Despliegue y funcionamiento

de Wazuh

Alejandro Montes Delgado

IES Gonzalo Nazareno

Índice

1. Objetivos que se quieren conseguir.

El objetivo para la finalización de este proyecto es tener desplegada la herramienta wazuh en un contenedor Docker, con métricas recogidas de un agente wazuh para su posterior explicación del funcionamiento de dicha herramienta y análisis de las métricas recogidas.

2. Escenario necesario para la realización del proyecto.

El escenario consiste en una instancia en Proxmox o OCI (Oracle Cloud Infraestructure) con Docker donde realizaremos el despliegue de dicha herramienta.

3. Fundamentos teóricos y conceptos.

3.1. Definición y funciones.

Wazuh es una plataforma Open-Source que unifica las capacidades de XDR (detección y respuesta extendidas) y SIEM (gestión de información y eventos de seguridad), y está integrado con Elastic Stack, utilizada para la prevención, detección y respuesta a las amenazas, es capaz de proteger cargas de trabajos en entornos locales, virtualizados, en contenedores y en la nube.

La solución Wazuh consta de agentes de seguridad desplegado en los sistemas supervisados, y un servidor que recoge y analiza los datos recopilados por los agentes.

Los servicios que brinda wazuh son los siguientes:

- Análisis de seguridad: Wazuh es utilizado para recolectar, agregar, indexar y analizar información de seguridad, ayudando a las organizaciones a detectar intrusos, amenazas y anomalías de comportamiento.

- Detección de intrusos: Los agentes de Wazuh monitorizan los sistemas buscando malware, rootkits y anomalías sospechosas. Puede detectar archivos ocultos, procesos ocultos, listener de red no registrados o inconsistencias en respuestas de llamadas al sistema. Además, el servidor usa un enfoque basado en firmas para la detección usando un motor de expresiones regulares para analizar la información de los logs recolectados.

- Análisis de logs: Los agentes de Wazuh leen el SO y los logs de las aplicaciones, y los envían de manera segura al administrador central para ser analizados y almacenados. Las reglas de Wazuh ayudan a mantener la consistencia frente a errores de aplicación o sistemas, malas configuraciones, intentos de actividades maliciosas, violación de políticas entre otros problemas operacionales y referidos a la seguridad.

- Monitorización de la integridad de los Archivos: Wazuh monitorea el sistema de archivos, identificando cambios en el contenido, permisos, propiedad, y los atributos que necesitas mantener bajo control.

- Detección de vulnerabilidades: Los agentes de Wazuh envían información referida al servidor, donde es correlacionada con la base de datos de actualizaciones continuas de CVE (Common Vulnerabilities and Exposure), para identificar las vulnerabilidades de software conocidas

- Configuración de Evaluación: Wazuh monitorea los ajustes de configuración del sistema y las aplicaciones para asegurar que se alinean con las políticas de seguridad. Los agentes actúan escaneando periódicamente para detectar aplicaciones que son conocidas como vulnerables, no parcheadas o configuradas inseguramente. Adicionalmente, la configuración puede ser personalizada, adaptándose para ser alineadas con las propiedades de tu organización. Las alertas incluyen recomendaciones para mejorar la configuración, referencias y mapeo con cumplimiento normativo.

- Respuesta ante incidentes: Wazuh provee respuestas activas «*out-of-the-box*» para accionar varias contramedidas para la dirección de amenazas activas, como bloquear acceso al sistema para las fuentes de amenazas cuando ciertos criterios son cumplidos. Adicionalmente, Wazuh puede ser usado para correr

comandos remotamente o solicitudes del sistema, identificando indicadores de compromisos (IOCs) y ayudando a realizar otras tareas forenses o tareas automatizadas en respuestas a incidentes.

- Cumplimiento Normativo: Wazuh provee algunos de los controles de seguridad necesarios para cumplir con los estándares y regulaciones de la industria. Estas características combinadas con su escalabilidad y soporte multiplataforma ayuda a las organizaciones al cumplimiento técnico de los requerimientos.

- Seguridad en la nube: Wazuh ayuda a monitorear infraestructuras en la nube a un nivel API, usando integración de módulos que son capaces de enviar datos de manera segura desde proveedores de servicios en la nube bien conocidos, como Amazon AWS, Azure o Google Cloud.

- Seguridad en contenedores: Wazuh provee seguridad en los host y contenedores Docker, monitorizando su comportamiento y detectando amenazas y anomalías. El agente Wazuh tiene integración nativa con el motor de Docker permitiendo a los usuarios monitorizar las imágenes, volúmenes,modo configuraciones de red, y contenedores corriendo.

3.2. Componentes.

Wazuh consta de los siguientes componentes:

- OSSEC HIDS: Es un sistema de detección de intrusos(HIDS) que es usado para la detección, visibilidad y monitorización del cumplimiento de eventos de seguridad. Está basado en un agente multiplataforma que envía datos del sistema a un gestor central, donde es analizado y procesado, dando como resultado alertas de seguridad. También es un servidor centralizado de logs.

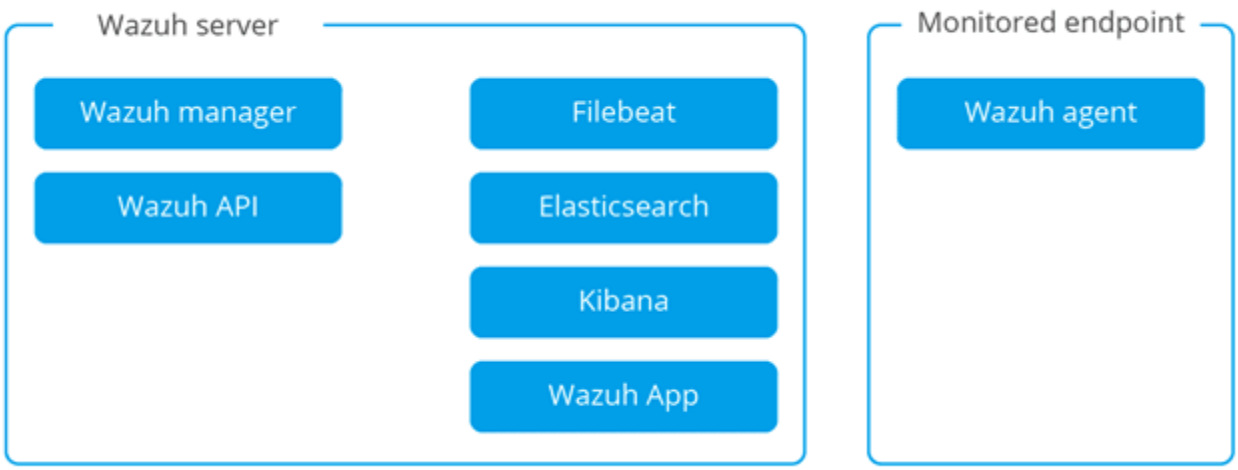
- OpenSCAP: Es un intérprete que se usa para chequear las configuraciones del sistema y detectar aplicaciones vulnerables. Es una herramienta diseñada para el cumplimiento de la seguridad y el bastionado de los sistemas en un entorno empresarial.

- Elastic Stack: Es un conjunto de software(Elasticsearch, Kibana y Filebeat) que es usado para recolectar, comparar, almacenar, indexar, buscar y mostrar datos de logs. Proporciona una interfaz web que muestra los datos a través de un dashboard(panel de control).

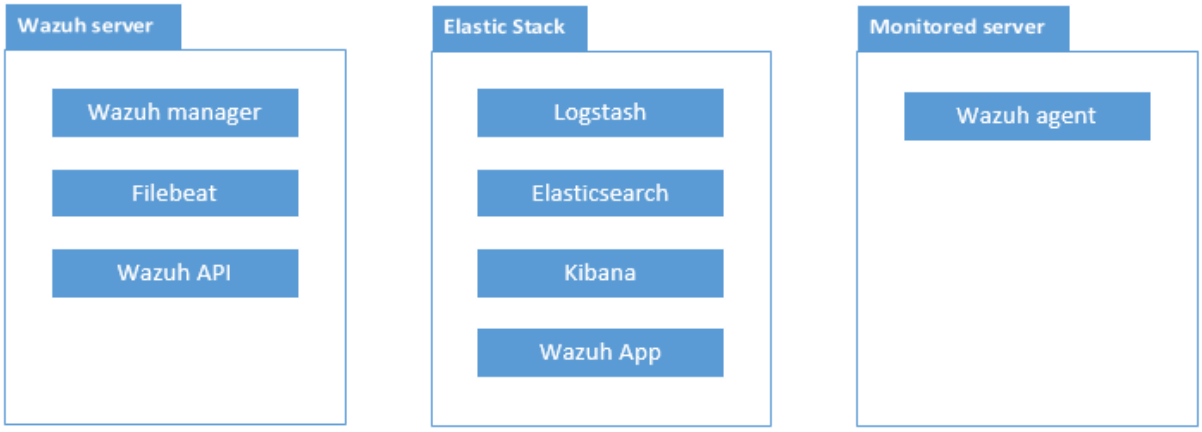
3.3. Arquitectura de Wazuh.

Wazuh tiene dos componentes principales que instalar: Wazuh Manager y Elastic Stack. Hay dos tipos de arquitectura para la instalación de Wazuh:

- Arquitectura centralizada: Wazuh y Elastic Stack se ejecutan en el mismo servidor. La siguiente imagen muestra como es este tipo de arquitectura:



- Arquitectura distribuida: Wazuh y Elastic Stack se ejecutan en distintos servidores y en uno o varios formando así un clúster. En la siguiente imagen se muestra su arquitectura:



4. Despliegue y configuración de Wazuh.

El escenario que se ha realizado para la demostración del proyecto es el siguiente:

- Servidor Wazuh con Docker.
- Agente de wazuh Debian.

En el servidor wazuh mostraré toda la configuración realizada y en el agente de wazuh mostraré como se instala e instalaremos algún servicio para que el servidor wazuh tenga logs que recoger.

4.1. Despliegue de Wazuh con Docker.

Lo primero que instalaremos en el servidor será docker. A continuación muestro los comandos usados para la instalación de dicha herramienta.

Lo primero que haremos será actualizara nuestro sistema e instalar los siguientes paquetes:

```
$ sudo apt-get update
$ sudo apt-get install ca-certificates curl gnupg
```

Añadimos los repositorios oficiales de Docker:

```
$ sudo install -m 0755 -d /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
$ sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

Usamos el siguiente comando para añadir el repositorio:

```
$ echo \
"deb [arch="$(dpkg --print-architecture)"
signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian \
"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Ahora instalamos Docker Engine, para ello actualizaremos nuevamente los repositorios una vez hayamos añadido el de Docker:

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
docker-buildx-plugin docker-compose-plugin
```

Una vez hayamos instalado Docker, instalaremos git y clonaremos un repositorio con el que realizaremos el despliegue de wazuh:

```
$ sudo apt install git
$ git clone https://github.com/wazuh/wazuh-docker.git -b
v4.4.1
```

Dentro del directorio clonado, tendremos los siguientes archivos:

```
usuario@wazuh:~/wazuh-docker$ ls -l
total 76
drwxr-xr-x 5 usuario usuario  4096 may 12 17:57 build-docker-images
-rw-r--r-- 1 usuario usuario 14554 may 12 17:57 CHANGELOG.md
drwxr-xr-x 3 usuario usuario  4096 may 12 17:57 indexer-certs-creator
-rw-r--r-- 1 usuario usuario 24736 may 12 17:57 LICENSE
drwxr-xr-x 3 usuario usuario  4096 may 12 17:57 multi-node
-rw-r--r-- 1 usuario usuario 11578 may 12 17:57 README.md
drwxr-xr-x 3 usuario usuario  4096 may 13 19:22 single-node
-rw-r--r-- 1 usuario usuario    46 may 12 17:57 VERSION
```

Las carpetas que nos interesan son single-node o multi-node, la cual elegiremos si lo queremos instalar en un solo nodo o como un clúster. En mi caso al estar desplegándolo en una máquina en Proxmox la cual no tiene demasiada potencia, he

elegido la opción de single-node. Dentro de la carpeta single-node ejecutamos lo siguiente:

```
$ docker-compose -f generate-indexer-certs.yml run --rm generator
```

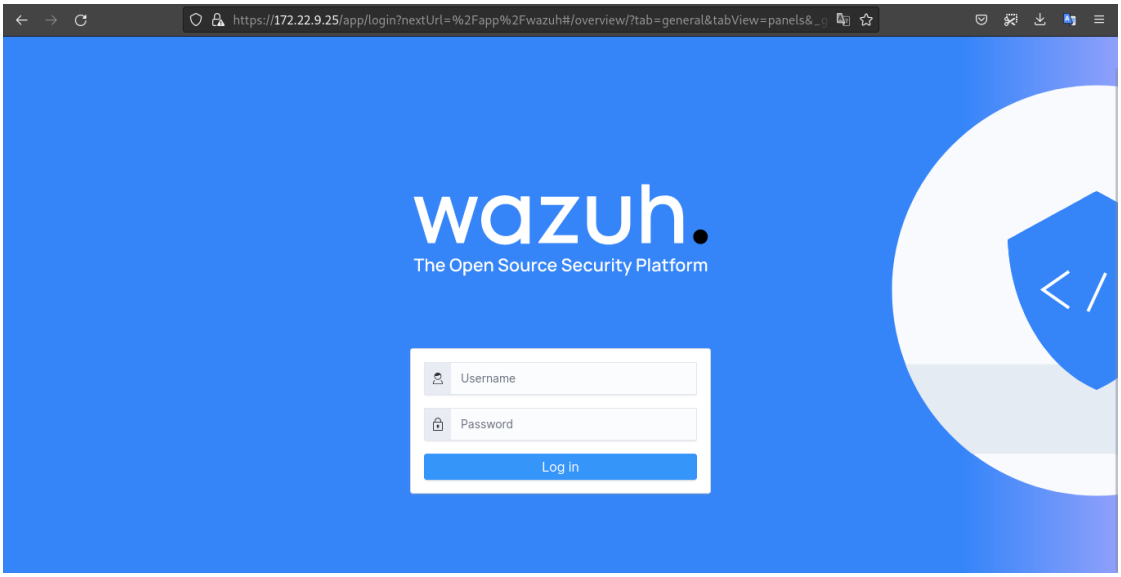
Esto nos generará los certificados correspondiente para los contenedores. Una vez el comando haya finalizado levantaremos el escenario con el siguiente comando:

```
$ docker up -d
```

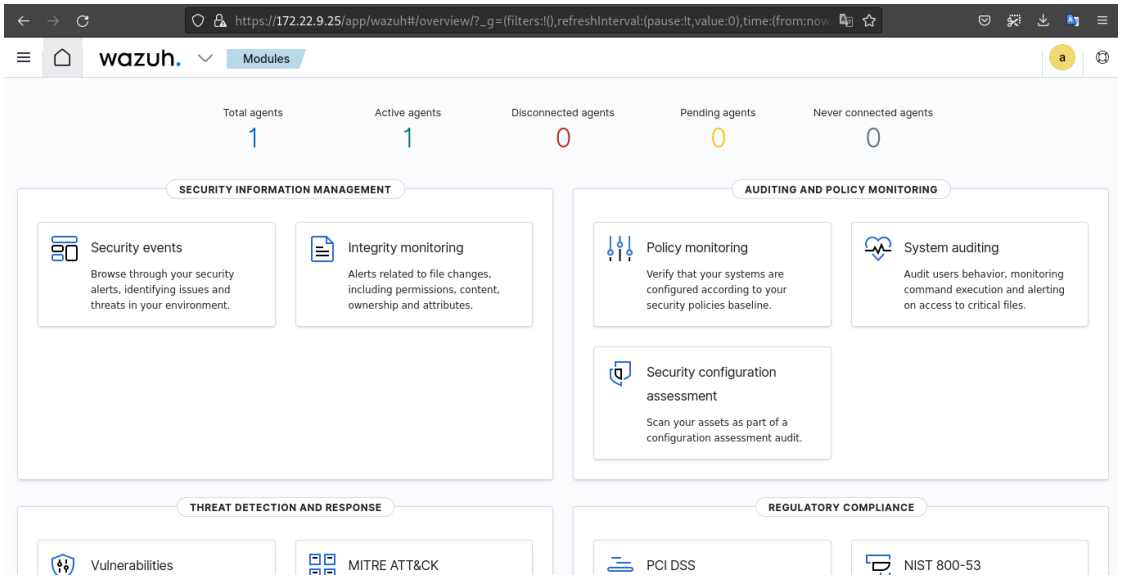
Esto conllevará varios minutos ya que se tienen que bajar muchas imágenes. Una vez finalizado como podremos ver tendremos desplegado los contenedores:

```
usuario@wazuh:~$ docker ps --format '{{ .ID }}\t{{.Image}}\t{{ .Names }}'
905bdffa99fd    wazuh/wazuh-dashboard:4.4.1     single-node-wazuh.dashboard-1
95efcc35f6c0    wazuh/wazuh-manager:4.4.1       single-node-wazuh.manager-1
ea86bccffe9d    wazuh/wazuh-indexer:4.4.1       single-node-wazuh.indexer-1
```

Accederemos al dashboard para comprobar que se ha desplegado correctamente:



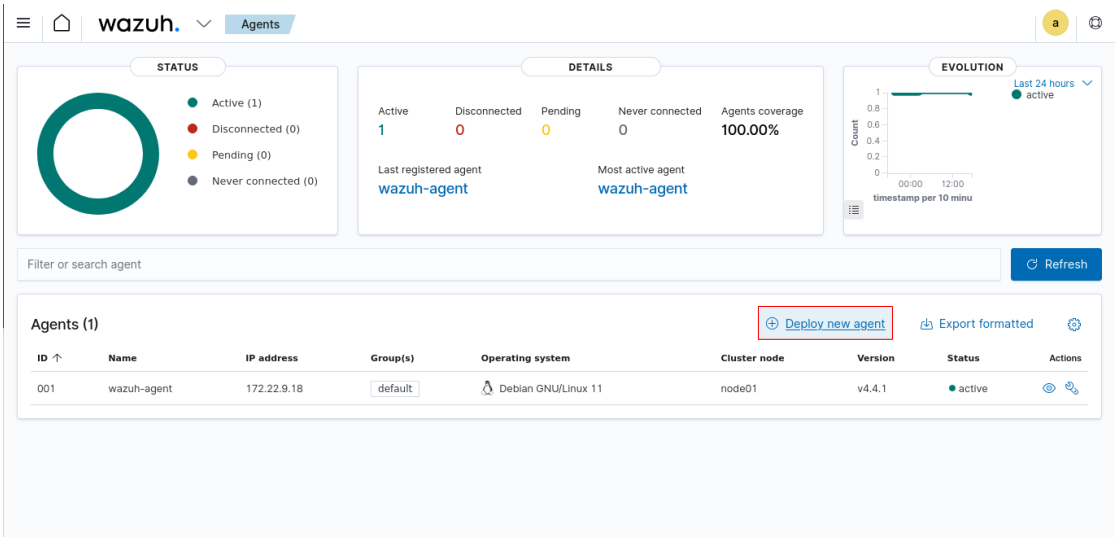
Por defecto el usuario y contraseña son: admin y SecretPassword. Muestro el dashboard una vez iniciada la sesión:



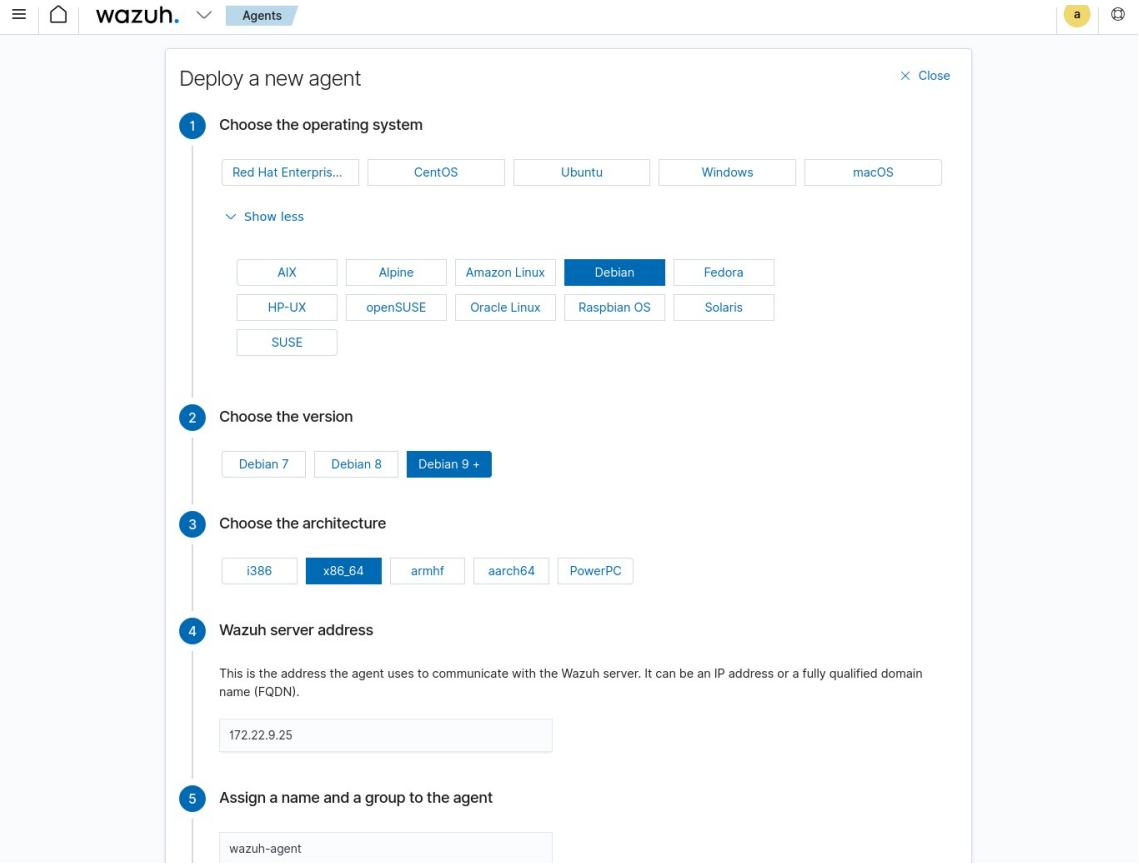
Como podemos ver, hay un agente wazuh conectado, el cual procederemos a mostrar paso a paso como se instala a continuación.

4.2. Instalación agente de wazuh.

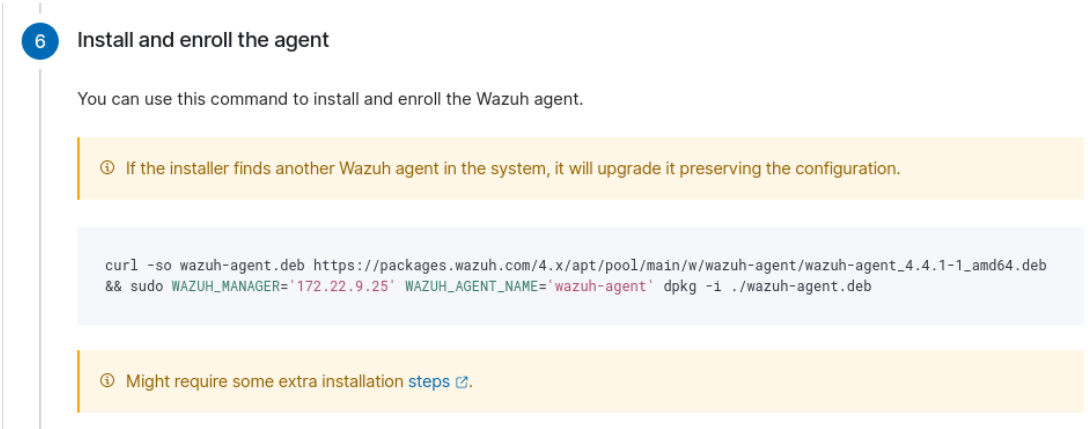
Para instalar una agente de wazuh, lo primero que haremos ser ir al apartado de *Agents* y seleccionar *Deploy new Agent*:



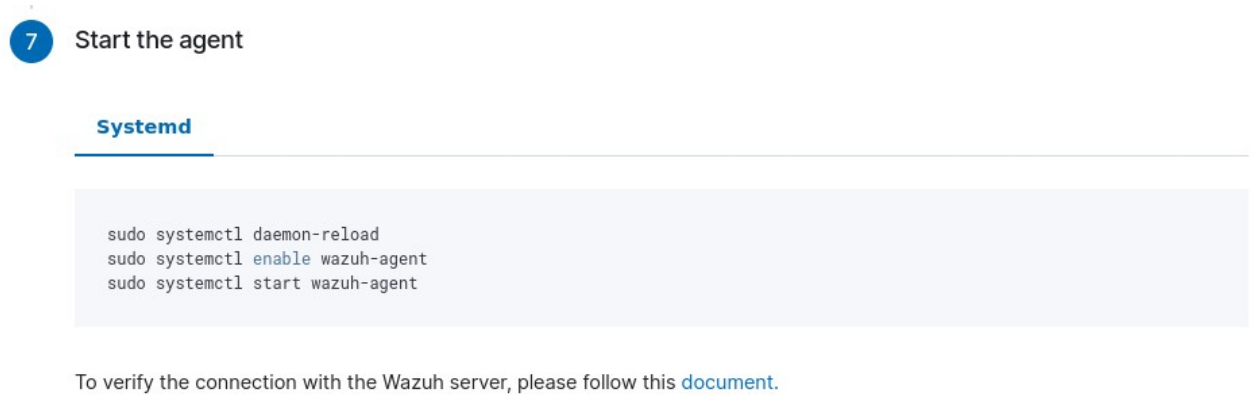
Nos aparecerá la siguiente ventana la cual tendremos que rellenar con la información que nos pide, el sistema operativo del agente, la dirección del servidor wazuh...



En los dos últimos puntos nos muestra los comandos necesarios para realizar la instalación en el cliente:



Y usamos los siguientes comando para iniciar el agente:



4.3. Configuraciones extras.

4.3.1. Activar detección de vulnerabilidades.

La primera configuración que realizaremos será activar la detección de vulnerabilidades en nuestro wazuh manager. Para ello primero tendremos que acceder al contenedor usando el siguiente comando:

```
$ docker exec -ti 95efcc35f6c0 /bin/bash
```

Una vez hayamos entrado en la terminal del contenedor tendremos que editar el siguiente fichero:

```
root@wazuh:/# nano /var/ossec/etc/ossec.conf
```

Si no tenemos nano instalado procederemos a instalarlo. En el fichero tendremos que editar lo siguiente:

```
  <vulnerability-detector>
    <enabled>yes</enabled>
    <interval>5m</interval>
    <min_full_scan_interval>6h</min_full_scan_interval>
    <run_on_start>yes</run_on_start>
```

```
    <!-- Debian OS vulnerabilities -->
    <provider name="debian">
      <enabled>yes</enabled>
      <os>buster</os>
      <os>bullseye</os>
      <update_interval>1h</update_interval>
    </provider>
```

En la etiqueta enabled, tenemos que cambiar de “no” a “yes”. Una vez hecho esto, reiniciaremos wazuh para que se apliquen los cambios:

```
root@wazuh:/# service wazuh-manager restart
```

4.3.2. Configurar servidor SMTP.

Una vez hecho esto, procederemos a configurar el servidor SMTP para enviar el reporte diario a nuestra cuenta de correos. Lo primero que haremos será instalar los siguientes paquetes:

```
root@wazuh:/# apt-get update && apt-get install postfix
mailutils libsasl2-2 ca-certificates libsasl2-modules
```

Si durante la instalación nos da a elegir entre cuatro opciones, elegiremos *Internet Sites*. Ahora editaremos el fichero de configuración de postfix y añadiremos lo siguiente:

```
relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtp_use_tls = yes
smtpd_relay_restrictions = permit_mynetworks,
permit_sasl_authenticated, defer_unauth_destination
```

Ahora para que una cuenta de gmail envíe correos, tendremos que crear una contraseña de aplicación, en mi caso he usado gmail. Adjunto el enlace con los pasos

a seguir para crear la contraseña de aplicación. Una vez tengamos la contraseña de 16 caracteres, ejecutaremos los siguientes comandos:

```
echo [smtp.gmail.com]:587 usuario@gmail.com:contraseña > /etc/postfix/sasl_passwd
postmap /etc/postfix/sasl_passwd
chmod 400 /etc/postfix/sasl_passwd
```

También securizaremos la contraseña en el archivo de base de datos generado por el comando postmap.

```
chown root:root /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db
chmod 0600 /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db
```

Reiniciamos postfix para que se apliquen los cambios:

```
service postfix restart
```

4.3.3. Envío del reporte diario y alertas de riesgo por correo.

Ahora procederemos a configurar que envíe un reporte diario a nuestro correo y alertas de nivel 12 o mayor que son de riesgo crítico. Para ello editaremos el siguiente fichero dentro del contenedor de wazuh manager:

```
root@wazuh:/# nano /var/ossec/etc/ossec.conf
```

Buscaremos la etiqueta email_notification y la cambiaremos a “yes”, indicaremos que el contenedor docker actuará como servidor SMTP ya que previamente lo hemos configurado, indicando localhost. Indicamos la cuenta de correos de la cual enviarán los mails, en mi caso he usado mi cuenta de correos personal y la cuenta de correos que recibirá los emails:

```
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>yes</email_notification>
    <smtp_server>localhost</smtp_server>
    <email_from>aaleemd11@gmail.com</email_from>
    <email_to>amontes.alertswazuh@gmail.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
<agents_disconnection_alert_time>0</agents_disconnection_alert
_time>
  </global>
```

Al final del archivo añadiremos lo siguiente para que envíe el correo diario de reporte:

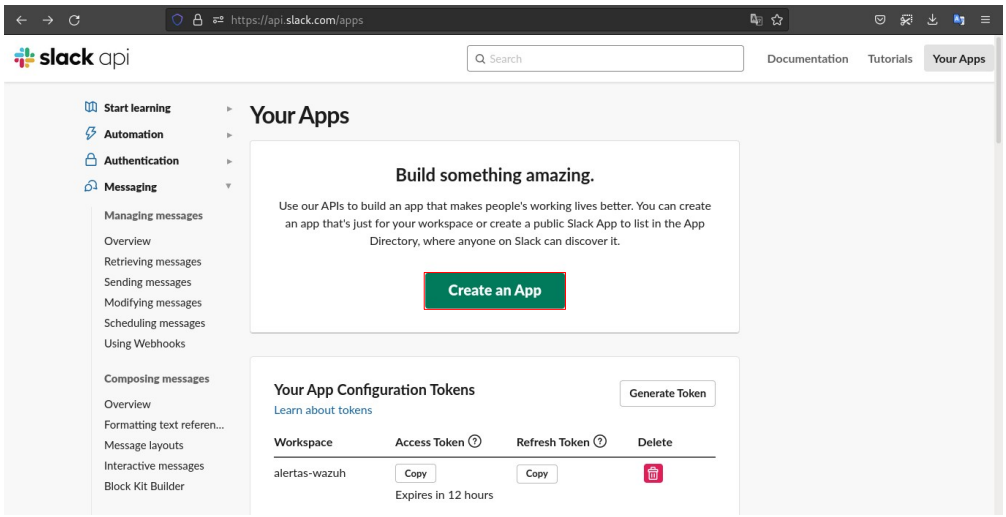
```
<ossec_config>
  <reports>
      <category>syscheck</category>
      <title>Reporte diario de Wazuh</title>
      <email_to>amontes.alertswazuh@gmail.com</email_to>
  </reports>
</ossec_config>
```

Volvemos a reiniciar el servicio para que se apliquen los cambios:

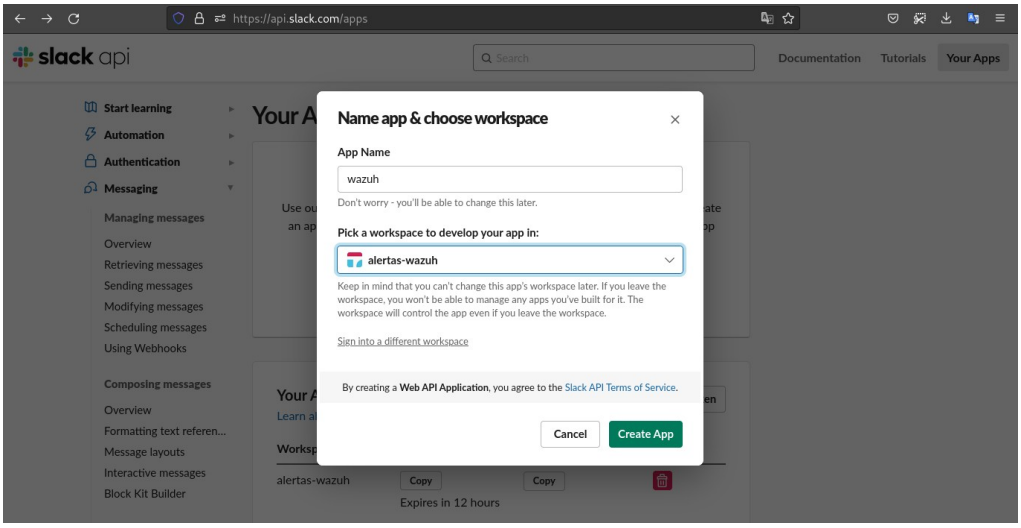
```
root@wazuh:/# service wazuh-manager restart
```

4.3.4. Envío de alertas mediante Slack.

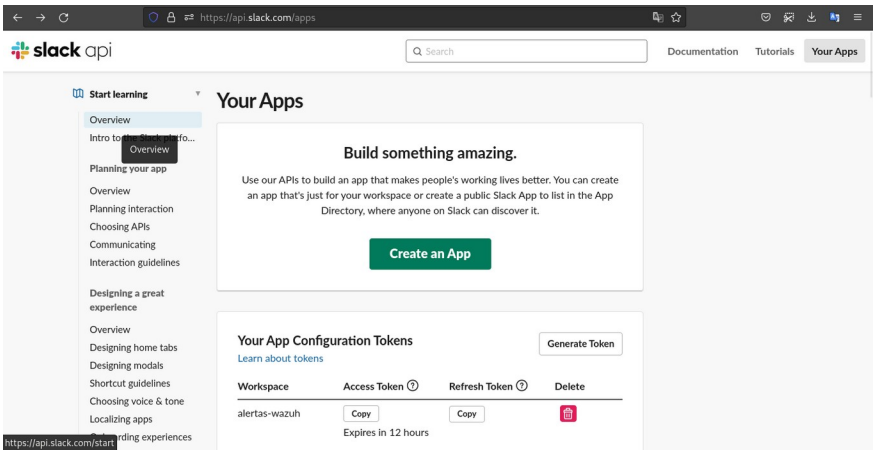
Ahora configuraremos para que nos lleguen alertas a un canal de slack. Para ello, hemos creado en slack un espacio de trabajo llamado *alertas-wazuh*. A partir de este enlace, crearemos una app:



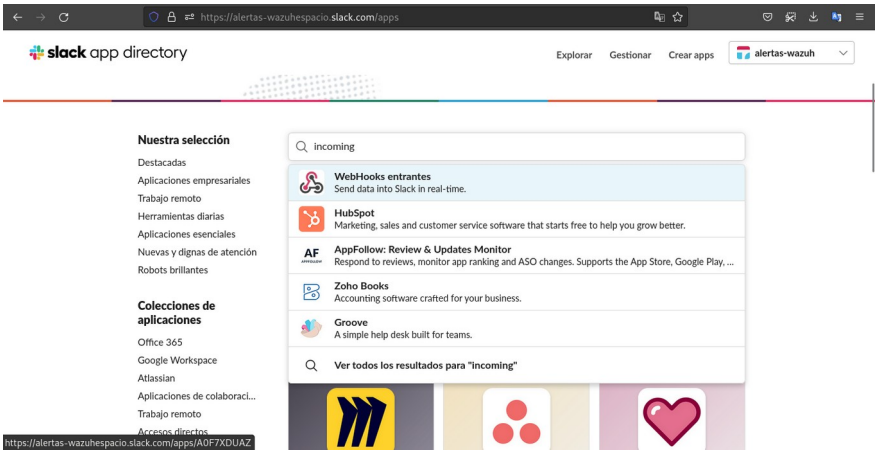
Elegiremos un nombre y el espacio de trabajo al que nos enviaran los mensajes y crearemos la app:



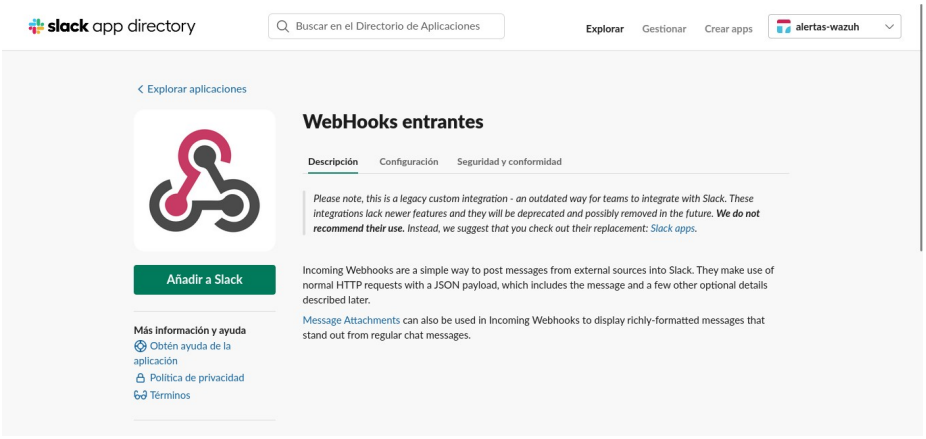
Una vez creada la app, nos dirigiremos a *Start learning → Overview.*



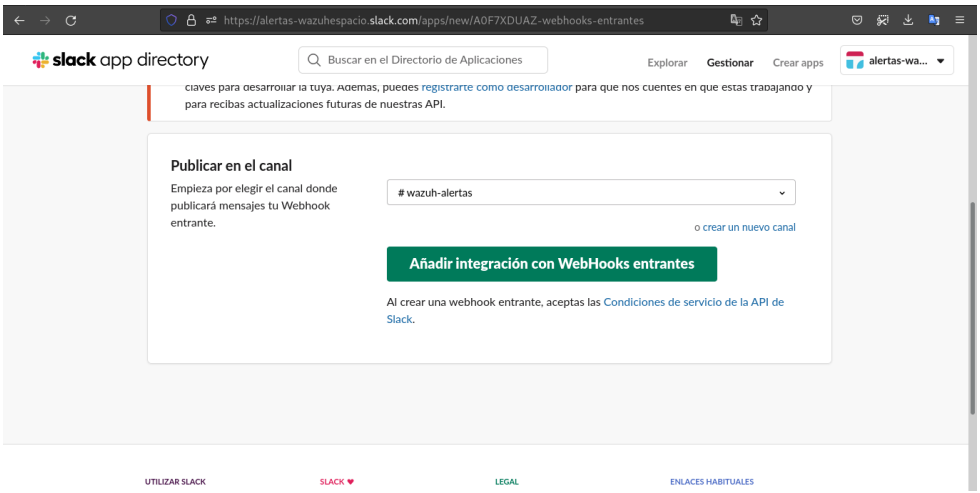
Buscamos *Webhooks entrantes:*



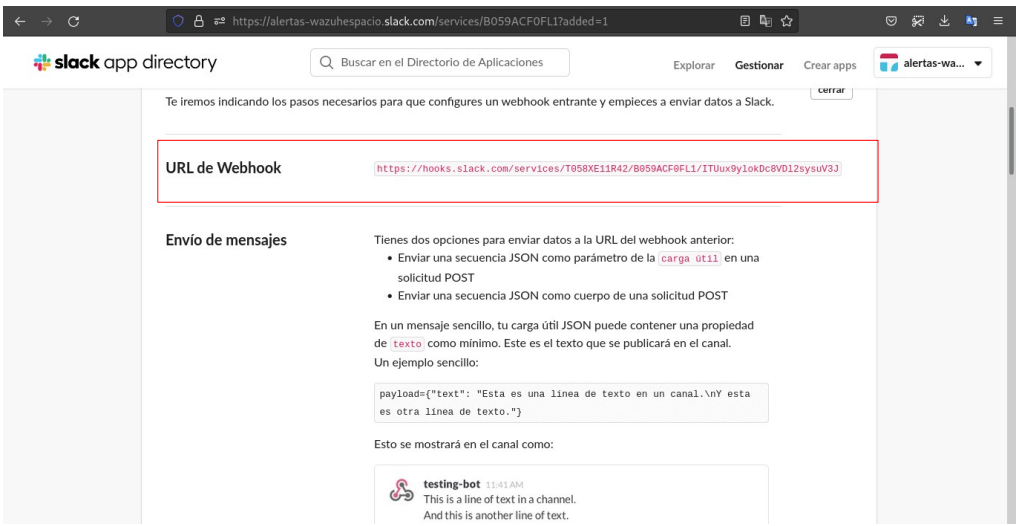
Y lo añadimos a slack:



Elegimos el canal donde nos enviaran los mensajes y pulsamos el botón para añadirlo.



Al añadir la integración nos proporciona una URL webhook:



Con esta URL, editaremos el fichero de configuración que hemos editado anteriormente y añadiremos lo siguiente:

```
root@wazuh:/# nano /var/ossec/etc/ossec.conf
```

```
  <integration>
    <name>slack</name>

<hook_url>https://hooks.slack.com/services/T058XE11R42/…/P45YO
O06hO8cHCg3sRhmGGYo</hook_url>
    <alert_format>json</alert_format>
    <level>3</level>
  </integration>
```

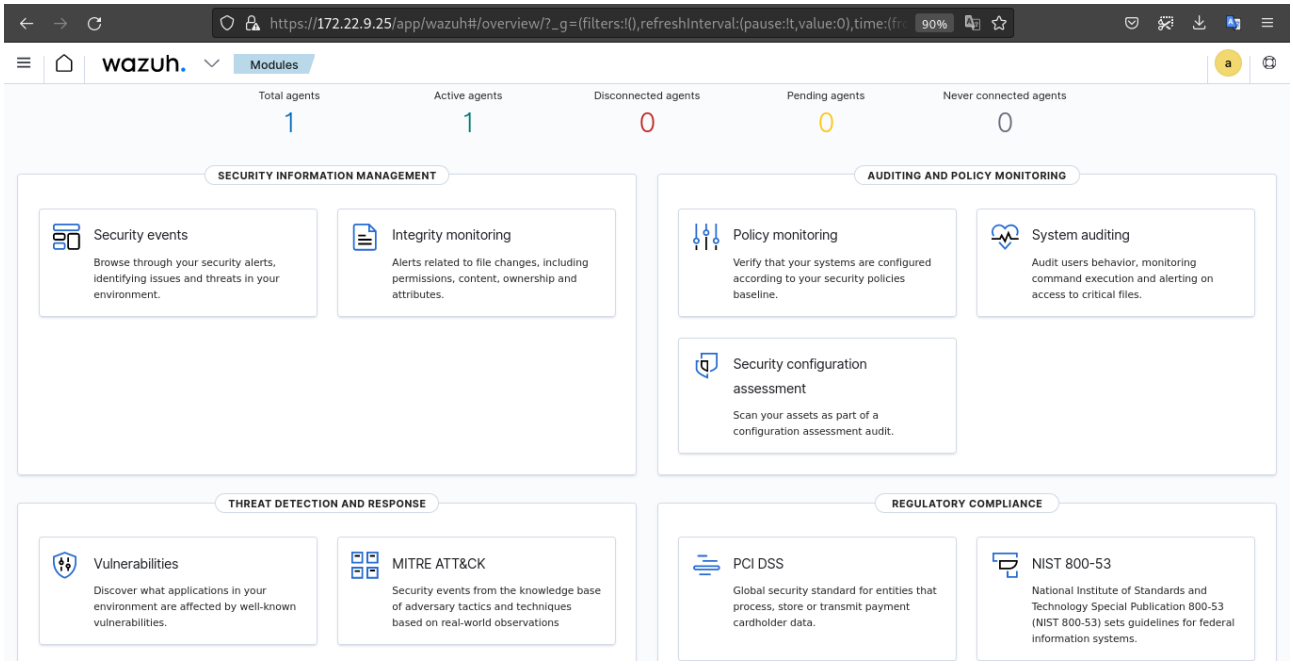
Solo nos queda reiniciar el servicio para que se apliquen los cambios:

```
root@wazuh:/# service wazuh-manager restart
```

5. Explicación y prueba de funcionamiento.

5.1. Explicación del funcionamiento de Wazuh.

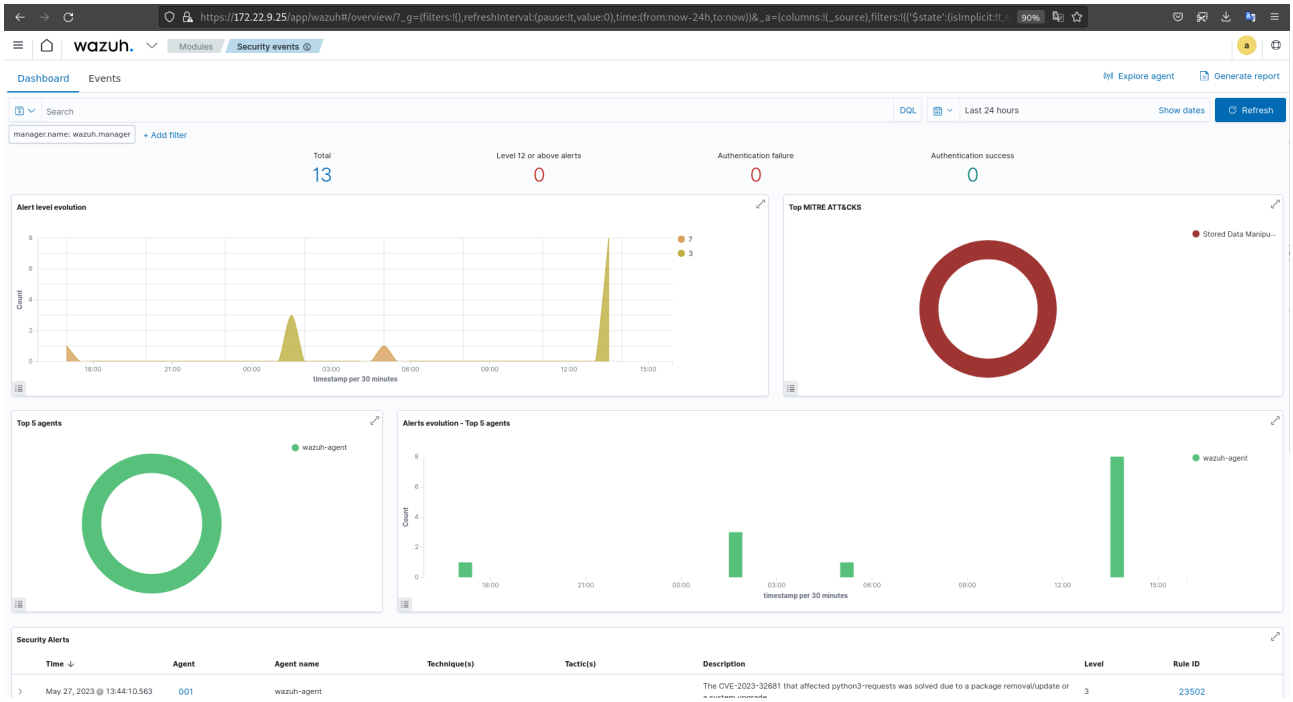
Lo primero que veremos será la pantalla de inicio de Wazuh.



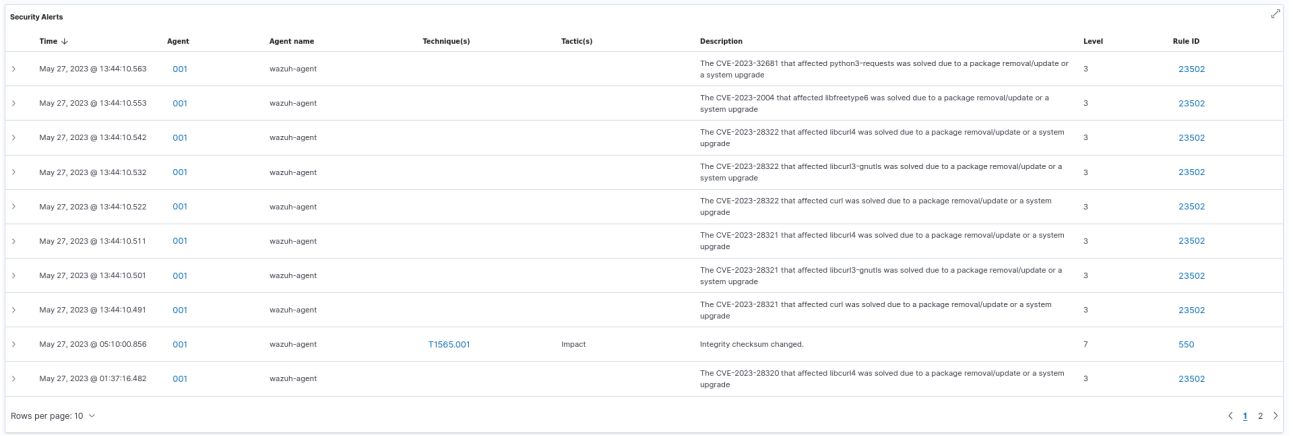
Como podemos ver, la pantalla de inicio está dividida en 4 apartados, gestión de la información de seguridad, auditoría y control de políticas, detección de amenazas y respuesta, cumplimiento normativo y arriba de estos apartados nos indica los agentes conectados y el estado de los mismos.

En el apartado de gestión de la información de seguridad, nos encontramos 2 herramientas, eventos de seguridad que examina las alertas de seguridad, identificando los problemas y las amenazas de nuestro entorno y monitorización de integridad que nos alerta de cambios en los archivos incluidos permisos, contenido, propiedad y atributos.

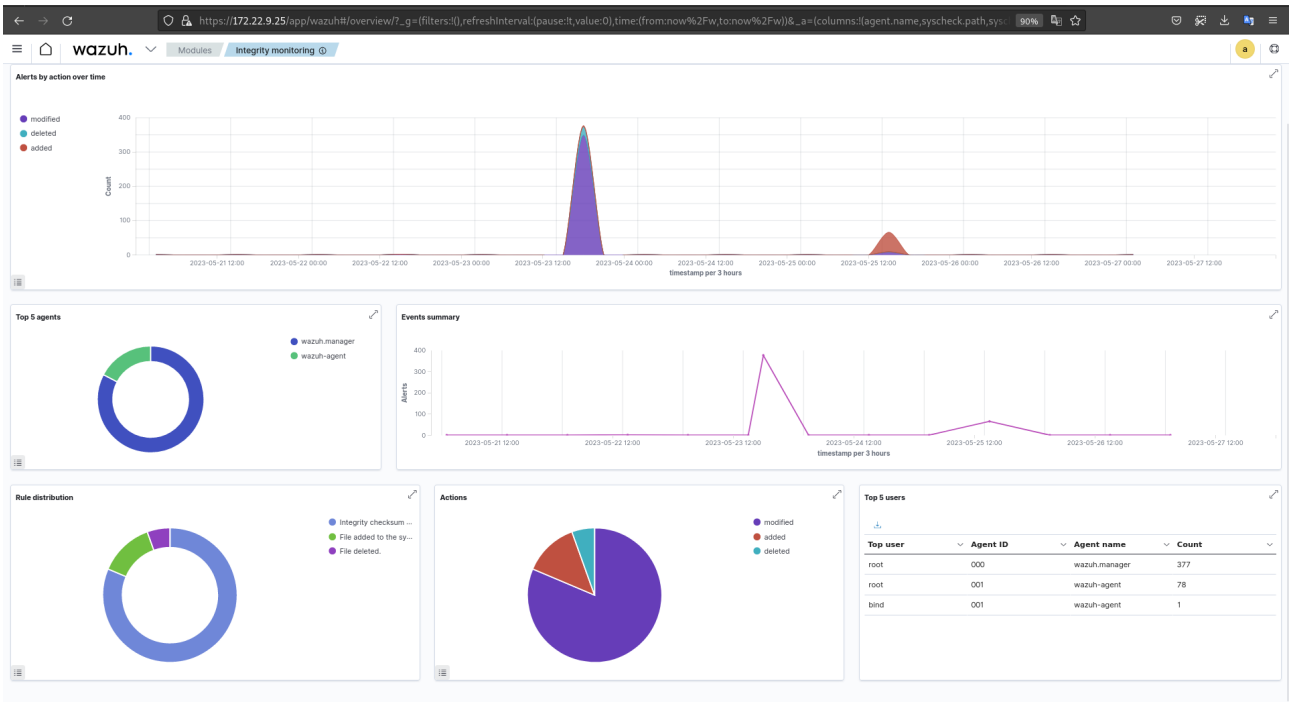
Si entramos en el apartado de eventos de seguridad, nos muestra mediante gráficos las alertas, el nivel de alertas y los agentes que tienen dicha alerta, como en mi caso solo he indicado un agente, solo nos aparecerá alertas de un agente.



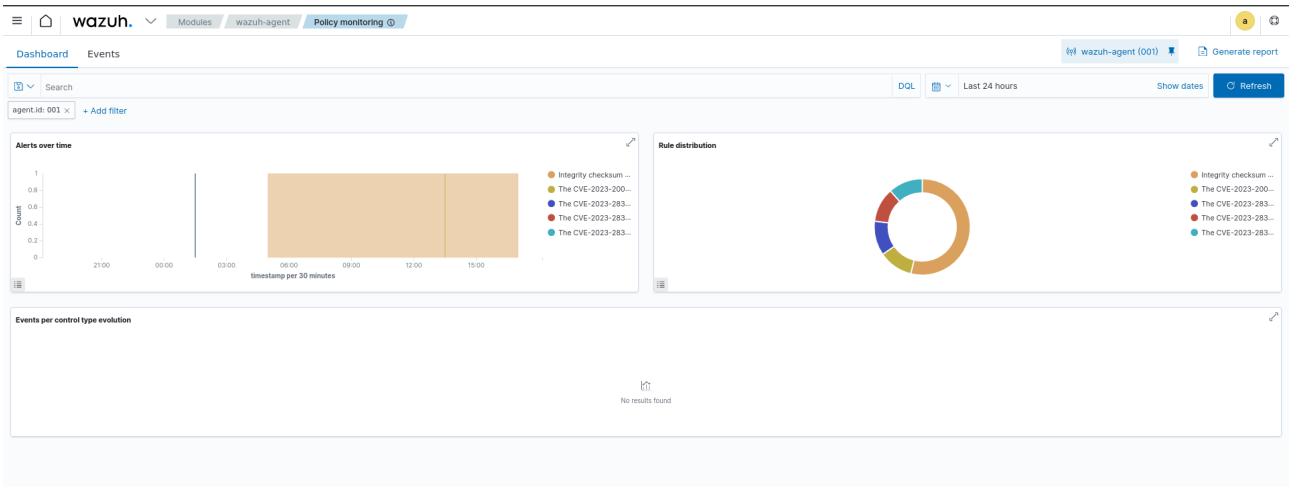
Si scrolleamos, al final nos aparecen todas las alertas de seguridad, indicándonos la hora en que se ha producido, el nombre del agente, una descripción de la alerta y el nivel de la alerta.



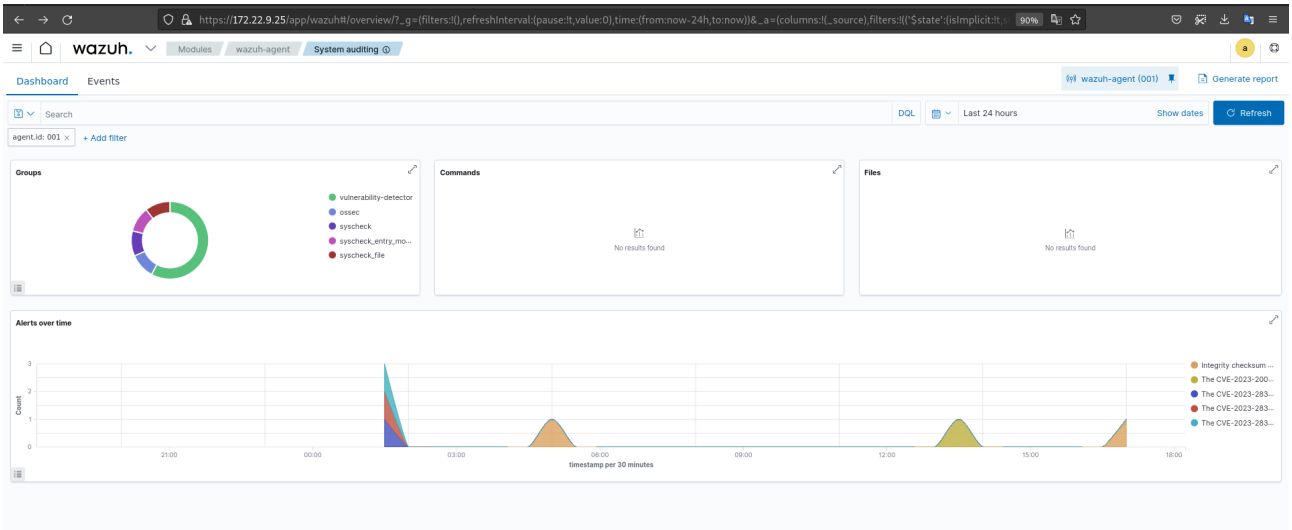
En el apartado de monitorización de la integridad, nos muestra gráficos con alertas de modificación, borrado o añadido, los agentes que lo realizan y los usuarios.



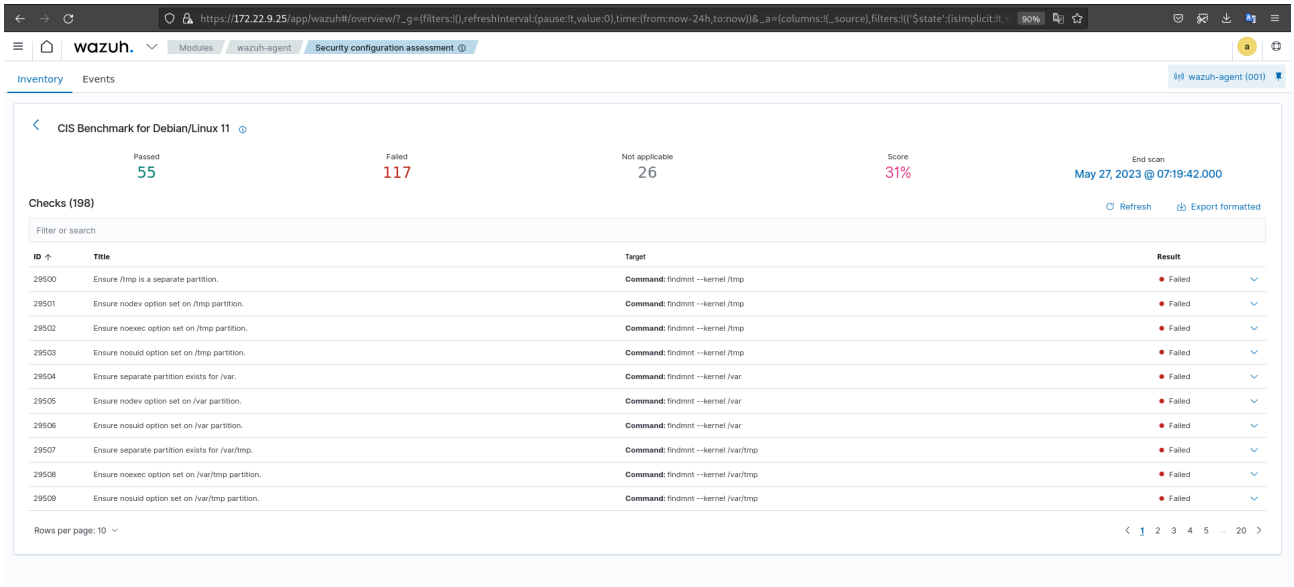
Ahora, veremos en detalle el apartado de auditoría y políticas de monitorización. La herramienta de monitorización de politicas, verifica que nuestro sistema esté configurado acorde a las politícas de seguridad que hemos indicado.



En el apartado auditoría del sistema, audita el comportamiento de los usuarios, supervisando la ejecución de comandos y alertando sobre el acceso a archivos críticos.

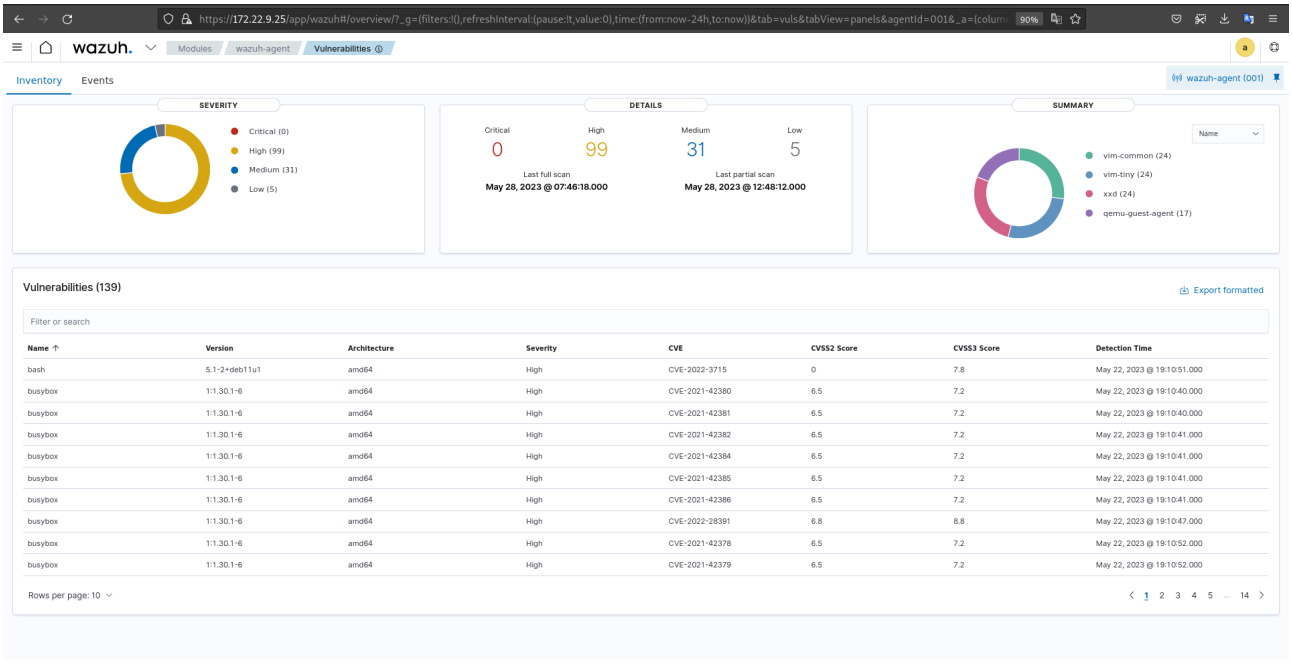


En el apartado de evaluación de la configuración de seguridad(SCA), escanea el agente como parte de una auditoría de evaluación de la configuración.



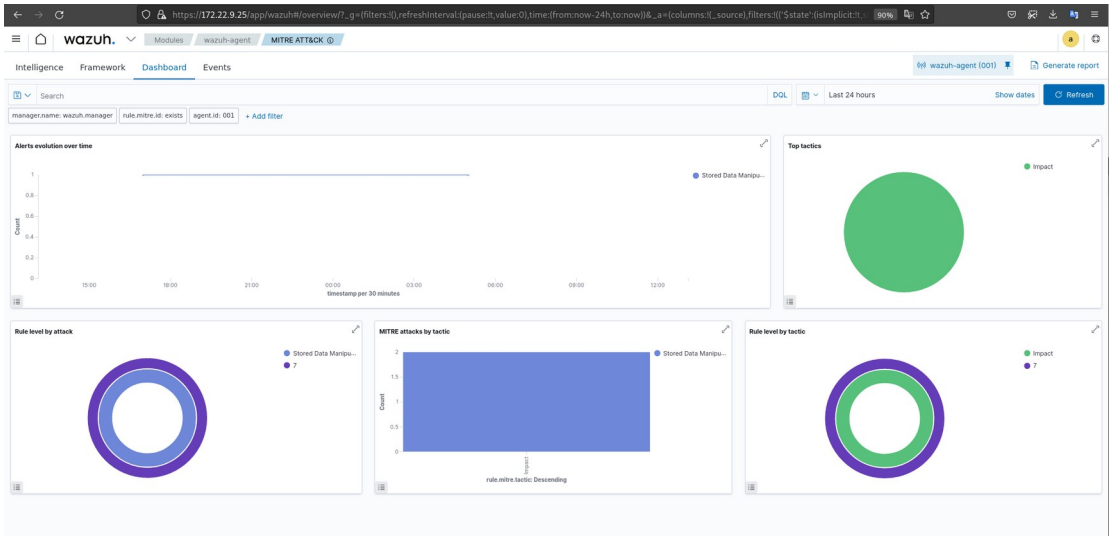
Nos muestra los chequeo que realiza, el comando y el resultado. Arriba nos muestra las evaluaciones exitosas, las que han fallado y las que no son aplicables y nos da un porcentaje que es la puntuación, cuanto mas alto es el porcentaje, significa que más chequeos han pasado la prueba.

Ahora veremos el apartado de detección y respuesta ante amenazas, la herramienta de vulnerabilidades nos muestra las aplicaciones de nuestro escenario que están afectadas por vulnerabilidades bien conocidas.



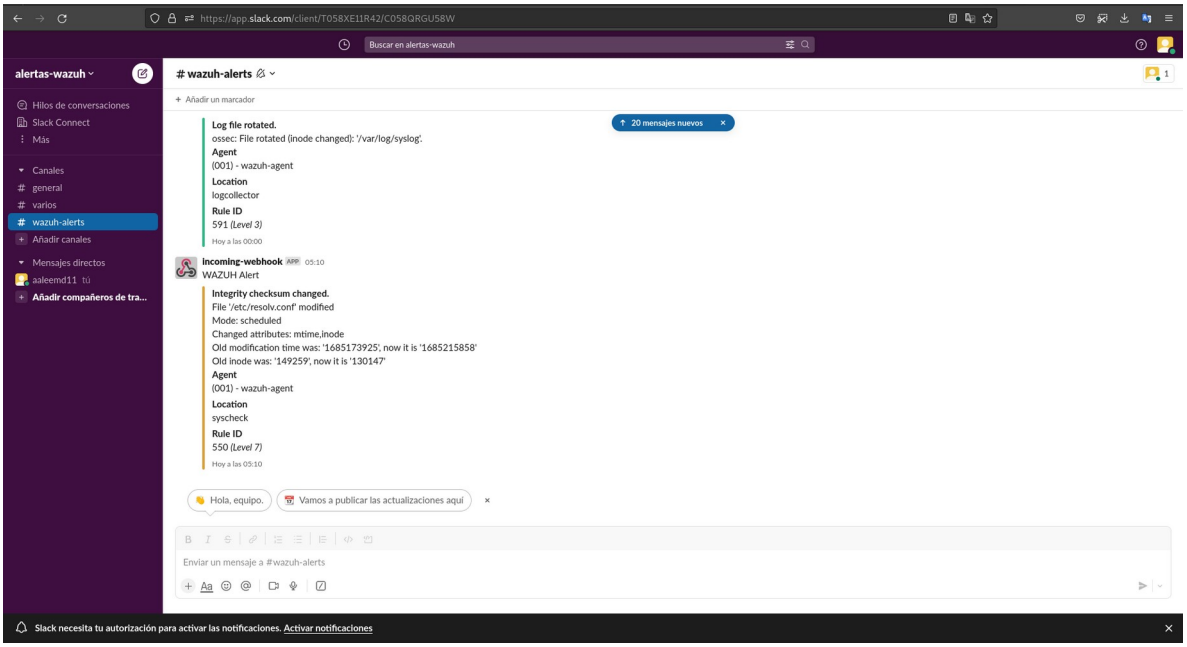
Como vemos, nos muestra un gráfico con la gravedad de la vulnerabilidad, también nos detalla el numero de vulnerabilidades según su gravedad y un gráfico con las aplicaciones afectadas. En la parte de abajo, nos indica todas las vulnerabilidades con el nombre, la versión, la arquitectura de la máquina, el nivel de gravedad, el CVE-ID que es una lista donde se encuentra registradas todas las vulnerabilidades y nos muestra puntuaciones del nivel de gravedad y la fecha en la que ha sido detectada la vulnerabilidad.

Por último, veremos el apartado de MITRE ATT&CK que indica eventos de seguridad de la base de conocimientos y técnicas de los adversarios basados en observaciones del mundo real.



5.2. Prueba de funcionamiento de las alertas en Slack.

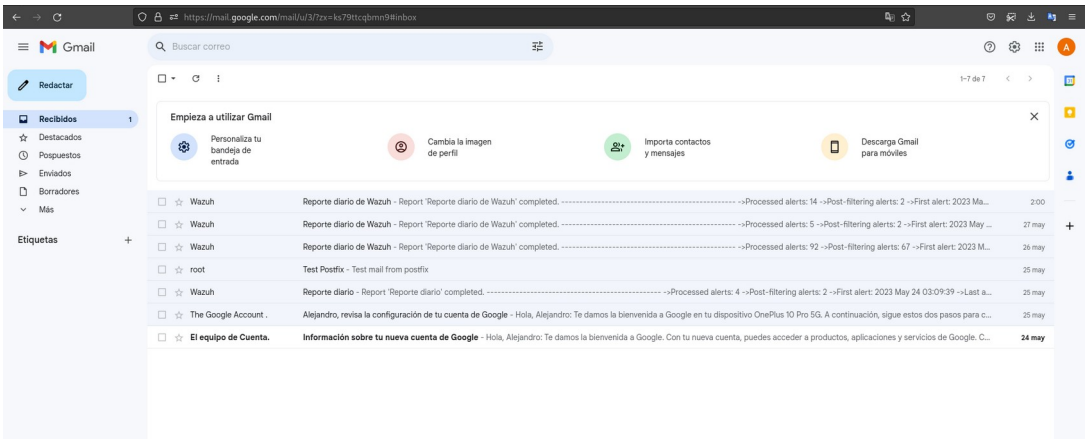
A continuación, voy a demostrar que el servidor Wazuh envía correctamente las alertas mediante Slack entrando en el canal de alertas-wazuh.



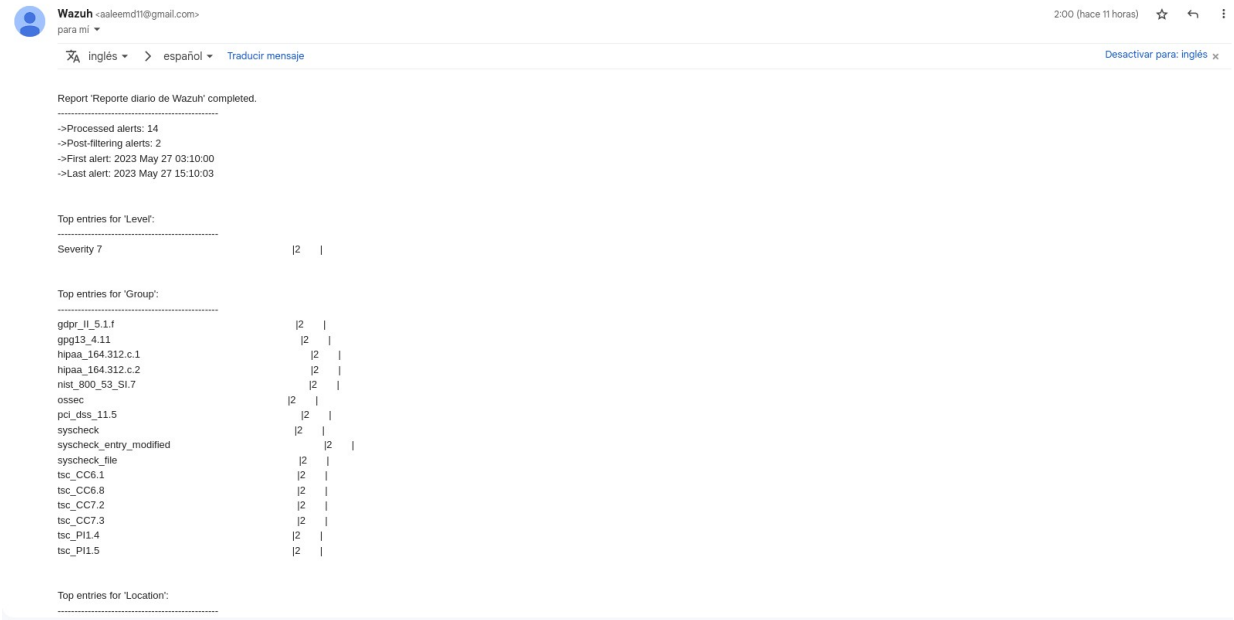
Como se puede apreciar en la imagen, vemos una alerta de nivel 7 y nos indica que el fichero /etc/resolv.conf ha sido modificado y que los atributos afectados son el mtime y el inodo.

5.3. Prueba de funcionamiento del reporte diario vía mail.

Para ello iniciaré sesión en la cuenta de gmail la cual indiqué que iba a recibir los correos.



Como podemos ver, hay 4 correos enviados a través del nombre Wazuh con el asunto Reporte diario de Wazuh, el primer correo en el asunto indica reporte diario porque más adelante decidí cambiar el asunto para indicara el nombre de la herramienta. Ahora muestro el contenido del correo y lo explicaré brevemente.



Primero nos muestra el numero de alertas procesadas y el número de alertas post filtrado, no indica cuando fue la primera alerta del día y la última, nos indica cual fue el nivel mas alto de alerta y por cada grupo cuantas alertas hubo.

6. Conclusiones.

Este proyecto a nivel me personal me ha resultado muy positivo ya que he aprendido mucho sobre una herramienta la cual desconocía, en cuanto al despliegue he tenido algunos problemas ya que en primera instancia mi idea era usar OpenStack y Kubernetes pero no ha sido posible debido a que en la red del instituto el puerto 80 está capado por Andared y he tenido muchos problemas para el despliegue así que decidí usar una instancia en OpenStack y Docker para el despliegue del contenedor. Claramente para un entorno de pruebas y un escenario pequeño sí es viable desplegar un contenedor para ver el funcionamiento de dicha herramienta pero en un escenario real es necesario el despliegue de un clúster.

La mayor ventaja de Wazuh es que es una herramienta muy completa que ofrece muchas funcionalidades y es modular, es decir, se pueden añadir módulos externos para complementar dicha herramienta.

Una de las principales desventajas que veo en esta herramienta es que sólo soporta sistemas operativos conocidos y no se puede instalar en un router o en un switch o un firewall etc.

Para seguir profundizando creo que sería interesante añadir otros módulos, ver como se instala y su funcionamiento, por otro lado también sería interesante indagar en el apartado de *Regulatory Compliance* aunque yo he decidido no verlo en este proyecto ya que no lo veo tan interesante, está mas enfocado al cumplimiento normativo de una empresa y haría demasiado largo el proyecto.

7. Bibliografía.

- Instalación de Docker: https://docs.docker.com/engine/install/debian/

- Despliegue de wazuh: https://documentation.wazuh.com/current/deployment-options/docker/wazuh-container.html

- Instalación de wazuh agent: https://documentation.wazuh.com/4.4/installation-guide/wazuh-agent/wazuh-agent-package-linux.html

- Activar detección de vulnerabilidades:https://tutorialesit.com/habilitar-la-deteccion-de-vulnerabilidades-en-wazuh/

- Configurar SMTP: https://documentation.wazuh.com/current/user-manual/manager/manual-email-report/smtp-authentication.html

- Configurar alertas por e-mail: https://documentation.wazuh.com/current/user-manual/manager/manual-email-report/index.html

- Configurar reporte diario e-mail: https://documentation.wazuh.com/current/user-manual/manager/automatic-reports.html

- Integrar wazuh con API: https://documentation.wazuh.com/current/user-manual/manager/manual-integration.html

- Envío de alertas por Slack: https://www.youtube.com/watch?v=9SaPUxSxTy0