

# Manual Técnico

**Integración XDR- Plataform y SIEM Wazuh**

# Contenido del instructivo

- Introducción..... 3
- Infraestructura requerida..... 4
- Configuración ..... 4
  - Configuración del punto final del conector de Ubuntu ..... 5
    - Prueba del conector ..... 7
  - Uso de la API de la XDR Plataform ..... 8
  - Exportación de registros de la XDR Plataform al punto final del conector de Ubuntu 8
  - Configuración de Rsyslog en el punto final del conector de Ubuntu ..... 10
- Configuración del servidor Wazuh..... 11
  - Decodificador personalizado ..... 14
  - Reglas personalizadas..... 15

## Introducción

En el entorno actual, las organizaciones enfrentan amenazas cada vez más avanzadas, lo que hace necesaria una estrategia de seguridad proactiva y en múltiples capas. Para responder eficazmente, muchas empresas implementan diversas soluciones de seguridad con el fin de fortalecer su postura defensiva. Sin embargo, el uso de múltiples herramientas requiere una **visibilidad centralizada** que facilite la eficiencia operativa y la respuesta ante incidentes.

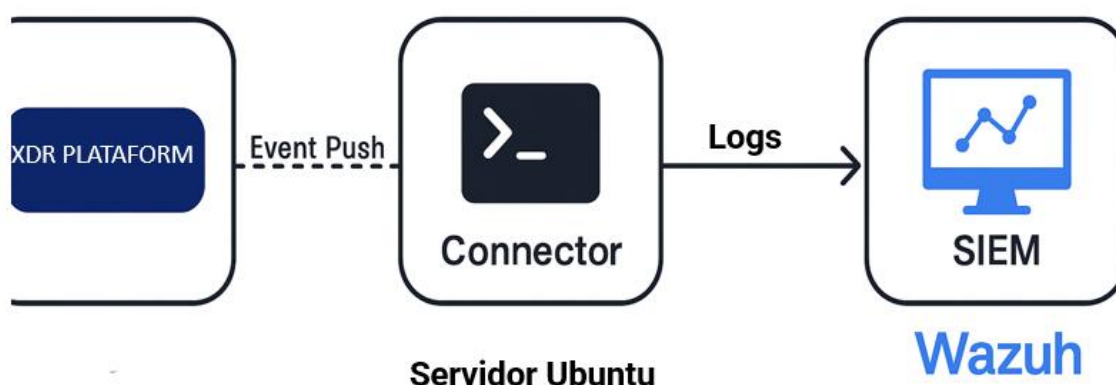
**Wazuh**, una plataforma de código abierto que combina capacidades de **SIEM** (Gestión de Información y Eventos de Seguridad) y **XDR** (Extended Endpoint Detection and Response), permite unificar, correlacionar y analizar eventos de seguridad provenientes de diversas fuentes.

En esta documentación se describe cómo integrar **XDR Platform** con Wazuh, permitiendo reenviar los registros generados por GravityZone hacia el servidor Wazuh. Esta integración proporciona una vista unificada de los eventos de seguridad, lo que mejora las capacidades de detección, respuesta y monitoreo continuo por parte de los equipos de ciberseguridad.

## Infraestructura requerida

Utilizamos la siguiente infraestructura para la integración de *XDR Plataform* con Wazuh.

- Componentes centrales de Wazuh (servidor Wazuh, indexador Wazuh y panel de control Wazuh).
- Un Servidor Ubuntu 24.04 LTS con un requisito mínimo de hardware de 2 GB de RAM, 80 GB de disco duro, 1 procesador, NIC virtual de 1 Gbit y una dirección IP pública que actúe como conector de registros. El endpoint del conector Ubuntu almacena los datos de syslog de la XDR Plataform y los reenvía al servidor Wazuh.
- Una cuenta de la XDR Plataform.



## Configuración

Realizaremos los siguientes pasos de configuración.

- Configuración del punto final del conector de Ubuntu .
- Uso de la API de la XDR Plataform .
- Configuración de Rsyslog en el punto final del conector de Ubuntu .
- Configuración del servidor Wazuh

## Configuración del punto final del conector de Ubuntu

El endpoint de Ubuntu funciona como servidor conector entre la XDR Plataforma y Wazuh. Siga estos pasos en el endpoint de Ubuntu para configurar la recopilación de registros de la XDR Plataforma mediante la API:

Estos pasos se omitirán ya que contienen información referente a la XDR Plataforma

Una vez que se haya realizado con éxito la configuración recomendada por el proveedor y la marca de la XDR Plataforma, sigue validar certificados.

## Certificados

Para asegurar que los eventos enviados desde la plataforma XDR hacia el conector (o directamente al SIEM) viajan de forma segura y no pueden ser interceptados o alterados en tránsito, se usa el protocolo HTTPS/TLS, que requiere un certificado válido. Si no se utiliza HTTPS o si el certificado no es validado correctamente, la transmisión de eventos puede fallar o quedar expuesta a ataques de intermediarios (MITM).

Para la integración entre la plataforma XDR y Wazuh SIEM fue necesario habilitar certificados TLS/SSL con el fin de:

- Asegurar la confidencialidad de los eventos en tránsito (cifrado punto a punto).
- Autenticar el origen y destino de la comunicación, evitando conexiones con servidores no autorizados.
- Cumplir con políticas corporativas que exigen el uso de certificados emitidos por una Autoridad Certificadora (CA) pública y confiable.

En este proyecto se utilizó Let's Encrypt como proveedor de certificados SSL/TLS:

- Permite obtener certificados gratuitos y renovables automáticamente.
- Es reconocido por los navegadores y sistemas operativos, lo que evita errores de validación SSL.
- Garantiza que la comunicación entre el conector y el SIEM se realice sobre un canal seguro y confiable.

Entonces por defecto luego de ejecutar en el paso anterior de la configuración del conector **config.sh** se generan 2 certificados auto firmados para el servidor del conector HTTPS, se pueden visualizar en este apartado:

**api/config/server.key → certificado público**

**api/config/server.crt → llave privada**

Por seguridad se recomienda utilizar certificados validados por una autoridad de certificación CA pública de confianza.

Como los certificados emitidos en la configuración del conector son autoafirmados, se modificaron por certificados de Let's Encrypt:

Normalmente Let's Encrypt guarda los archivos en:

**/etc/letsencrypt/live/<tu-dominio>/**

Allí están:

- privkey.pem → úsalo como server.key
- fullchain.pem → úsalo como server.crt

Para revisarlos puedes usar **ls -l /etc/letsencrypt/live/tu-dominio/**

Copia al directorio del conector:

```
sudo cp /etc/letsencrypt/live/tu-dominio/privkey.pem /opt/XDR-plataform/xdr/api/config/server.key
```

```
sudo cp /etc/letsencrypt/live/tu-dominio/fullchain.pem /opt/XDR-plataform/xdr/api/config/server.crt
```

Ajusta permisos para que el conector pueda leerlos:

```
sudo chmod 600 /opt/XDR-plataform/xdr/api/config/server.key
```

```
sudo chmod 644 /opt/XDR-plataform/xdr/api/config/server.crt
```

Al final puedes revisarlo

```
sudo ls -l /opt/XDR-plataform/xdr/api/config/
```

## Opcional

Si quieres ver solo el nombre del dominio del certificado:

```
sudo openssl x509 -in /opt/XDR-plataform/xdr/api/config/server.crt -noout -subject
```

Y para verificación rápida de la expiración:

```
sudo openssl x509 -in /opt/XDR-plataform/xdr/api/config/server.crt -noout -dates
```

Para verificar qué certificado está realmente dentro de server.crt, puedes usar OpenSSL:

```
sudo openssl x509 -in /opt/XDR-plataform/xdr/api/config/server.crt -text -noout
```

Una vez validados los certificados se debe Habilitar el servicio de envío de eventos en el punto final del conector de Ubuntu e Iniciar el servicio de envío de eventos.

Importante: Puedes utilizar cualquier CA pública.

## Prueba del conector

El conector configurado se puede probar enviando una carga útil al servicio de recopilación mediante el siguiente comando. Reemplace el encabezado de autorización y la URL con los detalles configurados previamente durante la ejecución del **config.sh** script . Tras la ejecución correcta de los siguientes comandos, se genera el código de respuesta OK.

Debe devolver:

```
maquina@xdr-conector/opt/XDR-plataform/xdr/api/config/$ sudo cat /opt/XDR-plataform/xdr/api/config/config.json { "port": 0000, "syslog_port": 514, "transport": "Tcp", "target": "x.x.x.x", "authentication_string": "Basic xxxxxxxxxxxxxxxxxxxxx=", "secure": { "enabled": true, "key": "api/config/server.key", "cert": "api/config/server.crt" } }
```

## Uso de la API de la XDR Plataforma

La integración aprovechó la **API de la plataforma XDR** para obtener y enviar información relevante hacia Wazuh. El uso de la API permitió:

- **Automatizar la recolección de eventos de seguridad expuestos por la API** como detecciones en endpoints, estado de módulos, ejecución de tareas, y cambios operativos relevantes (registro, instalación, movimientos, etc.).
- **Facilitar la autenticación y comunicación segura** mediante credenciales API y certificados TLS.
- **Exportar eventos en formatos estandarizados** (CEF/JSON), lo que simplificó la normalización dentro de Wazuh.
- **Integración flexible:** al ser una API abierta, se pudieron implementar scripts y conectores que adaptan la salida de la XDR Platform a los requerimientos del SIEM.

## Exportación de registros de la XDR Plataforma al punto final del conector de Ubuntu

Los registros de la XDR Plataforma se envían primero al endpoint del conector de Ubuntu y luego se reenvían al servidor Wazuh para su análisis.

En esta sección se debe habilitar el envío de eventos de seguridad desde la plataforma XDR hacia el conector, se configuró la API de eventos mediante autenticación basada en clave API codificada en Base64.

Una vez aplicada la configuración, se validó exitosamente la recepción de eventos desde la plataforma en el conector, confirmando el enlace seguro y funcional hacia el SIEM.

Importante se debe validar conectividad con el puerto usado por la XDR Plataforma o en algunos casos utilizan ciertas IP las cuales se deberán agregar en lista blancas del firewall.

Para este proyecto como se desarrolló en Google Cloud, se realizaron las respectivas reglas en Firewall.



<input type="checkbox"/>	Nombre	Tipo	Destinos	Filtros	Protocolos/puertos	Acción	Prioridad	Red <span>↑</span>
<input type="checkbox"/>	<a href="#">XDR PLATAFORM</a>	Entrada	Aplicar a	Rangos de IP:	tcp: 0000	Permitir	900	<a href="#">default</a>
<input type="checkbox"/>	<a href="#">allow-wazuh-services</a>	Entrada	Aplicar a	Rangos de IP:	tcp:443, 1514, 1515, 9200 udp:1514	Permitir	1000	<a href="#">default</a>
<input type="checkbox"/>	<a href="#">dashboard-wazuh5601</a>	Entrada	Aplicar a	Rangos de IP:	tcp:5601	Permitir	1000	<a href="#">default</a>
<input type="checkbox"/>	<a href="#">default-allow-http</a>	Entrada	http-server	Rangos de IP:	tcp:80	Permitir	1000	<a href="#">default</a>
<input type="checkbox"/>	<a href="#">default-allow-https</a>	Entrada	https-server	Rangos de IP:	tcp:443	Permitir	1000	<a href="#">default</a>
<input type="checkbox"/>	<a href="#">syslog-</a>	Entrada	Aplicar a	Rangos de IP:	tcp:514	Permitir	1000	<a href="#">default</a>

Se pueden realizar varias pruebas para determinar si existe conectividad con el puerto utilizado por la XDR Plataforma.

- **Test-NetConnection -ComputerName TUIPPUBLICACONECTOR -Port 0000**
  - Si responde, el puerto está accesible.
  - Si da timeout o failed to connect, significa que el tráfico está bloqueado.
- **nc -vz TUIPPUBLICACONECTOR 0000**
  - Test con nc (netcat) desde otra instancia
  - succeeded → puerto accesible
  - failed → bloqueado (ver firewall, reglas de GCP o UFW en el conector)
- **sudo lsof -ltcp -Stcp:listen -P | grep 0000**
  - Comprueba desde el conector que escucha correctamente
  - Debe mostrar algo como: node 46416 TCP \*:0000 (LISTEN)
  - Confirma que NodeJS está escuchando en todas las interfaces (\*) y no solo en 127.0.0.1

*Si falla y no tienes comunicación con la IP o el puerto, verifica en el firewall perimetral o de la maquina Ubuntu. También puedes hacer ping a las IP que proporciona la XDR Plataforma e incluyendo el puerto todas deben tener conectividad.*

**Nota :** Los registros exportados desde la XDR Plataforma se pueden encontrar en la ruta del archivo:

```
/opt/xdr-plataform/var/log/xdr/log.txt
```

```
sudo tail -f /var/log/syslog | grep "CEF"
```

El parámetro -f mantiene el seguimiento en tiempo real, mostrando nuevas entradas a medida que llegan.

Asi deberían visualizarse los logs que estan llegando al conector.

```

Event key = 2 is - CFEI0 [6.64.0-1|190000|Network Attack Defense|10] network-monitor
dvchost=- CFONCE [redacted] it-sd-cponce. dvc=- deviceExternalId= B
EndpointId=5d42fe MainAction=block DetectionName=Attack.BruteForce.RDP DetectionAttackT
echnique=CredentialAccess DetectionAttackerIp=91. 96 | DetectionVictimIp=200. F DetectionLocalPort=3389 B
DetectionTime=2025-08-17T17:18:24.000Z

Event key = 3 is - CFEI0 [6.64.0-1|170000|New Incident|3] Module=new-incident dvchost=J20 1245 ComputerFQ
DN=20cac11 dvc=2000. IncidentId=68a20eaf529 DeviceExternalId=66 029284 IncidentId=68a20eaf529
SeverityScore=50 AttackEventId=12372329 MainAction=blocked DetectionName=Exploit.SMB.CVE-2017-0143.Do
ubUppercase request=10 spt=445 src= 12 sproc=cysystem AttackTypes=["Malware","Exploit","Other"]
T1190,"T1571","T1543","T1021","T1569","T1135","T1095"] start=2025-08-17T17:18:38.476Z
Id=664e0 @ AttackId=["T1071"] Endpoint

```

## Configuración de Rsyslog en el punto final del conector de Ubuntu

En esta sección, instalamos y configuramos Rsyslog para reenviar los eventos desde el punto final del conector de Ubuntu al administrador de Wazuh. Aunque algunas distribuciones de Linux tienen Rsyslog preinstalado, puede instalarlo en el punto final del conector de Ubuntu siguiendo los pasos que se describen a continuación si aún no lo tiene instalado.

1. Ejecute los siguientes comandos para instalar Rsyslog:

```
# apt-get update
# apt-get install -y rsyslog
```

2. Modifique el **/etc/rsyslog.conf** archivo descomentando los parámetros a continuación para habilitar el registro remoto. Por ejemplo, en nuestro caso usamos TCP

- Para TCP
  - `módulo(carga="imtcp")`
  - `entrada(tipo="imtcp" puerto="514")`
- For UDP
  - `module(load="imudp")`
  - `input(type="imudp" port="514")`

### 3. Inicie y habilite Rsyslog:

```
#systemctl start rsyslog
#systemctl enable Rsyslog
```

Con esta configuración ya el conector deberá estar recibiendo los logs desde la XDR Plataforma los puedes ver


```
/opt/xdr-plataform/var/log/xdr-event/log.txt
sudo cat /opt/xdr-plataform/var/log/xdr-event /log.txt
```

Entonces podemos verificar si los logs **realmente están llegando al Wazuh server** usando tcpdump.

Ejecuta esto **en el Wazuh server** mientras envías un log de prueba desde el conector o se debería ver el envío de logs:

Explicación rápida:

- -i any → escucha en **todas las interfaces de red**.
- port 514 → filtra solo tráfico que llega al puerto 514 (donde está escuchando Wazuh).
- -nn → no resuelve nombres de host ni puertos, muestra direcciones y números de puerto.
- -A → muestra el contenido en ASCII para leer los logs.



## Configuración del servidor Wazuh

Realice los siguientes pasos en el servidor Wazuh para configurar la recopilación de syslog desde el punto final del conector de Ubuntu.

1. Agregue lo siguiente al **<ossec\_config>** bloque del **/var/ossec/etc/ossec.conf** archivo de configuración:

```
<ossec_config>
<remote>

  <connection>syslog</connection>
    <port><PORT></port>
    <protocol><PROTOCOL></protocol>
```

```
<allowed-ips><ALLOWED_IP></allowed-ips>
<local_ip><WAZUH_MANAGER_IP></local_ip>
</remote>
</ossec_config>
```

- **Connection** Indica el tipo de conexión entrante que se permitirá. En este caso, utilizamos syslog, un protocolo estándar para la transmisión de mensajes de registro a través de una red.
- **PORT:** Especifica el número de puerto de red para la conexión remota. Reemplázelo **<PORT>** con el número de puerto deseado. Recuerde que debe configurarse el mismo número de puerto en el punto final del conector de Ubuntu.
- **PROTOCOL** Indica el protocolo de comunicación. Reemplázelo **<PROTOCOL>** con el protocolo seleccionado, ya sea TCP o UDP. Asegúrese de configurar el mismo protocolo en el punto final del conector de Ubuntu.
- **Allowed-ips** Indica qué puntos finales pueden enviar registros a este servidor. Reemplázelo **<ALLOWED\_IP>** con la dirección IP del punto final del conector de Ubuntu.
- **Local-ip:** Identifica la dirección IP local del servidor Wazuh que se usará para esta conexión. Reemplázela **<WAZUH\_MANAGER\_IP>** con la dirección IP de su servidor Wazuh.

2. Agregue la siguiente configuración al **/var/ossec/etc/ossec.conf** archivo desde el que se leen los registros de Bitdefender.

```
<ossec_config>
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/messages</location>
</localfile>
</ossec_config>
```

TENER EN CUENTA, **<location>/var/log/messages</location>** la recomendación es **<location>/var/log/syslog</location>**.

3. Reinicie el administrador de Wazuh para aplicar los cambios:

```
#systemctl restart wazuh-manager
```

4. Revise los registros de syslog **/var/log/messages** para confirmar que los registros de BitDefender Gravity Zone se reciben en el servidor Wazuh:

```
cat /var/log/(messages o syslog)| grep xdr-plataform
```

Puede usar, `sudo ss -tulnp | grep 514 tcp` y ver quien está recibiendo por el puerto 514

```
sudo ss -tulnp | grep 514 tcp
```

```
LISTEN 0 128 10.128.0.10:514 0.0.0.0:* users:(("wazuh-remoted",pid=130144,fd=4))
```

El **puerto 514/TCP** en tu **Wazuh Server** está **escuchando**, pero no por rsyslog, sino por **wazuh-remoted**.

wazuh-remoted es el componente de Wazuh que recibe **logs remotos** (syslog, agentes, conectores) directamente, sin pasar por /var/log/syslog.

*En caso de que te suceda*

Los eventos se procesan directamente y se escriben en:

```
/var/ossec/logs/alerts/alerts.json
```

Esto es **normal en instalaciones modernas**: Wazuh no depende de rsyslog en el servidor para recibir eventos de agentes/conectores.

La instrucción de "Revisar /var/log/syslog" en la documentación es **opcional**, solo sirve si quieres auditar el tráfico en crudo antes de que lo procese Wazuh.

La prueba real de que los logs llegan correctamente es que **aparezcan en alerts.json** o en el **Dashboard**.

### 1. Ubicación de los logs en el sistema (Linux)

Wazuh guarda los logs procesados por el manager en varios archivos según el tipo de actividad:

- **/var/ossec/logs/ossec.log** → principal log del Wazuh Manager: errores de arranque, problemas de decoders, alertas internas.

- **/var/ossec/logs/alerts/alerts.json** → alertas generadas por decoders y reglas en formato JSON (para que las consuma Kibana/Wazuh Dashboard).
- **/var/ossec/logs/alerts/alerts.log** → versión en texto plano de las alertas.
- **/var/ossec/logs/active-responses.log** → ejecución de respuestas activas.

En tu caso, todos los eventos de GravityZone que están disparando la regla **100601** aparecerán en **alerts.json** y **alerts.log**.

## 2. Visualización en tiempo real

Puedes hacer tail para ver los logs en tiempo real mientras llegan:

```
sudo tail -f /var/ossec/logs/alerts/alerts.json
```

```
sudo tail -f /var/ossec/logs/ossec.log
```

Esto te permite verificar que los eventos están siendo parseados correctamente por el decoder y disparando la regla.

## Decodificador personalizado

Siga los pasos a continuación para agregar un decodificador personalizado SOLO PARA ANTIMAWLARE al servidor Wazuh para decodificar los registros de xdr-plataform.

1. Crea un archivo en el directorio: `/var/ossec/etc/decoders`

```
# touch /var/ossec/etc/decoders/xdr-plataform.xml
```

2. Agregue el siguiente decodificador:

```
<decoder name="xdr-plataform">
  <program_name>CEF</program_name>
  <prematch>xdr-plataform</prematch>
</decoder>
```

```
<decoder name="xdr-plataform_child">
  <parent>xdr-plataform</parent>
```

```
<regex type="pcre2">.*(AntiMalware).*dvchost=(\S+) \S+ dvc=(\S+) \S+
XDRMalwareType=(\S+) XDRMalwareName=(.*) XDRMalwareHash=(\S+) act=(\S+)
filePath=(\S+) XDRDetectionTime=(\S+).*XDRCleanedMalwareCnt=(\d+)
XDRBlockedMalwareCnt=(\d+) XDRDeletedMalwareCnt=(\d+)
XDRQuarantinedMalwareCnt=(\d+) XDRIgnoredMalwareCnt=(\d+)
XDRPresentMalwareCnt=(\d+) suser=(\S+).*</regex>
```

```
<order>Module,DeviceHostName,DeviceIP,MalwareType,MalwareName,MalwareHash,
Action,FilePath,DetectionTime,CleanedMalwareCount,BlockedMalwareCount,DeletedM
alwareCount,QuarantinedMalwareCount,IgnoredMalwareCount,PresentMalwareCount,
User</order>
```

```
</decoder>
```

## Reglas personalizadas

Agregue las siguientes reglas personalizadas al servidor Wazuh para generar alertas cuando se reciben los registros de Bitdefender.

1. Crea un archivo **xdr-plataform.xml** en el directorio **/var/ossec/etc/rules/**

```
# touch /var/ossec/etc/rules/xdr-plataform.xml
```

2. Agregue las siguientes reglas personalizadas al archivo

**/var/ossec/etc/rules/xdr-plataform.xml**

```
<!-- XDR Platform logs grouped -->
<group name="xdr-plataform">
  <rule id="100600" level="0">
    <decoded_as>xdr-plataform</decoded_as>
    <description>Logs from XDR Platform.</description>
  </rule>
  <!-- Rule to detect active threat -->
  <rule id="100601" level="10">
    <if_sid>100600</if_sid>
    <field name="Module">AntiMalware</field>
```

<description>XDR Platform: \$(MalwareName) detected on \$(DeviceHostName) and \$(Action).</description>

</rule>

</group>

Dónde:

- Los ID de regla 100600 agrupan todos los eventos de Bitdefender.
- La identificación de regla 100601 se activa cuando se detecta y elimina una amenaza en un punto final monitoreado.

3. Reinicie el servicio del administrador de Wazuh para aplicar los cambios:

```
#systemctl restart wazuh-manager
```

Al finalizar la implementación los eventos se muestran de la siguiente manera en el SIEM:

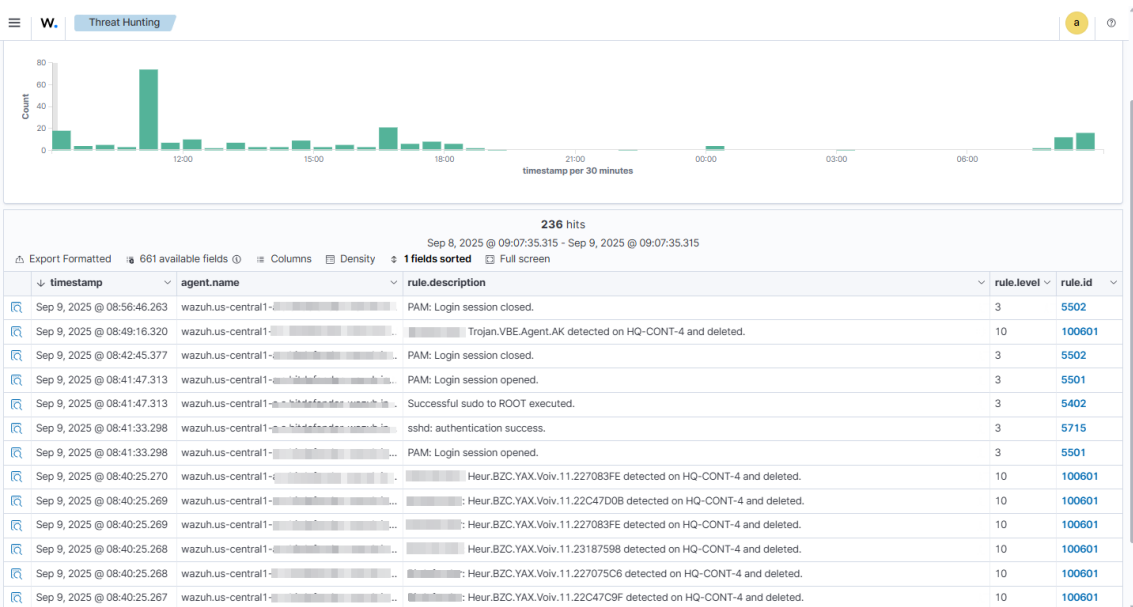


Figura 1 Alertas de XDR-Plataform en el panel de Wazuh.



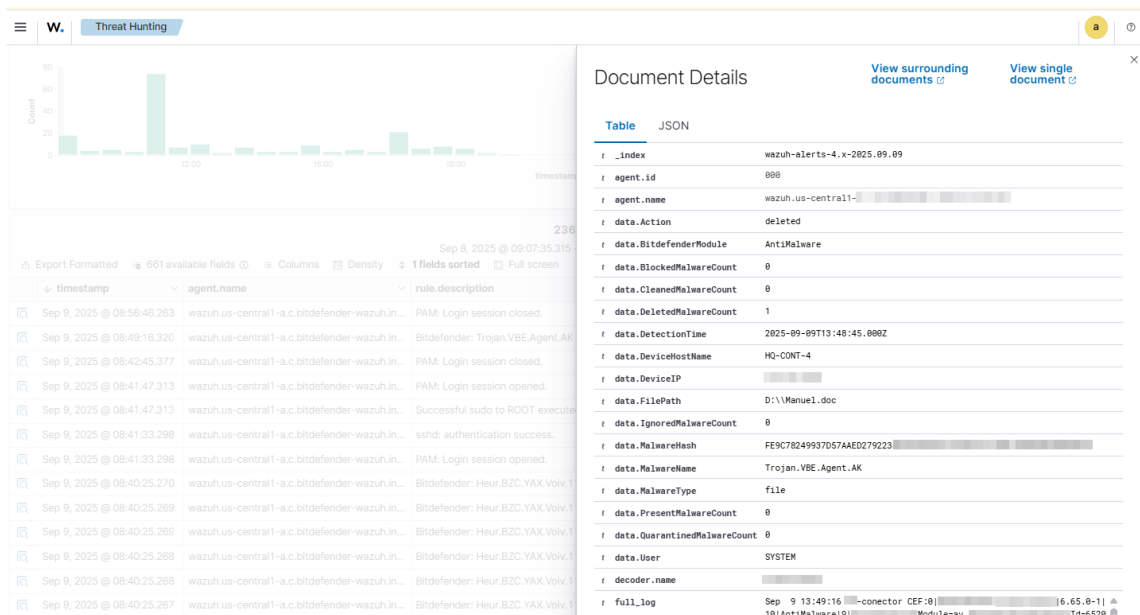


Figura 2 Detalles de alerta de XDR-Plataform en el panel de Wazuh.