encontrar el **chat_id** del grupo y añadir correctamente el bot.

Dificultades con OPNsense

En una ocasión perdió la dirección WAN por cambios en adaptadores de VirtualBox, afectando a:

- conectividad del servidor,
- actualización de paquetes,
- comunicación entre máquinas.

7.3 Mejoras

Aunque el sistema ya es plenamente funcional, existen varias líneas de mejora que permitirían ampliar, optimizar o profesionalizar aún más la infraestructura:

Mejoras técnicas

- **Revisión y corrección definitiva del cliente 1**, asegurando la comunicación con el manager.
- **Implementación de reglas personalizadas** en Wazuh basadas en el comportamiento específico de la empresa.
- **Incorporación de más sistemas operativos** como clientes (Windows Server, Windows Desktop, Ubuntu Desktop, etc.).
- **Automatización de despliegue** mediante Ansible o scripts Bash.

- **Uso de contenedores Docker** para la infraestructura Wazuh, simplificando reinstalaciones.
- **Integración con sistemas SIEM externos** para análisis avanzado.

Mejoras en seguridad

- Añadir IDS/IPS adicional (Suricata) en OPNsense.
- Implementar VPN para acceso remoto seguro de administradores.
- Crear listas negras automáticas según alertas de Wazuh.
- Configurar copias de seguridad automáticas del servidor.

Mejoras en la presentación y usabilidad

- Crear paneles en Wazuh Dashboard adaptados a cada rol (administrador, auditor, técnico).
- Documentar flujos de incidentes para respuesta rápida.
- Generar informes automáticos semanales o mensuales.

Escalabilidad futura

Debido al diseño modular, el entorno puede ampliarse fácilmente:

- añadir nuevos clientes;
- incorporar servicios adicionales (servidor web, base de datos, NAS);
- evolucionar hacia un entorno empresarial completo.

8. Índice de Imágenes:

9. Contenidos repositorio GitHub

10. ANEXOS Y DOCUMENTOS COMPLEMENTARIOS

Active response:
https://documentation.wazuh.com/4.9/user-manual/capabilities/active-response/ar-use-cases/blocking-ssh-brute-force.html

https://documentation.wazuh.com/4.9/user-manual/capabilities/active-response/ar-use-cases/restarting-wazuh-agent.html

https://documentation.wazuh.com/4.9/user-manual/capabilities/active-response/ar-use-cases/disabling-user-account.html

https://wazuh.com/blog/how-to-detect-active-directory-attacks-with-wazuh-part-1-of-2/

SMTP:

https://dos4s.netrunners.sh/blog/sys/wazuh-email-alerts#introduction

Telegram:

https://www.cgii.gob.bo/bookstack/books/manuales-y-guias-de-seguridad/page/envio-de-alertas-mediante-bot-de-telegram

Instalación Wazuh:
https://www.youtube.com/watch?v=Ag6llMxBFZ4&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