

Wazuh

Introducción

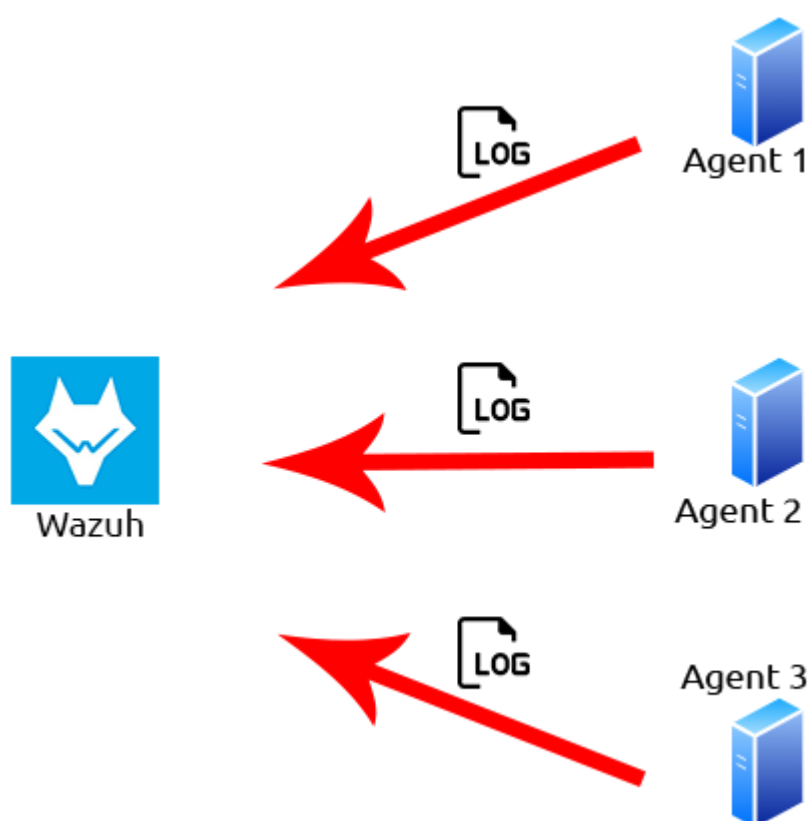
Bienvenido a una sala que muestra las capacidades de la solución de software Wazuh EDR . En esta sala, aprenderá lo siguiente:

- ¿Qué es un EDR y por qué son soluciones útiles?
- Dónde se utiliza un EDR como Wazuh
- Accediendo a Wazuh
- Navegando por Wazuh
- Conozca las reglas y alertas de Wazuh
- Digerir registros para ver eventos específicos en dispositivos, incluidos Linux y Windows
- Cómo puedes ampliar Wazuh usando complementos y su API

En primer lugar, comprendamos qué son exactamente las soluciones EDR . La detección y respuesta de endpoints (EDR) consiste en una serie de herramientas y aplicaciones que monitorizan los dispositivos para detectar cualquier actividad que pueda indicar una amenaza o una brecha de seguridad. Estas herramientas y aplicaciones incluyen:

- Auditar un dispositivo para detectar vulnerabilidades comunes
- Monitoreo proactivo de un dispositivo para detectar actividades sospechosas, como inicios de sesión no autorizados, ataques de fuerza bruta o escaladas de privilegios.
- Visualizar datos y eventos complejos en gráficos claros y modernos
- Registrar el comportamiento operativo normal de un dispositivo para ayudar a detectar anomalías

Creada en 2015, **Wazuh** es una solución EDR de código abierto, gratuita y extensa . Puede utilizarse en entornos de todas las escalas. Wazuh opera con un módulo de administración y agentes. En pocas palabras, un dispositivo, denominado administrador, se encarga de ejecutar Wazuh . Wazuh opera con un modelo de administración y agentes, donde el administrador es responsable de administrar los agentes instalados en los dispositivos que se desean monitorear. Veamos este modelo en el diagrama a continuación:



Podemos ver registros de tres agentes que se envían al servidor Wazuh .

Responda las preguntas a continuación

¿Cuándo fue lanzado Wazuh?

2015

Respuesta correcta

¿Cuál es el término que Wazuh utiliza para denominar un dispositivo que está siendo monitoreado en busca de actividad sospechosa y posibles amenazas a la seguridad?

Agent

Respuesta correcta

Por último, ¿cuál es el término para un dispositivo que se encarga de gestionar estos dispositivos?

Manager

Respuesta correcta

Obligatorio: Implementar el servidor Wazuh

Conéctese a la red TryHackMe e implemente el servidor de administración de Wazuh adjunto a esta tarea y espere un **mínimo de cinco minutos** antes de visitar el servidor Wazuh en **HTTP ://MACHINE_IP** .

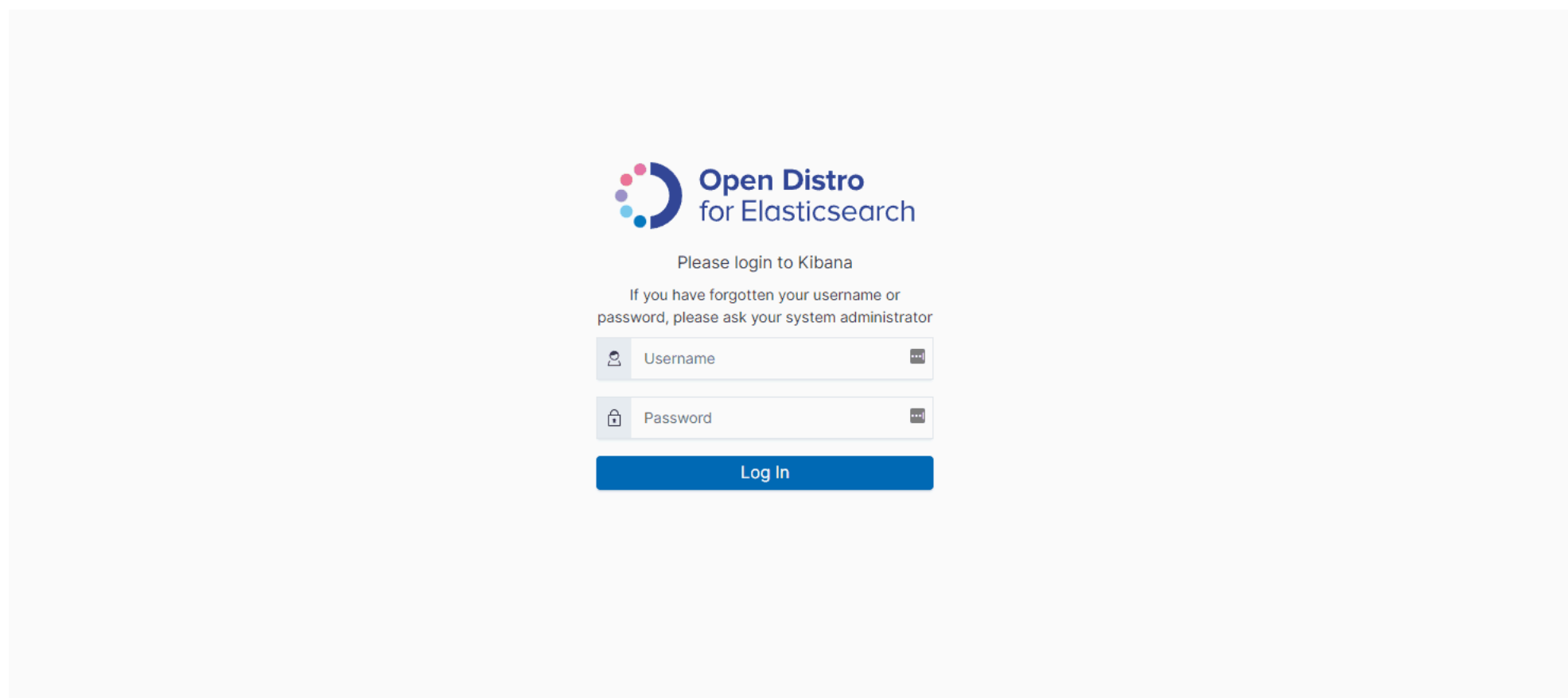
Si carga el servidor de administración de Wazuh demasiado pronto, aparecerá el mensaje " El servidor Kibana aún no está listo". Espere unos minutos más antes de actualizar la página e intentarlo nuevamente.

Una vez iniciado, inicie sesión utilizando las siguientes credenciales:

Nombre de usuario: wazuh (**¡asegúrese de que esté en minúsculas!**)

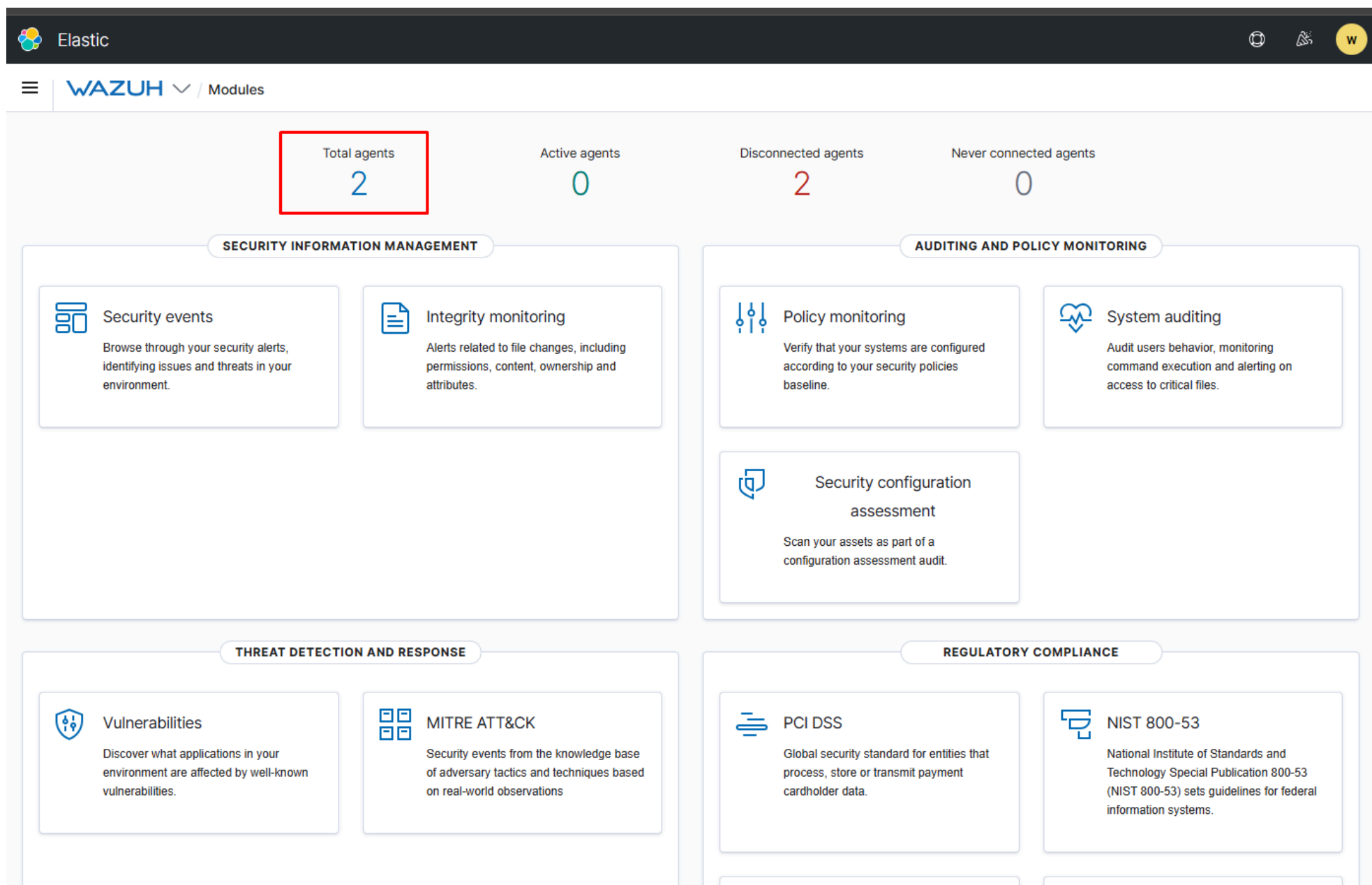
Contraseña: eYa0M1-hG0e7rjGi-IRB2qGYVoonsG1K

Seleccione "**Inquilino global**" después de iniciar sesión correctamente. Si tiene problemas, consulte el GIF animado a continuación del proceso.



Nota: Las preguntas dentro de las tareas de esta sala **esperarán los datos almacenados en este servidor de administración de Wazuh** , por lo que es vital que pueda conectarse a este servidor antes de continuar.

El servidor de administración de Wazuh en esta sala mostrará que los agentes están **desconectados** : esto es lo esperado.



Agentes de Wazuh

Los dispositivos que registran los eventos y procesos de un sistema se denominan agentes. Los agentes supervisan los procesos y eventos que ocurren en el dispositivo, como la autenticación y la gestión de usuarios. Los agentes transfieren estos registros a un recopilador designado para su procesamiento, como Wazuh .

Para que Wazuh se complete, es necesario instalar agentes en los dispositivos para registrar dichos eventos. Wazuh puede guiarle en el proceso de implementación del agente siempre que cumpla con algunos requisitos previos, como :

- **Sistema operativo**
- **La dirección del servidor Wazuh al que el agente debe enviar registros (puede ser una entrada DNS o una dirección IP)**
- **En qué grupo estará el agente: puedes ordenar los agentes en grupos dentro de Wazuh si lo deseas**

Este asistente se puede iniciar navegando a la siguiente ubicación en el servidor Wazuh : **Wazuh -> Agentes -> Implementar nuevo agente** como se ilustra en la siguiente captura de pantalla:

Elastic

WAZUH 1. Agents

Modules
Management
Agents
Tools
Security
Settings

STATUS

Active
Disconnected
Never connected

DETAILS

Active 2 Disconnected 0 Never connected 0 Agents coverage 100.00%

Last registered agent ip-10-10-73-118 Most active agent CHANGE-MY-HOSTNAME

EVOLUTION

active

Filter or search agent Refresh

Agents (2)

3. Deploy new agent Export formatted

ID ↑	Name	IP	Group(s)	OS	Cluster node	Ver...	Registratio...	Last keep al...	Status	Act...
001	CHANGE-MY-HO...	10.10.20.2...	default	Microsoft Win...	node01	v4....	Oct 7, 202...	Oct 14, 20...	●	👁️ 🔗
002	ip-10-10-73-118	10.10.73.1...	default	Ubuntu 20.04....	node01	v4....	Oct 7, 202...	Oct 14, 20...	●	👁️ 🔗

Una vez que acceda a esta pantalla, tendrá a su disposición el asistente intuitivo. He compartido capturas de pantalla del asistente para instalar el agente de Wazuh tanto en Windows como en Debian/Ubuntu. En la etapa 4, recibirá un comando para copiar y pegar en el portapapeles, lo que instalará y configurará el agente en el dispositivo del que desea recopilar registros.

Instalación del agente Wazuh en Windows:

1 Choose the Operating system

Red Hat / CentOS

Debian / Ubuntu

Windows

MacOS

2 Wazuh server address

You can predefine the Wazuh server address with the `enrollment.dns` Wazuh app setting.

wazuh.thm

3 Assign the agent to a group

Select one or more existing groups

default x



4 Install and enroll the agent

You can use this command to install and enroll the Wazuh agent in one or more hosts.

③ Running this command on a host with an agent already installed upgrades the agent package without enrolling the agent. To enroll it, see the [Wazuh documentation](#).

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.2.3-1.msi -OutFile wazuh-agent-4.2.3.msi; ./wazuh-agent-4.2.3.msi /q WAZUH_MANAGER='wazuh.thm' WAZUH_REGISTRATION_SERVER='wazuh.thm' WAZUH_AGENT_GROUP='default'
```

Instalación del agente Wazuh en Debian/Ubuntu:

1 Choose the Operating system

Red Hat / CentOS

Debian / Ubuntu

Windows

MacOS

2 Choose the architecture

i386

x86_64

armhf

aarch64

3 Wazuh server address

You can predefine the Wazuh server address with the `enrollment.dns` Wazuh app setting.

wazuh.thm

4 Assign the agent to a group

Select one or more existing groups

default ×

× ✓

5 Install and enroll the agent

You can use this command to install and enroll the Wazuh agent in one or more hosts.

ⓘ Running this command on a host with an agent already installed upgrades the agent package without enrolling the agent. To enroll it, see the [Wazuh documentation](#).

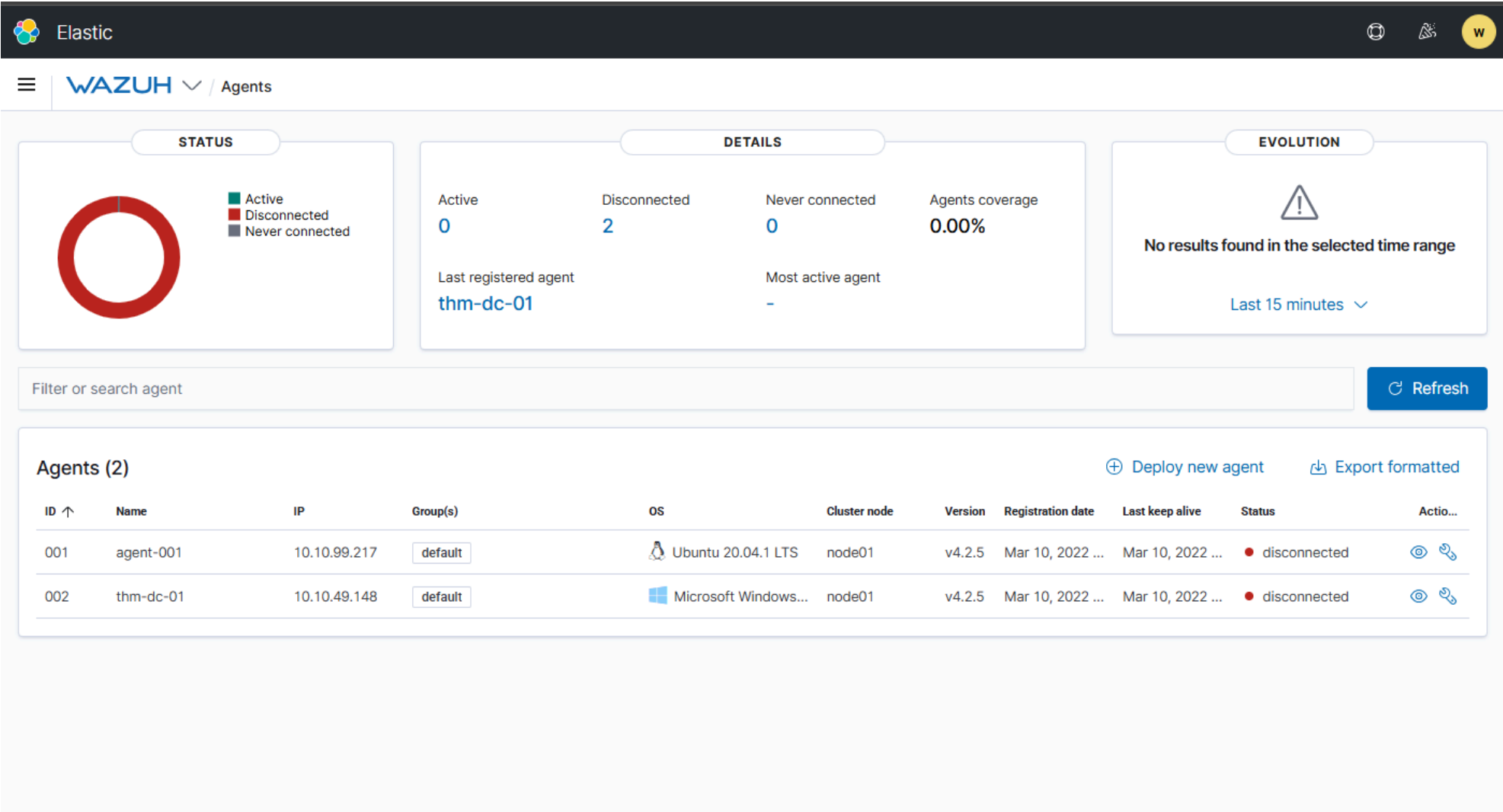
```
curl -so wazuh-agent-4.2.3.deb https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.2.3-1_amd64.deb && sudo WAZUH_MANAGER='wazuh.thm' WAZUH_AGENT_GROUP='default' dpkg -i ./wazuh-agent-4.2.3.deb
```

Responda las preguntas a continuación

Asegúrese de haber iniciado sesión en el servidor de administración de Wazuh en [HTTPS://10.10.141.81](https://10.10.141.81)

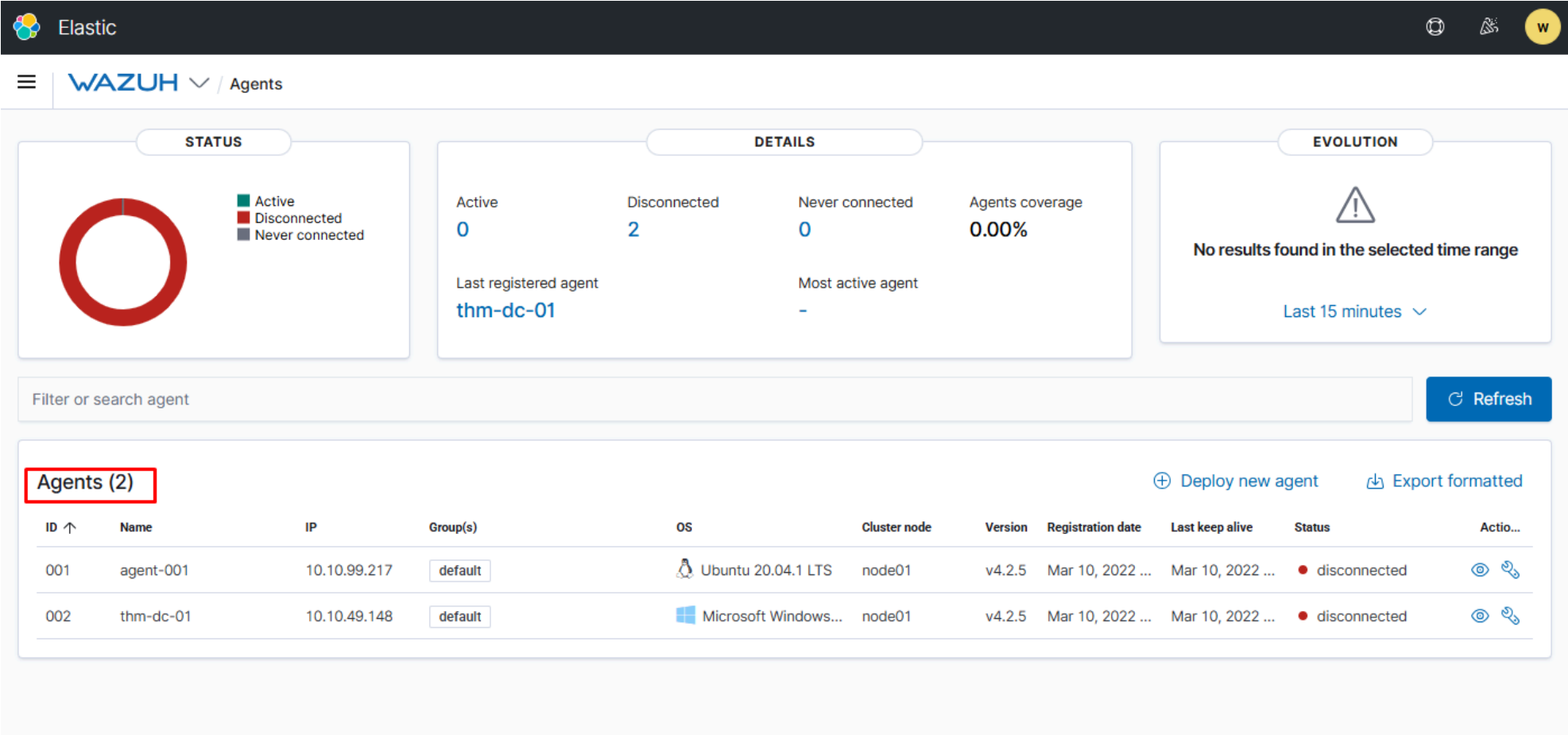
Completo

Vaya a la pestaña "Agentes" presionando **Wazuh -> Agentes**



Completo

¿Cuántos agentes administra este servidor de administración de Wazuh?



2

Entregar

¿Cuál es el estado de los agentes administrados por este servidor de administración de Wazuh?

Disconnected

Entregar

Evaluación de vulnerabilidades y eventos de seguridad de Wazuh

El módulo de evaluación de vulnerabilidades de Wazuh es una herramienta poderosa que se puede utilizar para escanear periódicamente el sistema operativo de un agente en busca de aplicaciones instaladas y sus números de versión.

Una vez recopilada esta información, se envía al servidor Wazuh y se compara con una base de datos de CVE para detectar posibles vulnerabilidades. Por ejemplo, el agente de la captura de pantalla a continuación tiene una versión de Vim vulnerable a **CVE -2019-12735**.

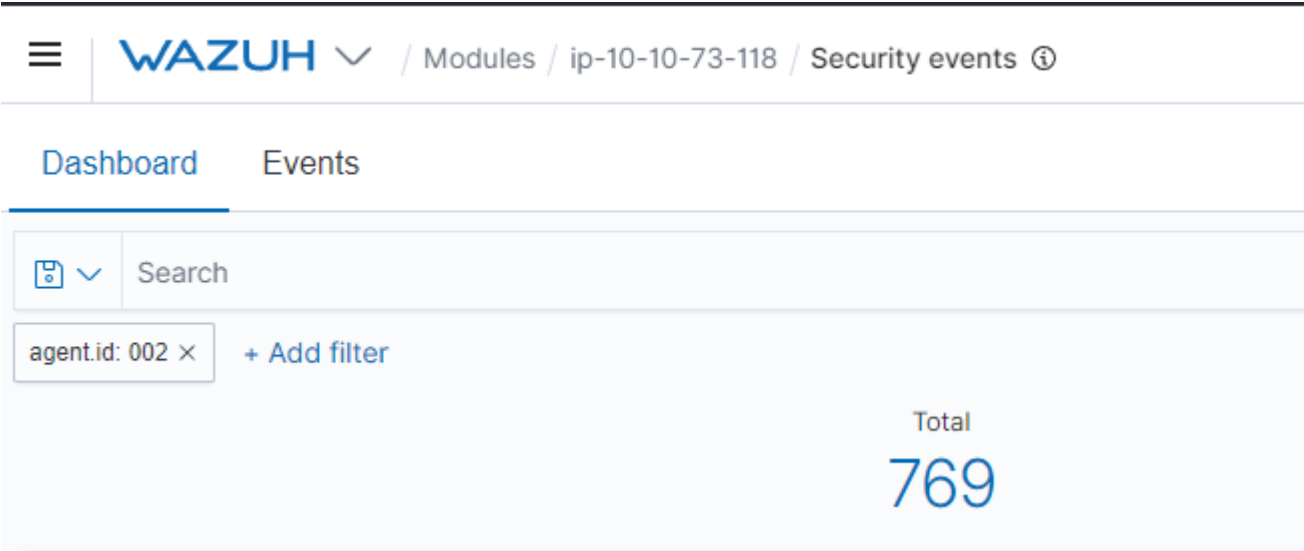
data.vulnerability.package.architecture	amd64
data.vulnerability.package.condition	Package less than 2:8.0.0197-4+deb9u2
data.vulnerability.package.name	vim
data.vulnerability.package.version	2:8.0.0197-4+deb9u1
data.vulnerability.published	Jun 5, 2019 @ 02:00:00.000
data.vulnerability.rationale	getchar.c in Vim before 8.1.1365 and Neovim before 0.3.6 allows remote attackers to execute arbitrary OS commands via the :source! command in a modeline, as demonstrated by execute in Vim, and assert_fails or nvim_input in Neovim.
data.vulnerability.references	> http://lists.opensuse.org/opensuse-security-announce/2019-06/msg00031.html, http://lists.opensuse.org/opensuse-security-announce/2019-06/msg00036.html, http://lists.opensuse.org/opensuse-security-announce/2019-06/msg00037.html, http://lists.opensuse.org/opensuse-security-announce/2019-07/msg00034.html, http://lists.opensuse.org/opensuse-security-announce/2019-07/msg00050.html, http://lists.opensuse.org/opensuse-security-announce/2019-08/msg00075.html http://www.securityfocus.com/bid/108724 http
data.vulnerability.severity	High
data.vulnerability.title	CVE-2019-12735
data.vulnerability.updated	Jun 13, 2019 @ 02:00:00.000

El módulo del escáner de vulnerabilidades realizará un análisis completo cuando el agente Wazuh se instale por primera vez en un dispositivo y **debe** configurarse para ejecutarse en un intervalo establecido y luego (de manera predeterminada, esto se establece en intervalos de 5 minutos cuando está habilitado) de la siguiente manera:

```
<vulnerability-detector>
<enabled>no</enabled>
<interval>5m</interval>
<ignore_time>6h</ignore_time>
<run_on_start>yes</run_on_start>
```

Configuración del servidor de administración de Wazuh para auditar agentes en busca de vulnerabilidades con frecuencia (/var/ossec/etc/ossec.conf)

Wazuh puede probar la configuración de un agente con ciertos conjuntos de reglas para verificar su cumplimiento. Sin embargo, de fábrica, es indudablemente sensible. Tomemos, por ejemplo, este host Linux que ejecuta el agente Wazuh . Se han producido un total de 769 eventos que el sistema realiza como parte de su mantenimiento diario.



Estas acciones frecuentes, como la eliminación de archivos, suelen detectarse como un evento de seguridad. Estos eventos y sus niveles de gravedad están determinados por los conjuntos de reglas de Wazuh , algo que exploraremos en otra tarea.

Podemos analizar estos eventos individualmente seleccionando el menú desplegable. Puedes ordenarlos según diversos factores, como la fecha y hora, las tácticas o la descripción.

WAZUH

Modules / ip-10-10-73-118 / Security events ⓘ

Security Alerts			
Time ↓	Technique(s)	Tactic(s)	Description
> Oct 15, 2021 @ 01:00:51.656			Log file rotated.
> Oct 14, 2021 @ 09:38:46.731	T1107 T1485	Defense Evasion, Impact	File deleted.
> Oct 14, 2021 @ 09:38:46.728	T1107 T1485	Defense Evasion, Impact	File deleted.
> Oct 14, 2021 @ 09:38:45.665			File added to the system.
> Oct 14, 2021 @ 09:38:45.660			File added to the system.
> Oct 14, 2021 @ 01:01:37.867			Log file rotated.
> Oct 13, 2021 @ 17:49:26.879			PAM: Login session closed.
> Oct 13, 2021 @ 16:44:55.276			PAM: Login session closed.
> Oct 13, 2021 @ 16:44:55.274			PAM: Login session closed.

Responda las preguntas a continuación

Asegúrese de haber iniciado sesión en el servidor de administración de Wazuh en **HTTP://10.10.141.81**
listo
Completo

Vaya a la pestaña Agentes presionando **Wazuh -> Agentes** de la siguiente manera
listo
Completo

Seleccione el agente llamado "**AGENT-001** "

Elastic

W

WAZUH

Agents

agent-001

agent-001 Security events Integrity monitoring SCA System Auditing Vulnerabilities More... Inventory data Stats Configuration

ID	Status	IP	Version	Groups	Operating system	Cluster node	Registration date	Last keep alive
001	● disconnected	10.10.99.217	Wazuh v4.2.5	default	Ubuntu 20.04.1 LTS	node01	Mar 10, 2022 @ 20:38:44.000	Mar 10, 2022 @ 20:42:57.000

Last 15 minutes ↓

MITRE

No results

No Mitre results were found in the selected time range.

Compliance

PCI DSS

No results

No PCI DSS results were found in the selected time range.

FIM: Recent events

Time ↓	Path	Action	Rule description	Rule Level	Rule Id
No recent events					

Events count evolution

No results found

SCA: Last scan

CIS Benchmark for Debian/Linux 10 cis_debian10

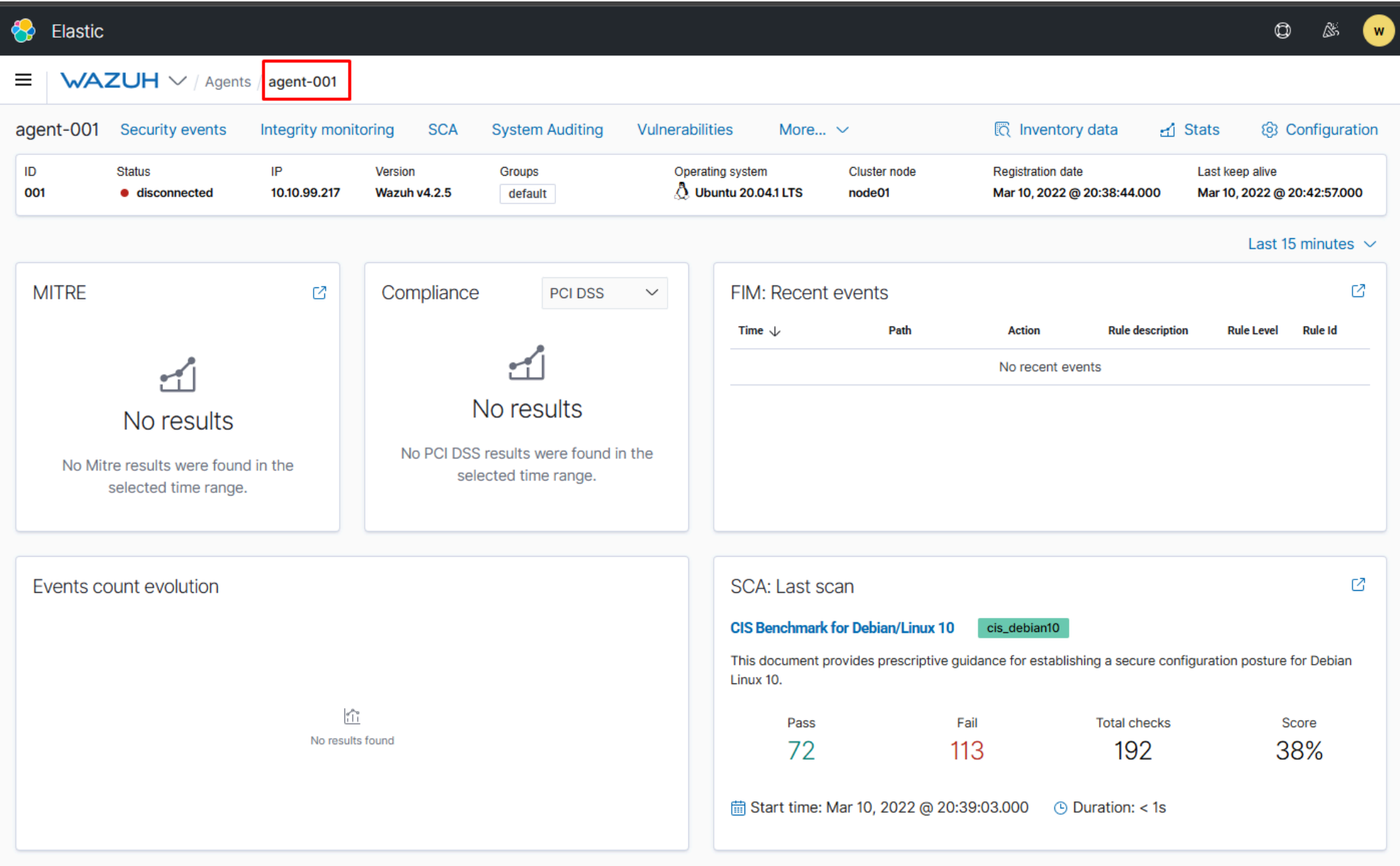
This document provides prescriptive guidance for establishing a secure configuration posture for Debian Linux 10.

Pass	Fail	Total checks	Score
72	113	192	38%

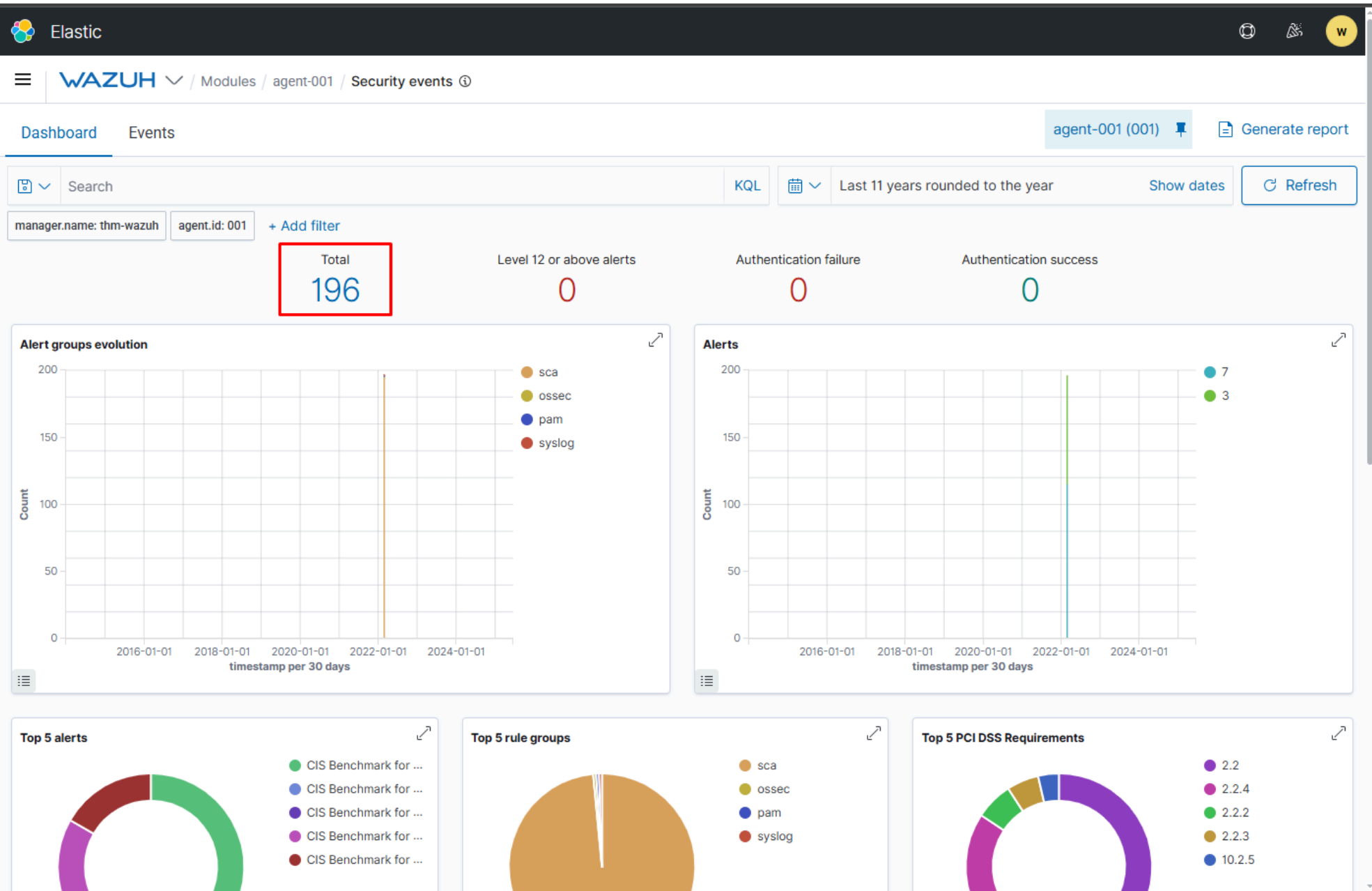
Start time: Mar 10, 2022 @ 20:39:03.000 Duration: < 1s

Completo

¿Cuántas alertas de "Evento de Seguridad" ha generado el agente "AGENT-001"?



Nota : Deberá asegurarse de que su rango de tiempo incluya el 11 de marzo de 2022.



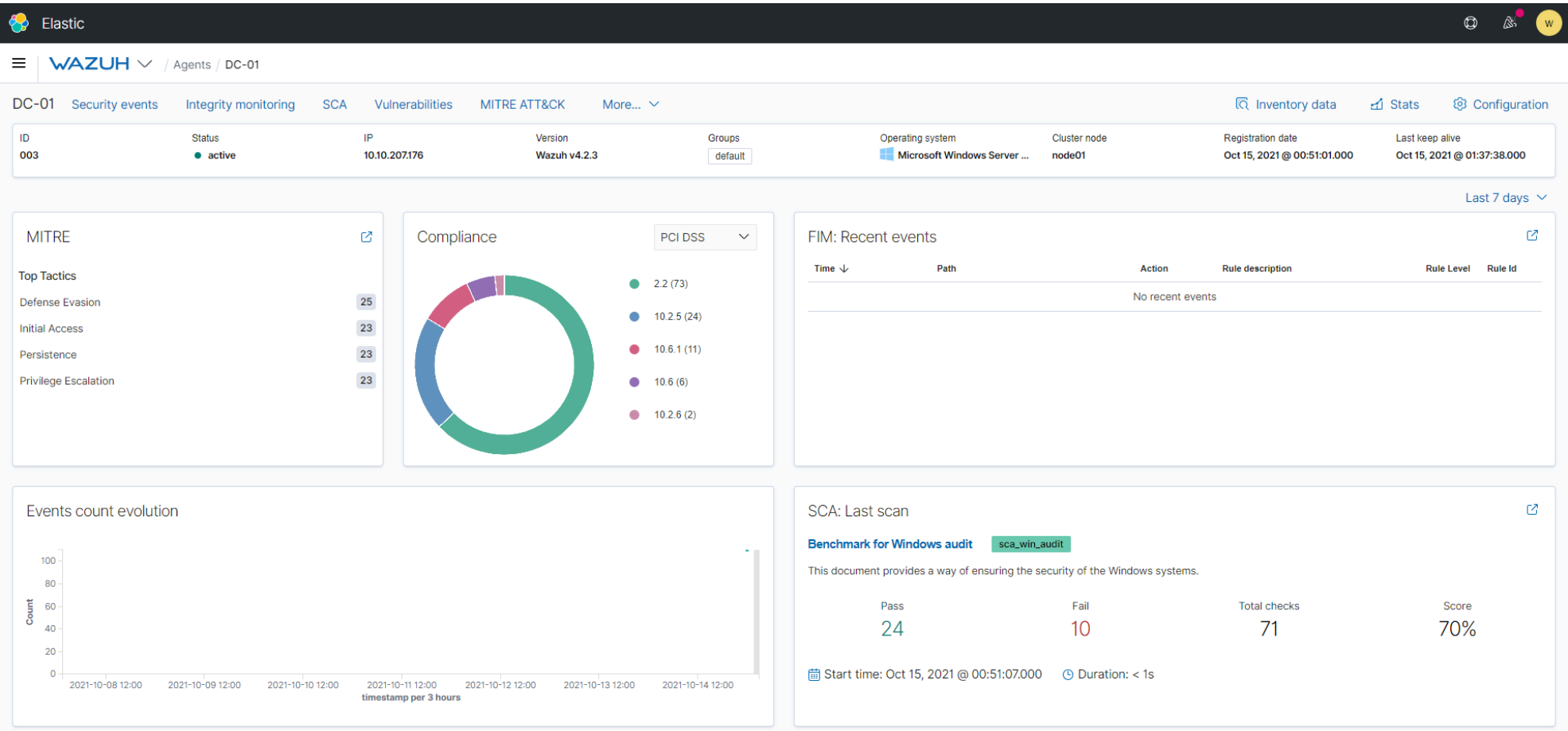
196
Entregar

Auditoría de políticas de Wazuh

Wazuh puede auditar y supervisar la configuración de un agente, a la vez que registra registros de eventos de forma proactiva. Al instalar el agente de Wazuh , se realiza una auditoría donde se genera una métrica utilizando múltiples

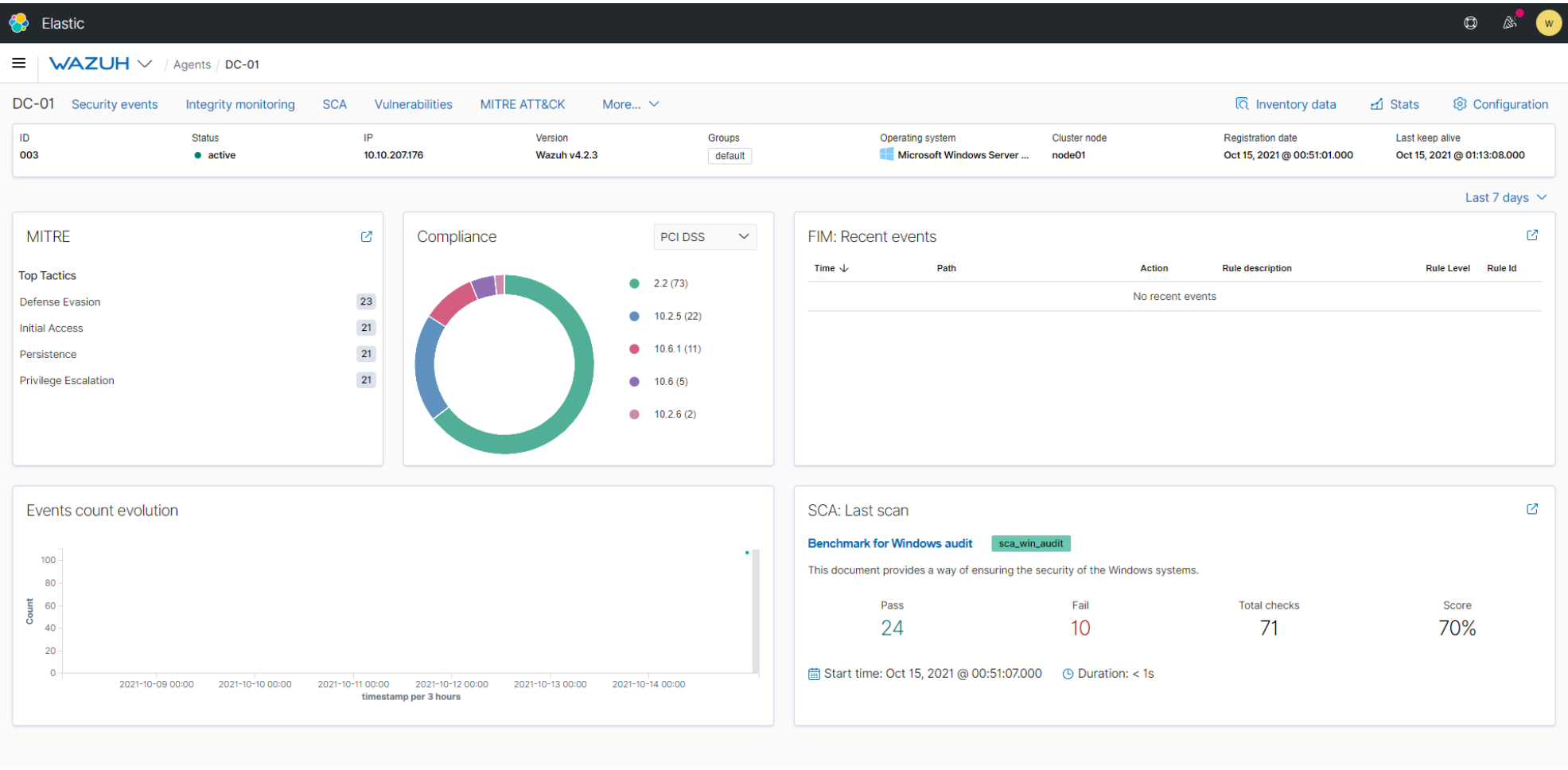
marcos y legislaciones como NIST , MITRE y GDPR.

Por ejemplo, vea cómo este agente DC -01 puntúa frente a MITRE , NIST y SCA :



Estos marcos se describen en la sala Fundamentos de Pentesting si desea obtener más información sobre ellos.

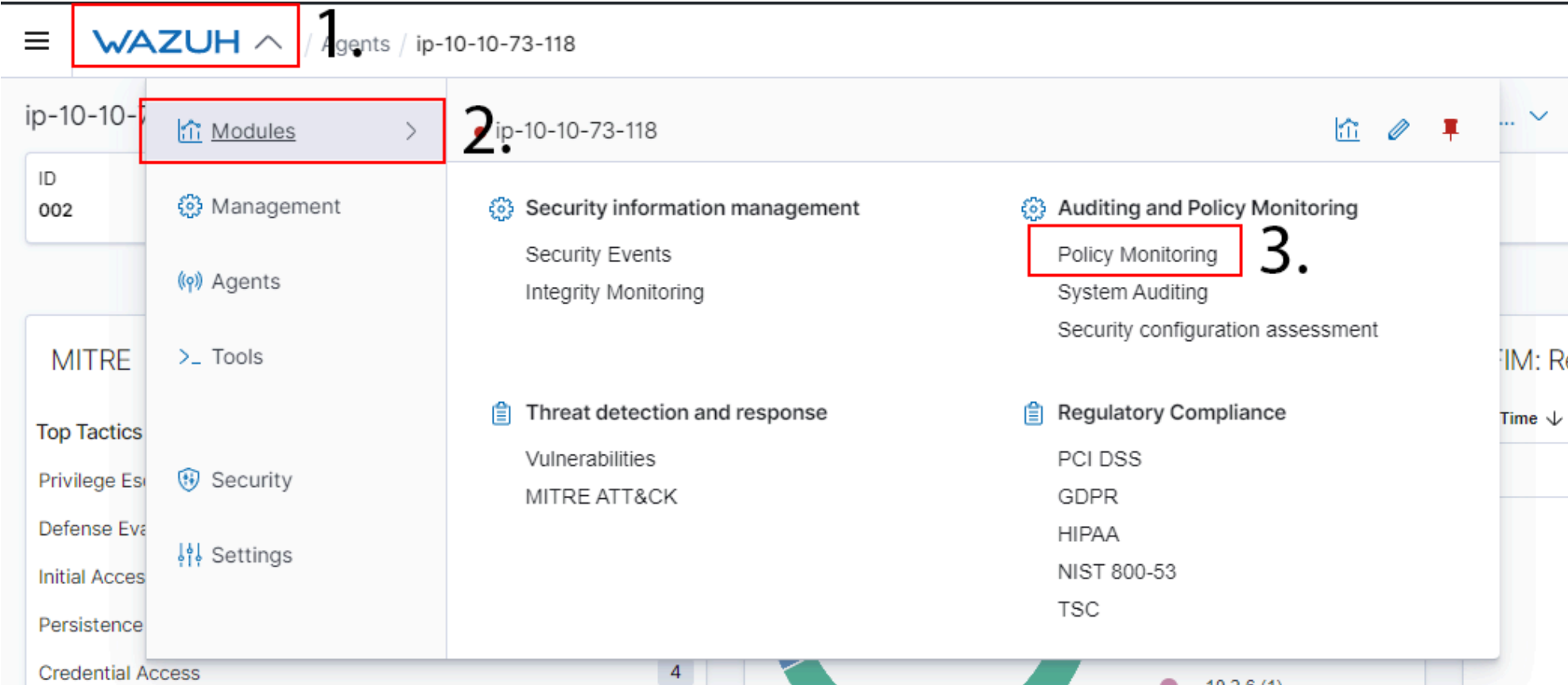
Wazuh presenta una ilustración general de los registros. Podemos usar las visualizaciones para desglosar estos datos y explorarlos más a fondo. Hagamos esto con el mismo agente. Por ejemplo, vea el benchmark de este controlador de dominio ejecutándose en un servidor Windows:



Responda las preguntas a continuación

Asegúrese de haber iniciado sesión en el servidor de administración de Wazuh en 10.10.141.81 ya Completo

Vaya a la pestaña "Módulos" presionando Wazuh -> Módulos y abra el módulo "Administración de políticas" de la siguiente manera:



Completo

Monitoreo de inicios de sesión con Wazuh

El monitor de eventos de seguridad de Wazuh puede registrar activamente tanto los intentos de autenticación exitosos como los fallidos. La regla con el ID 5710 detecta los intentos de conexión fallidos para el protocolo SSH . Veamos la imagen animada a continuación como ejemplo.

Security Alerts							
Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Oct 13, 2021 @ 07:51:40.060	002	ip-10-10-73-118	T1110	Credential Access	sshd: Attempt to login using a non-existent user	5	5710
Table JSON Rule							
agent.ip	10.10.73.118						
agent.name	ip-10-10-73-118						
agent.id	002						
manager.name	ip-10-10-218-190						
rule.mail	false						
rule.level	5						
rule.pci_dss	10.2.4, 10.2.5, 10.6.1						
rule.hipaa	164.312.b						

La alerta se generó porque alguien intentó iniciar sesión en el agente " ip-10-10-73-118 " con el usuario " cmnatic ", que no existe. He resumido esta alerta en la siguiente tabla:

Campo	Valor	Descripción
agente.ip	10.10.73.118	Esta es la dirección IP del agente en el que se activó la alerta.
agente.nombre	ip-10-10-73-118	Este es el nombre de host del agente en el que se activó la alerta.
regla.descripcion	sshd : Intento de iniciar sesión con un usuario inexistente	Este campo es una breve descripción de lo que el evento está alertando.
regla. inglete . técnica	Fuerza bruta	Este campo explica la técnica MITRE a la que pertenece la alerta.
regla.mitre.id	T1110	Este campo es el ID MITRE de la alerta
regla.id	5710	Este campo es el ID asignado a la alerta por el conjunto de reglas de Wazuh
ubicación	/var/log/auth.log	Este campo corresponde a la ubicación del archivo del agente desde el que se generó la alerta. En este ejemplo, se trata del registro de autenticación del agente de Linux .

Como referencia, esta alerta se almacena en un archivo específico del servidor de administración de Wazuh : `/var/ossec/logs/alerts/alerts.log` . Podemos usar un comando como `grep` o `nano` para buscar manualmente en este archivo en el servidor de administración.

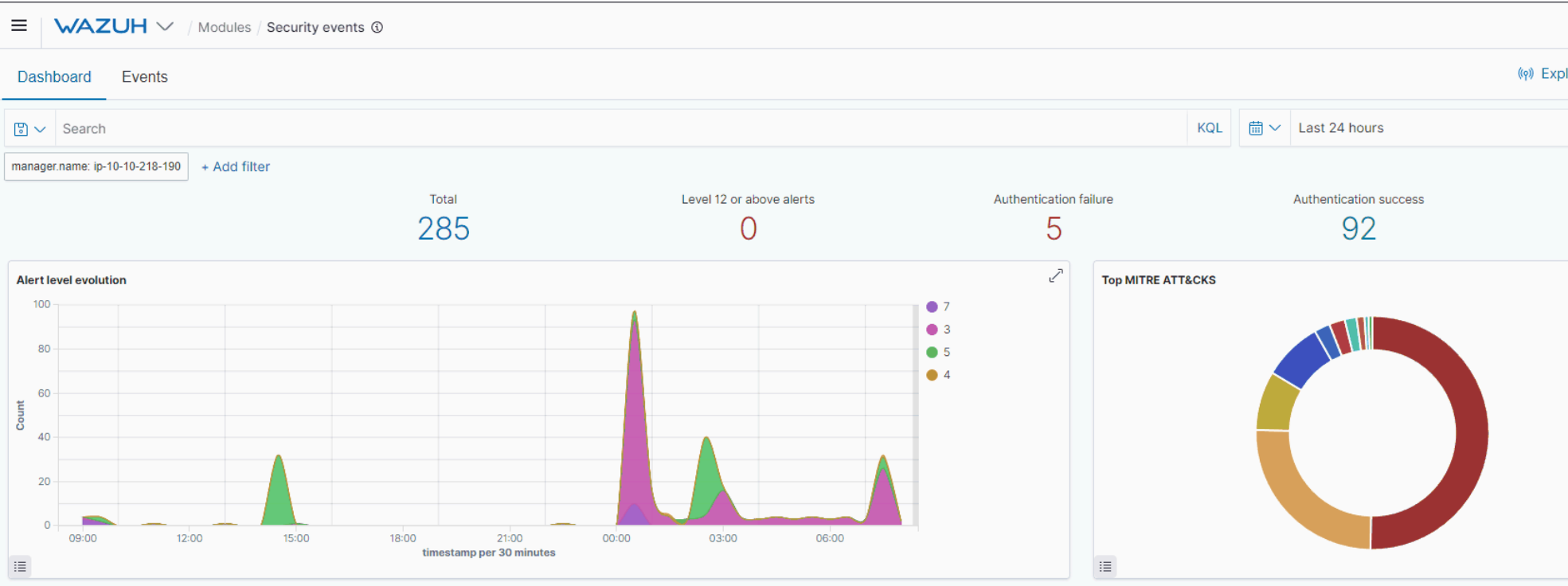
Viendo elWazuhRegistro de alerta de inicio de sesión para una sesión de inicio de sesión (su) en la cuenta raíz por parte del usuario de Ubuntu

```
ubuntu@wazuh-server:~$ sudo less /var/ossec/logs/alerts/alerts.log
** Alert 1634284538.566764: -
pam,syslog,authentication_success,pci_dss_10.2.5,gpg13_7.8,gpg13_7.9,gdpr_IV_32.2,hipaa_164.312.b,ni>
2021 Oct 15 07:55:38 ip-10-10-218-190->/var/log/auth.log
Rule: 5501 (level 3) -> 'PAM: Login session opened.'
User: root
Oct 15 07:55:37 ip-10-10-218-190 sudo: pam_unix(sudo:session): session opened for user root by
ubuntu(uid=0)
uid: 0
```

Al observar el GIF animado a continuación, podemos ver cómo Wazuh ha creado una alerta para un inicio de sesión exitoso en un servidor Windows que ejecuta el agente de Wazuh . Dado que este intento fue exitoso, la gravedad de la alerta se considera menor que la de un inicio de sesión fallido. Esto, por supuesto, se puede adaptar a su entorno. Por ejemplo, si un usuario poco utilizado inicia sesión, puede configurar Wazuh para que muestre esta alerta con mayor gravedad.

Security Alerts							
Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Oct 13, 2021 @ 22:58:37.401	001	DC-01	T1078	Defense Evasion, Initial Access, Persistence, Privilege Escalation	Windows Logon Success	3	60106
Table JSON Rule							
agent.ip		10.10.20.225					
agent.name		DC-01					
agent.id		001					
manager.name		ip-10-10-218-190					
rule.mail		false					
rule.level		3					
rule.pci_dss		10.2.5					
rule.hipaa		164.312.b					

El gif animado a continuación muestra la cantidad de eventos/alertas del agente de Windows activados para indicar cuántas veces un usuario ha iniciado sesión. En este caso, reduce el total de eventos de inicio de sesión de **285** a **79** .

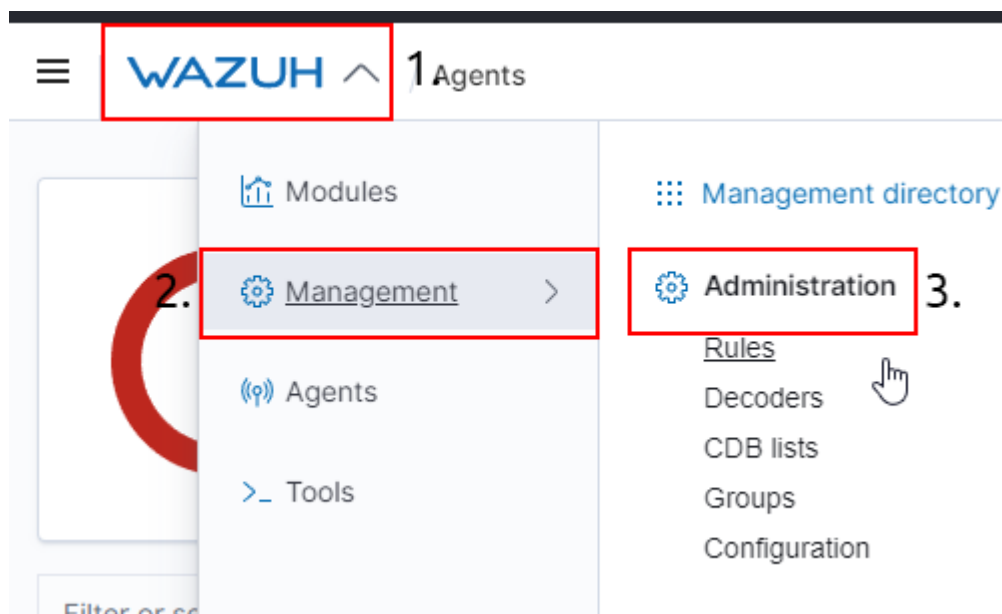


Responda las preguntas a continuación

Asegúrese de haber iniciado sesión en el servidor de administración de Wazuh en [HTTP://10.10.141.81](http://10.10.141.81)

Completo

Vaya a la pestaña "Administración" presionando Wazuh -> **Administración** y abra el módulo "Reglas" de la siguiente manera:



Completo

Recopilación de registros de Windows con Wazuh

En un sistema operativo Windows se capturan y registran todo tipo de acciones y eventos. Esto incluye intentos de autenticación, conexiones de red, archivos a los que se accedió y el comportamiento de aplicaciones y servicios. Esta información se almacena en el registro de eventos de Windows mediante una herramienta llamada Sysmon .

Podemos usar el agente de Wazuh para agregar estos eventos registrados por *Sysmon* y procesarlos en el administrador de Wazuh . Ahora, debemos configurar tanto el agente de Wazuh como la aplicación Sysmon . Sysmon utiliza reglas en formato XML para su activación. Por ejemplo, en el siguiente fragmento de XML , le indicamos a Sysmon que monitoree el evento de inicio del proceso .exe de PowerShell .

ASysmonarchivo de configuración para monitorear elPowerShellproceso

```
Sysmon schemaversion="3.30"
  HashAlgorithms md5 /HashAlgorithms
  EventFiltering
    !--SYSMON EVENT ID 1 : PROCESS CREATION--
    ProcessCreate onmatch="include"
    Image condition="contains" powershell.exe /Image
    /ProcessCreate
    !--SYSMON EVENT ID 2 : FILE CREATION TIME RETROACTIVELY CHANGED IN THE FILESYSTEM--
    FileCreateTime onmatch="include" /FileCreateTime
    !--SYSMON EVENT ID 3 : NETWORK CONNECTION INITIATED--
    NetworkConnect onmatch="include" /NetworkConnect
    !--SYSMON EVENT ID 4 : RESERVED FOR SYSMON STATUS MESSAGES, THIS LINE IS INCLUDED FOR DOCUMENTATION
    PURPOSES ONLY--
    !--SYSMON EVENT ID 5 : PROCESS ENDED--
    ProcessTerminate onmatch="include" /ProcessTerminate
    !--SYSMON EVENT ID 6 : DRIVER LOADED INTO KERNEL--
    DriverLoad onmatch="include" /DriverLoad
    !--SYSMON EVENT ID 7 : DLL (IMAGE) LOADED BY PROCESS--
    ImageLoad onmatch="include" /ImageLoad
    !--SYSMON EVENT ID 8 : REMOTE THREAD CREATED--
    CreateRemoteThread onmatch="include" /CreateRemoteThread
    !--SYSMON EVENT ID 9 : RAW DISK ACCESS--
    RawAccessRead onmatch="include" /RawAccessRead
    !--SYSMON EVENT ID 10 : INTER-PROCESS ACCESS--
    ProcessAccess onmatch="include" /ProcessAccess
    !--SYSMON EVENT ID 11 : FILE CREATED--
    FileCreate onmatch="include" /FileCreate
    !--SYSMON EVENT ID 12 & 13 & 14 : REGISTRY MODIFICATION--
    RegistryEvent onmatch="include" /RegistryEvent
    !--SYSMON EVENT ID 15 : ALTERNATE DATA STREAM CREATED--
    FileCreateStreamHash onmatch="include" /FileCreateStreamHash
    PipeEvent onmatch="include" /PipeEvent
  /EventFiltering
/Sysmon
```

Para indicarle a Sysmon que haga algo, debemos ejecutar la aplicación Sysmon y proporcionar el archivo de configuración mencionado anteriormente de la siguiente manera: `Sysmon64.exe -accepteula -i detect_powershell.xml`

Sysmon					
	Name	Date modified	Type	Size	
	detect_powershell.xml	11/21/2021 10:16 ...	XML Document	2 KB	
	Sysmon.exe	10/26/2021 7:08 PM	Application	6,894 KB	
	Sysmon64.exe	10/26/2021 7:08 PM	Application	3,713 KB	

Administrator: Command Prompt

```
C:\Users\Administrator\Desktop\Sysmon>dir
Volume in drive C has no label.
Volume Serial Number is A8A4-C362

Directory of C:\Users\Administrator\Desktop\Sysmon

11/21/2021  10:16 PM    <DIR>          .
11/21/2021  10:16 PM    <DIR>          ..
11/21/2021  10:16 PM                1,640 detect_powershell.xml
10/26/2021  07:08 PM           7,058,808 Sysmon.exe
10/26/2021  07:08 PM           3,801,488 Sysmon64.exe
               3 File(s)      10,861,936 bytes
               2 Dir(s)  14,597,455,872 bytes free

C:\Users\Administrator\Desktop\Sysmon>Sysmon64.exe -accepteula -i detect_powershell.xml
```

Administrator: Command Prompt

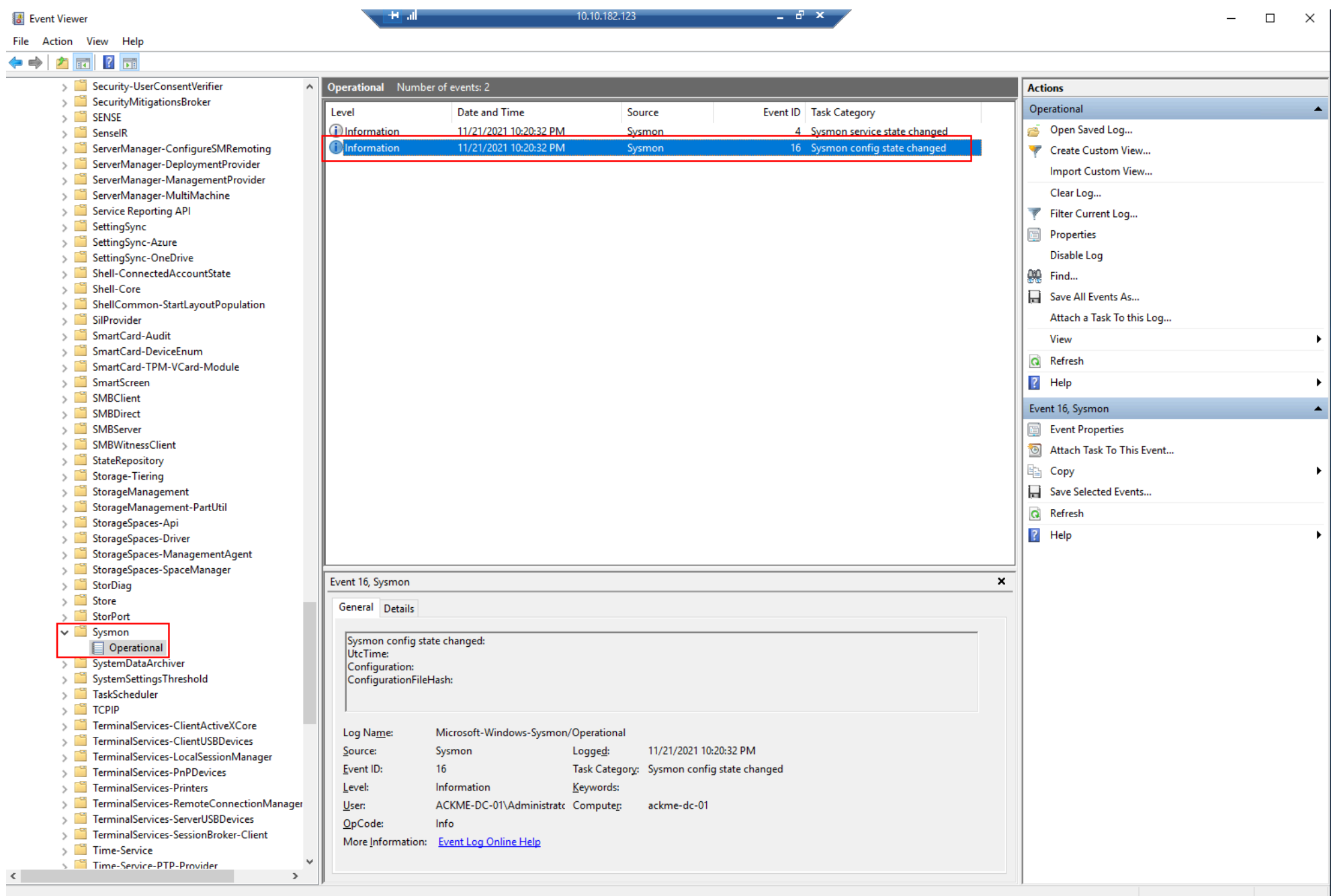
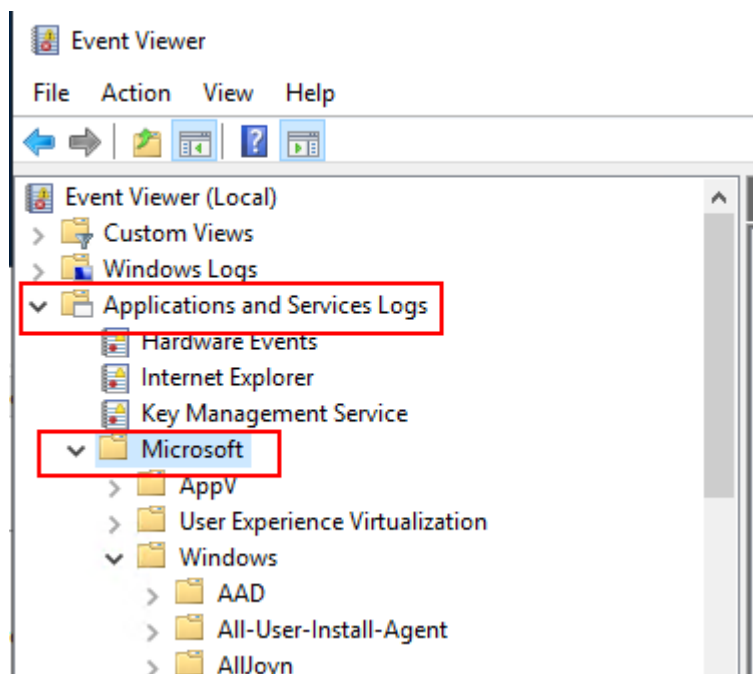
```
C:\Users\Administrator\Desktop\Sysmon>Sysmon64.exe -accepteula -i detect_powershell.xml

System Monitor v13.30 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2021 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

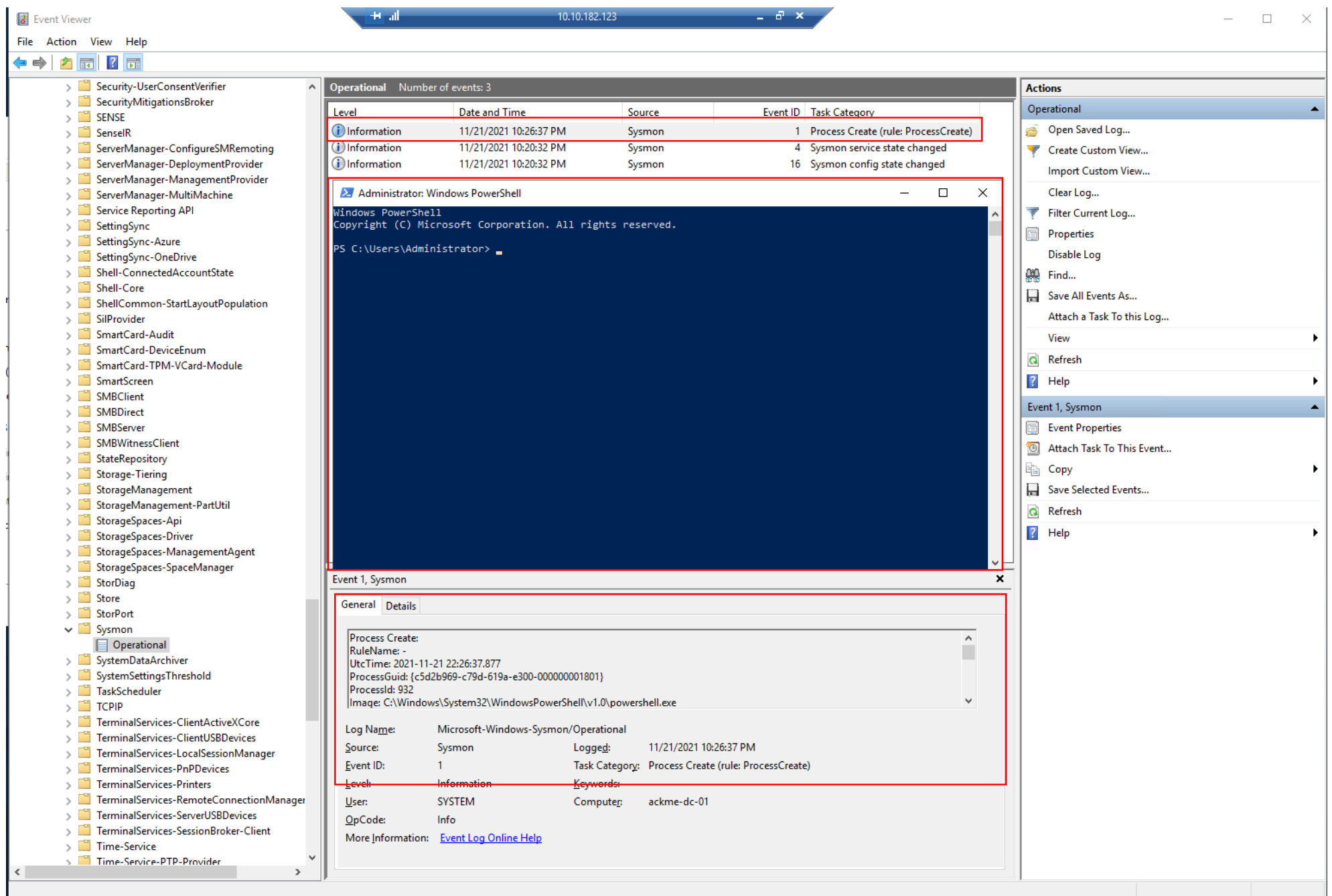
Loading configuration file with schema version 4.81
Configuration file validated.
Sysmon64 installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon64..
Sysmon64 started.

C:\Users\Administrator\Desktop\Sysmon>
```

Podemos verificar que Sysmon ha aceptado nuestro archivo de configuración navegando al Visor de eventos y buscando el módulo “ **Sysmon** ” de la siguiente manera:



Abramos un símbolo del sistema de PowerShell en Windows Server y regresemos al Visor de eventos. Ahora podemos ver un registro de la apertura de este símbolo del sistema , guardado en el Visor de eventos.



Ahora debemos configurar el agente de Wazuh en este servidor de Windows para que envíe estos eventos al servidor de administración de Wazuh . Para ello, debemos abrir el archivo del agente de Wazuh , ubicado en: **C:\Program Files (x86)\ossec-agent\ossec.conf**

```
ossec.conf - Notepad
File Edit Format View Help
<!--
Wazuh - Agent - Default configuration for Windows
More info at: https://documentation.wazuh.com
Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>

  <client>
    <server>
      <address>10.10.235.1</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <crypto_method>aes</crypto_method>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
  </client>

  <!-- Agent buffer options -->
  <client_buffer>
    <disabled>no</disabled>
    <queue_size>5000</queue_size>
    <events_per_second>500</events_per_second>
  </client_buffer>

  <!-- Log analysis -->
  <localfile>
    <location>Application</location>
    <log_format>eventchannel</log_format>
  </localfile>

  <localfile>
    <location>Security</location>
    <log_format>eventchannel</log_format>
    <query>Event/System[EventID != 5145 and EventID != 5156 and EventID != 5447 and
```

Para incluir el siguiente fragmento:

Configurando elWazuhConfiguración del agente

```
<localfile>
<location>Microsoft-Windows-Sysmon/Operational</location>
<log_format>eventchannel</log_format>
</localfile>
```

Luciendo así:

```
ossec.conf - Notepad
File Edit Format View Help

</server>
<crypto_method>aes</crypto_method>
<notify_time>10</notify_time>
<time-reconnect>60</time-reconnect>
<auto_restart>yes</auto_restart>
</client>

<!-- Agent buffer options -->
<client_buffer>
  <disabled>no</disabled>
  <queue_size>5000</queue_size>
  <events_per_second>500</events_per_second>
</client_buffer>

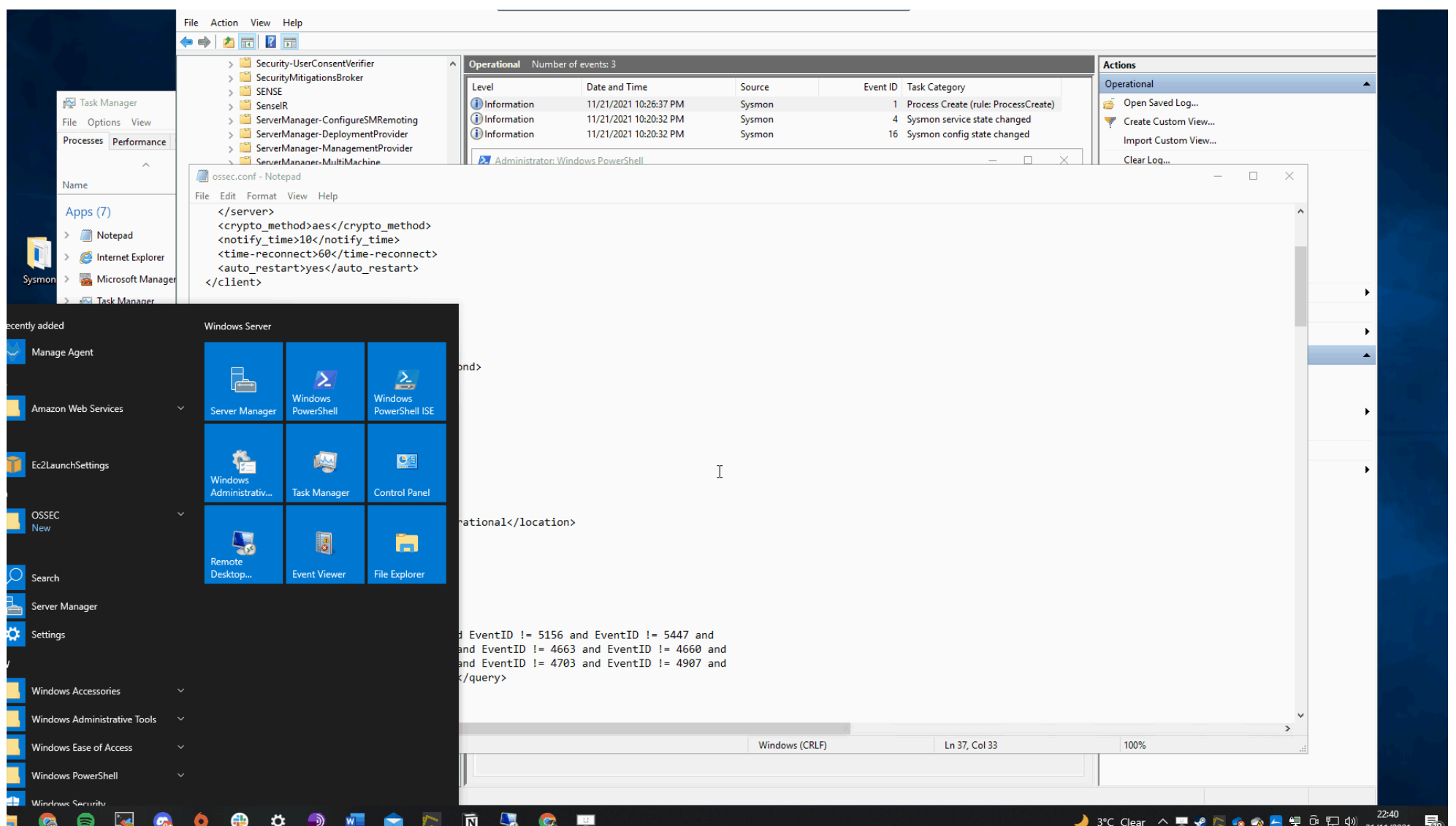
<!-- Log analysis -->
<localfile>
  <location>Application</location>
  <log_format>eventchannel</log_format>
</localfile>

<!-- Sysmon Analysis -->
<localfile>
  <location>Microsoft-Windows-Sysmon/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>

<localfile>
  <location>Security</location>
  <log_format>eventchannel</log_format>
  <query>Event/System[EventID != 5145 and EventID != 5156 and EventID != 4656 and EventID != 4658 and EventID != 4663 and EventID != 4670 and EventID != 4690 and EventID != 4703 and EventID != 5152 and EventID != 5157]</query>
</localfile>

<localfile>
```

Ahora, tendremos que reiniciar el agente de Wazuh . En este caso, estoy reiniciando el sistema operativo solo para asegurarme de que se hayan implementado los cambios.



Una vez hecho esto, debemos indicarle al servidor de administración de Wazuh que agregue Sysmon como regla para visualizar estos eventos. Esto se puede hacer agregando un archivo XML a las reglas locales ubicadas

en `/var/ossec/etc/rules/local_rules.xml`

Configurando elWazuhServidor para ingresar eventos de Ssymon

```
<group name="sysmon,">
  <rule id="255000" level="12">
    <if_group>sysmon_event1</if_group>
    <field name="sysmon.image">\\powershell.exe|\\\.ps1|\\\.ps2</field>
    <description>Sysmon - Event 1: Bad exe: $(sysmon.image)</description>
    <group>sysmon_event1,powershell_execution,</group>
  </rule>
</group>
```

Para que esto se aplique , deberá reiniciar el servidor de administración de Wazuh . Una vez hecho esto, podremos consultar nuestro servidor de administración de Wazuh y comprobar que se han recuperado los datos de un agente.

Responda las preguntas a continuación

¿Cómo se llama la herramienta que podemos utilizar para monitorear eventos del sistema?

Podemos usar el agente de Wazuh para agregar estos eventos registrados por *Sysmon* y procesarlos en el administrador de Wazuh . Ahora, debemos configurar tanto el agente de Wazuh como la aplicación Sysmon . Sysmon utiliza reglas en formato XML para su activación. Por ejemplo, en el siguiente fragmento de XML , le indicamos a Sysmon que monitoree el evento de inicio del proceso .exe de PowerShell .

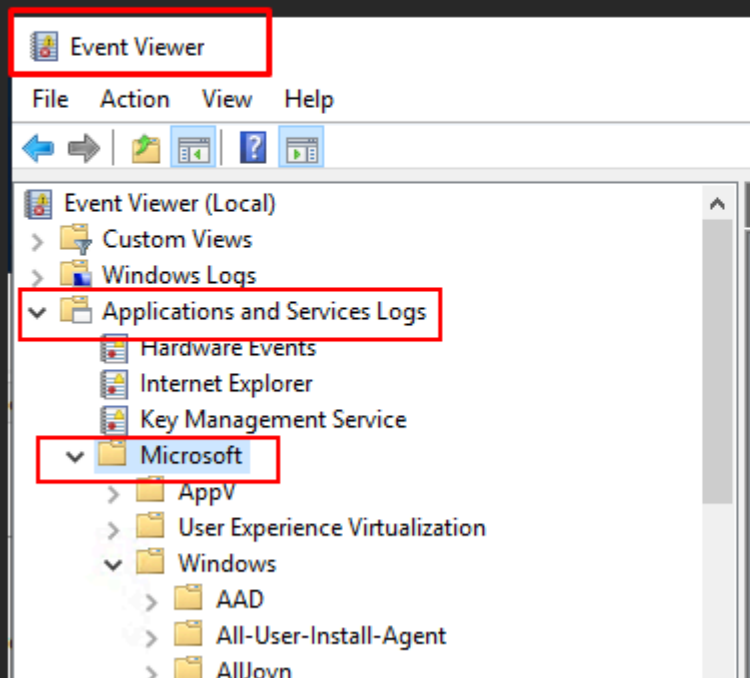
A Sysmon archivo de configuración para monitorear el PowerShell proceso

Sysmon

Entregar

¿En qué aplicación estándar de Windows se graban estos eventos del sistema?

Podemos verificar que Sysmon ha aceptado nuestro archivo de configuración navegando al Visor de eventos y buscando el módulo " Sysmon " de la siguiente manera:



Event Viewer

Entregar

Recopilación de registros de Linux con Wazuh

Capturar registros de un agente Linux es un proceso sencillo, similar a capturar eventos de un agente Windows. Utilizaremos el servicio de recopilación de registros de Wazuh para crear una entrada en el agente que indique qué registros deben enviarse al servidor de administración de Wazuh .

Por ejemplo, en esta tarea, supervisaremos los registros de un servidor web Apache2. Para comenzar, configuraremos el servicio de recopilación de registros en un servidor Linux con el agente Wazuh .

Wazuh incluye numerosas reglas que le permiten analizar archivos de registro y se pueden encontrar en [nombre del archivo] `/var/ossec/ruleset/rules` . Algunas aplicaciones comunes incluyen:

- Estibador
- FTP
- WordPress
- Servidor SQL
- MongoDB
- Cortafuegos
- Y muchos, muchos más (aproximadamente 900).

Sin embargo, siempre puedes crear tus propias reglas. En esta tarea, Wazuh procesará los registros *de Apache2* usando el `0250-apache_rules.xml` conjunto de reglas.

Este conjunto de reglas puede analizar los registros de Apache2 en busca de advertencias y mensajes de error como este: Necesitaremos insertar esto en el agente de Wazuh que envía registros al archivo de configuración de los servidores de administración de Wazuh ubicado en `/var/ossec/etc/ossec.conf`:

Análisis de registros de Apache2

```
<!-- Apache2 Log Analysis -->
<localfile>
  <location>/var/log/example.log</location>
  <log_format>syslog</log_format>
</localfile>
```

Ahora necesitaremos reiniciar el agente Linux que ejecuta el servicio Apache2.

Responda las preguntas a continuación

¿Cuál es la ruta completa del archivo de las reglas ubicadas en un servidor de administración de Wazuh?

Wazuh incluye numerosas reglas que le permiten analizar archivos de registro y se pueden encontrar en [nombre del archivo] `/var/ossec/ruleset/rules`. Algunas aplicaciones comunes incluyen:

`/var/ossec/ruleset/rules`

Entregar

Comandos de auditoría en Linux con Wazuh

Wazuh utiliza el `auditd` paquete que se puede instalar en agentes de Wazuh que se ejecutan en sistemas operativos Debian/Ubuntu y CentOS. En esta tarea, lo usaremos `auditd` en un sistema Ubuntu. `Auditd` Monitorea el sistema para detectar ciertas acciones y eventos y los registra en un archivo de registro.

Luego podemos usar el módulo recopilador de registros en un agente de Wazuh para leer este archivo de registro y enviarlo al servidor de administración de Wazuh para su procesamiento.

Primero, necesitaremos instalar el `auditd` paquete y un `auditd` complemento. Es posible que ya esté instalado en su sistema; sin embargo, instálelo para asegurarnos. Ejecutemos el comando y habilitemos este servicio para que se ejecute tanto en el momento como al arrancar `. sudo apt-get install auditd audispd-plugins`sudo systemctl enable auditd.service`sudo systemctl start auditd.service`

Necesitaremos configurar `auditd` una regla para los comandos y eventos que queremos que monitoree. En esta tarea, le pediremos `auditd` que monitoree cualquier comando ejecutado como root.

Puede ampliar esto para monitorear comandos como `tcpdump`, `netcat`, o archivos *catting* como `/etc/passwd`, que son todos signos distintivos de una infracción.

`Auditd` Las reglas se encuentran en el siguiente directorio: `/etc/audit/rules.d/audit.rules`. Las agregaremos manualmente.

Para esta tarea, necesitaremos abrir el archivo *audit.rules* y añadir nuestra regla. Primero, editemos el archivo usando `sudo nano /etc/audit/rules.d/audit.rules` y añadiendo `-a exit,always -F arch=64 -F euid=0 -S execve -`

k audit-wazuh-c

****Monitoreo de comandos ejecutados como root**

```
## First rule - delete all
-D

## Increase the buffers to survive stress events.
## Make this bigger for busy systems
-b 8192

## This determine how long to wait in burst of events
--backlog_wait_time 0

## Set failure mode to syslog
-f 1

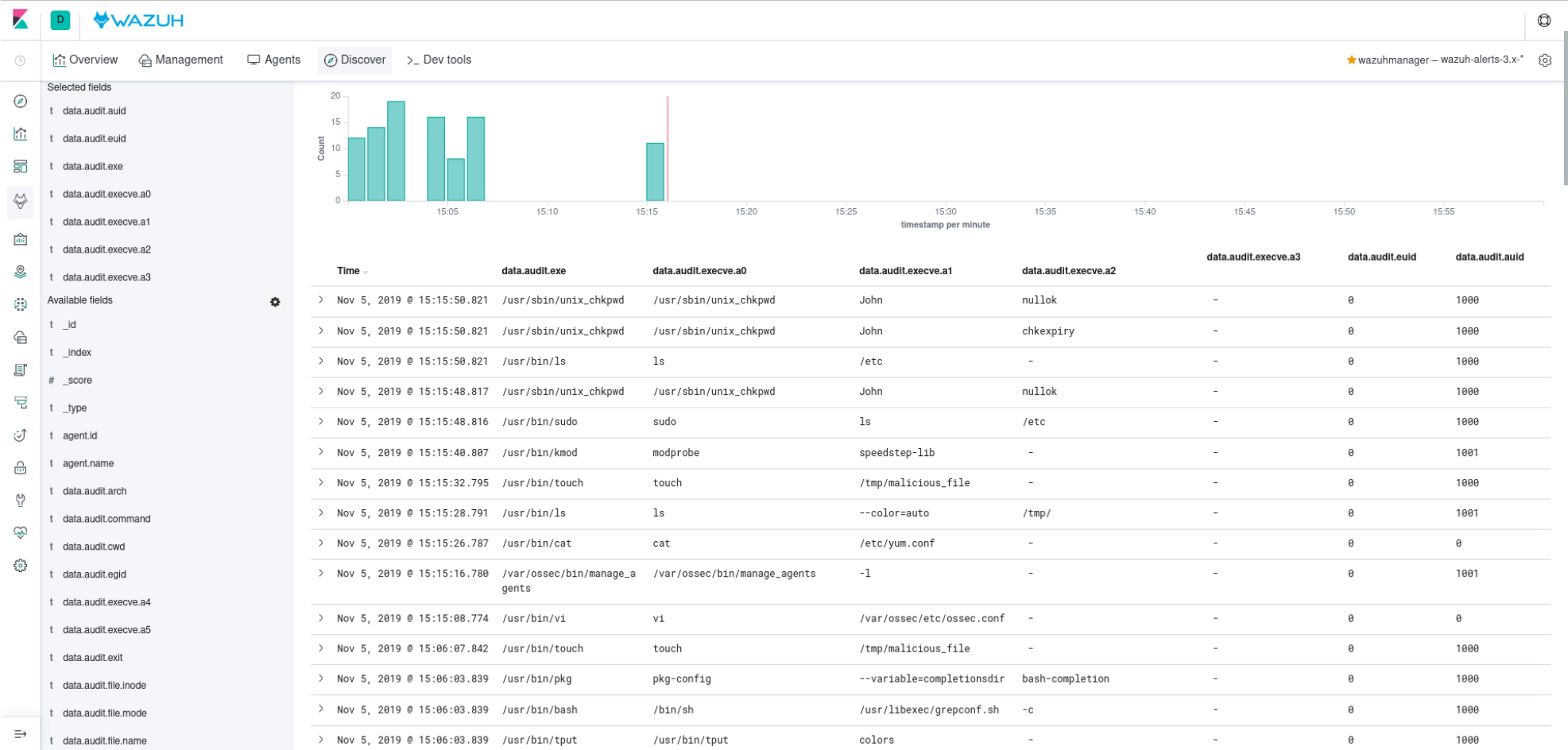
-a exit,always -F arch=b64 -F euid=0 -S execve -k audit-wazuh-c
```

Ahora necesitaremos informar a las auditorías sobre esta nueva regla, así que ejecutemos este comando `sudo auditctl -R /etc/audit/rules.d/audit.rules` para leer el nuevo archivo *audit.rules* que agregamos en la tarea anterior.

Ahora, configuremos el sistema que ejecuta un agente de Wazuh para monitorear estos eventos. En este caso, monitorearemos un host Linux , así que, al igual que en tareas anteriores, necesitaremos configurar el agente de Wazuh `auditd` para que detecte este nuevo archivo de registro generado `sudo nano /var/ossec/etc/ossec.conf` y lo agregue `auditd` de la siguiente manera:

Configurando elWazuhagente para agregar el registro auditado como un archivo de registro para enviar aWazuhservidor de administración

```
<localfile>
  <location>/var/log/audit/audit.log</location>
  <log_format>audit</log_format>
</localfile>
```



Responda las preguntas a continuación

¿Qué aplicación utilizamos en Linux para monitorizar eventos como la ejecución de comandos?

Primero, necesitaremos instalar el `auditd` paquete y un `auditd` complemento. Es posible que ya esté instalado en su sistema; sin embargo, instálelo para asegurarnos. Ejecutemos el comando y habilitemos este servicio para que se ejecute tanto en el momento como al arrancar `.sudo apt-get install auditd audispd-plugins`sudo systemctl enable auditd.service`sudo systemctl start auditd.service`

Necesitaremos configurar `auditd` una regla para los comandos y eventos que queremos que monitoree. En esta tarea, le pediremos `auditd` que monitoree cualquier comando ejecutado como root.

auditd

Entregar

¿Cuál es la ruta completa y el nombre del archivo donde la aplicación mencionada anteriormente almacena las reglas?

Ahora necesitaremos informar a las auditorías sobre esta nueva regla, así que ejecutemos este comando `sudo auditctl -R /etc/audit/rules.d/audit.rules` para leer el nuevo archivo `audit.rules` que agregamos en la tarea anterior.

/etc/audit/rules.d/audit.rules

Entregar

API de Wazuh

Usando nuestro propio cliente

El servidor de administración de Wazuh cuenta con una API completa y completa que permite interactuar con él mediante la línea de comandos. Dado que el servidor de administración de Wazuh requiere autenticación, primero debemos autenticar a nuestro cliente.

En esta tarea, usaremos una máquina Linux con la `curl` herramienta instalada para interactuar con la API del servidor de administración de Wazuh . Primero, necesitaremos autenticarnos proporcionando un conjunto válido de credenciales al punto de autenticación.

Una vez autenticados, el servidor de administración de Wazuh nos proporcionará un token (similar a una sesión) que necesitaremos para cualquier interacción posterior. Podemos almacenar este token como una variable de entorno en nuestra máquina Linux , como se muestra en el siguiente fragmento:

(reemplazando `WAZUH_MANAGEMENT_SERVER_IP` con la dirección IP del servidor de administración de Wazuh (es decir, 10.10.141.81):

```
TOKEN=$(curl -u : -k -X GET "https://WAZUH_MANAGEMENT_SERVER_IP:55000/security/user/authenticate?raw=true")
```

Confirmemos que nos hemos autenticado correctamente y que el servidor de administración de Wazuh nos ha proporcionado un token :

```
curl -k -X GET "https://10.10.141.81:55000/" -H "Authorization: Bearer $TOKEN"
```

*WazuhAPIVerificar autenticación

```
{
  "data": {
    "title": "Wazuh API",
    "api_version": "4.0.0",
    "revision": 4000,
    "license_name": "GPL 2.0",
    "license_url": "https://github.com/wazuh/wazuh/blob/master/LICENSE",
    "hostname": "wazuh-master",
    "timestamp": "2021-10-25T07:05:00+0000"
  },
  "error": 0
}
```

Podemos utilizar los métodos de solicitud HTTP estándar , por ejemplo, `GET/POST/PUT/DELETE` proporcionando la opción relevante después de un `-X` ie. `-X GET`

```
curl -k -X GET "https://10.10.141.81:55000/manager/status?pretty=true" -H "Authorization: Bearer $TOKEN"
```

Por ejemplo, usemos la API de Wazuh para enumerar algunas estadísticas e información importante sobre el servidor de administración de Wazuh , incluidos qué servicios se están monitoreando y algunas configuraciones generales sobre el servidor de administración de Wazuh :

```
curl -k -X GET "https://10.10.141.81:55000/manager/configuration?pretty=true&section=global" -H "Authorization: Bearer $TOKEN"
```

*Obtener información sobre laWazuhgerente

```
{
  "data": {
    "affected_items": [
      {
        "wazuh-agentlessd": "running",
        "wazuh-analysisd": "running",
        "wazuh-authd": "running",
        "wazuh-csyslogd": "running",
        "wazuh-dbd": "stopped",
        "wazuh-monitord": "running",
        "wazuh-execd": "running",
        "wazuh-integrator": "running",
        "wazuh-logcollector": "running",
        "wazuh-maild": "running",
        "wazuh-remoted": "running",
        "wazuh-reportd": "stopped",
        "wazuh-syscheckd": "running",
        "wazuh-clusterd": "running",
        "wazuh-modulesd": "running",
        "wazuh-db": "running",
        "wazuh-api": "stopped"
      }
    ],
    "total_affected_items": 1,
    "total_failed_items": 0,
    "failed_items": []
  },
  "message": "Processes status were successfully read in specified node",
  "error": 0
}
```

O quizás, podemos utilizar la API del servidor de administración de Wazuh para interactuar con un agente:

```
curl -k -X GET "https://10.10.141.81:55000/agents?pretty=true&offset=1&limit=2&select=status%2Cid%2Cmanager%2Cname%2Cnode_name%2Cversion&status=active" -H "Authorization: Bearer $TOKEN"
```

*Usando elWazuhservidores de administraciónAPIinteractuar con un agente

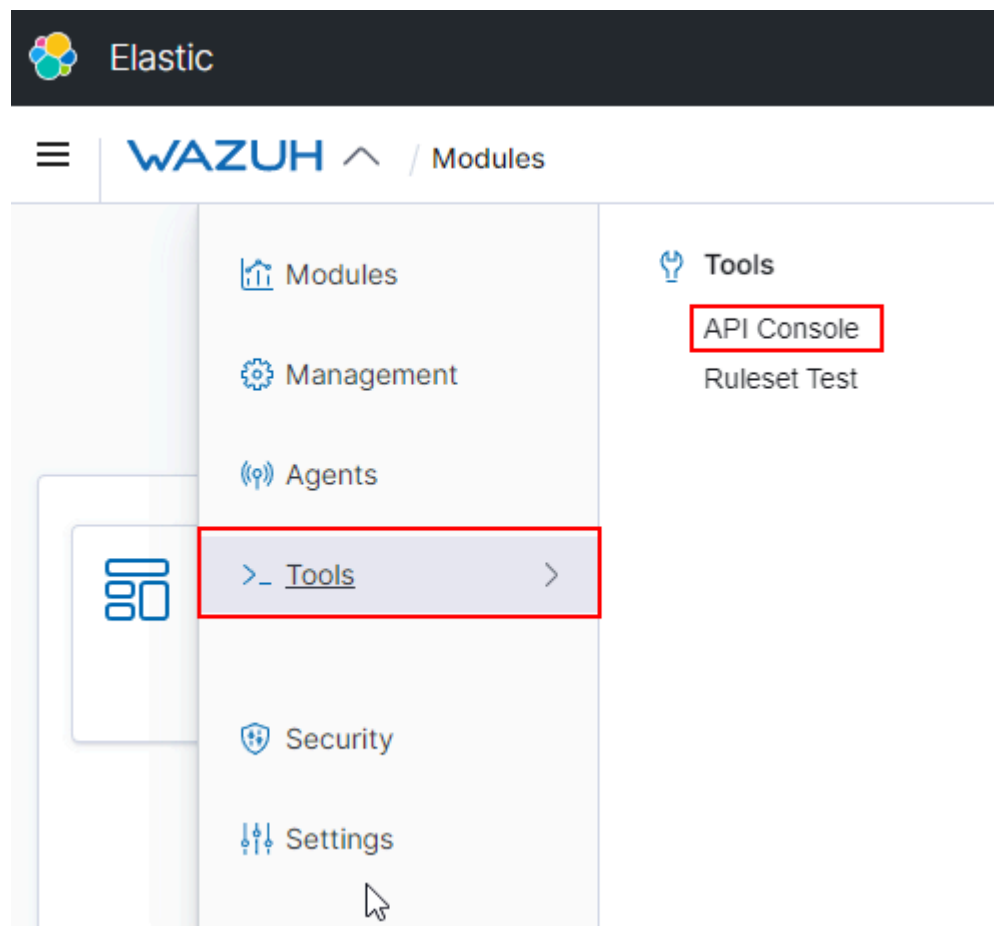
```
{
  "data": {
    "affected_items": [
      {
        "node_name": "worker2",
        "status": "active",
        "manager": "wazuh-worker2",
        "version": "Wazuh v3.13.1",
        "id": "001",
        "name": "wazuh-agent1"
      }
    ],
    "total_affected_items": 9,
    "total_failed_items": 0,
    "failed_items": []
  }
}
```

```
} ,
"message": "All selected agents information was returned",
"error": 0
}
```

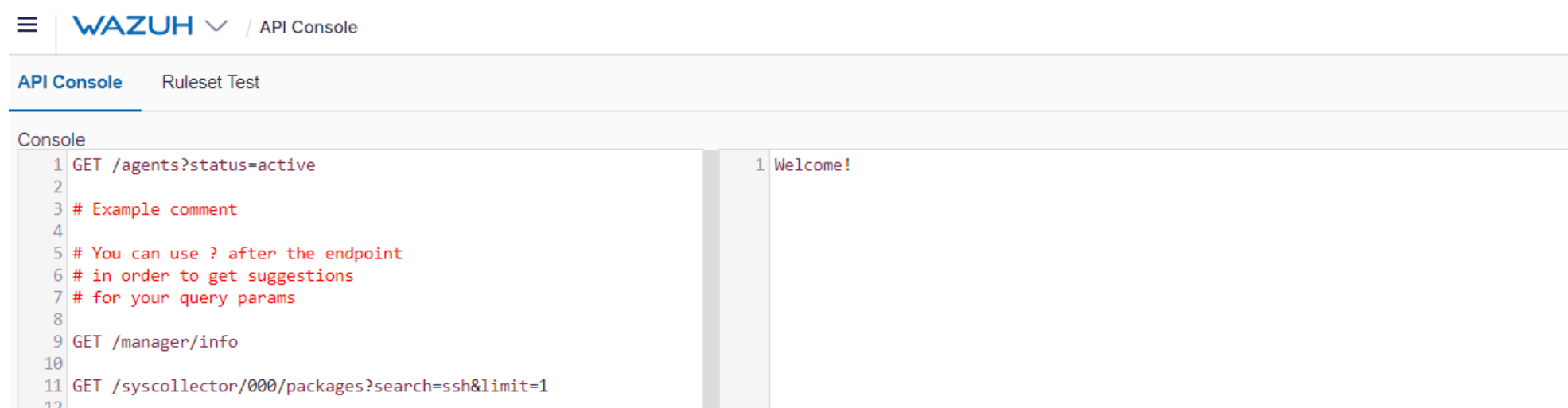
Uso de la consola API de Wazuh

Wazuh cuenta con una potente consola API integrada en su sitio web para consultar servidores y agentes de administración. Si bien no es tan completa como usar tu propio entorno (donde puedes crear y ejecutar scripts con Python, por ejemplo), resulta práctica.

Para encontrar esta consola API , necesitamos abrir la categoría "Herramientas" dentro del encabezado Wazuh en la parte superior:



Se le mostrarán algunas consultas de ejemplo que puede ejecutar. Simplemente *seleccione* la línea y *presione* la flecha verde de ejecución para ejecutar la consulta, como se muestra a continuación:



Recordatorio: la sintaxis para ejecutar consultas utiliza los mismos métodos web (p. ej., GET/PUT/POST) y puntos finales (p. ej., /manager/info) que usaría con curl. Puede consultar más opciones sobre los puntos finales de la API consultando la documentación detallada de la API de Wazuh [aquí](#).

Responda las preguntas a continuación

¿Cuál es el nombre de la herramienta estándar de Linux que podemos usar para realizar solicitudes al servidor de administración de Wazuh?

En esta tarea, usaremos una máquina Linux con la **curl** herramienta instalada para interactuar con la API del servidor de administración de Wazuh . Primero, necesitaremos autenticarnos proporcionando un conjunto válido de credenciales al punto de autenticación.

Curl
Entregar

¿Qué método HTTP usaríamos para recuperar información para una API de servidor de administración de Wazuh?

```
Podemos utilizar los métodos de solicitud HTTP estándar , por ejemplo, GET/POST/PUT/DELETE proporcionando la opción relevante después de un -X ie. -X GET

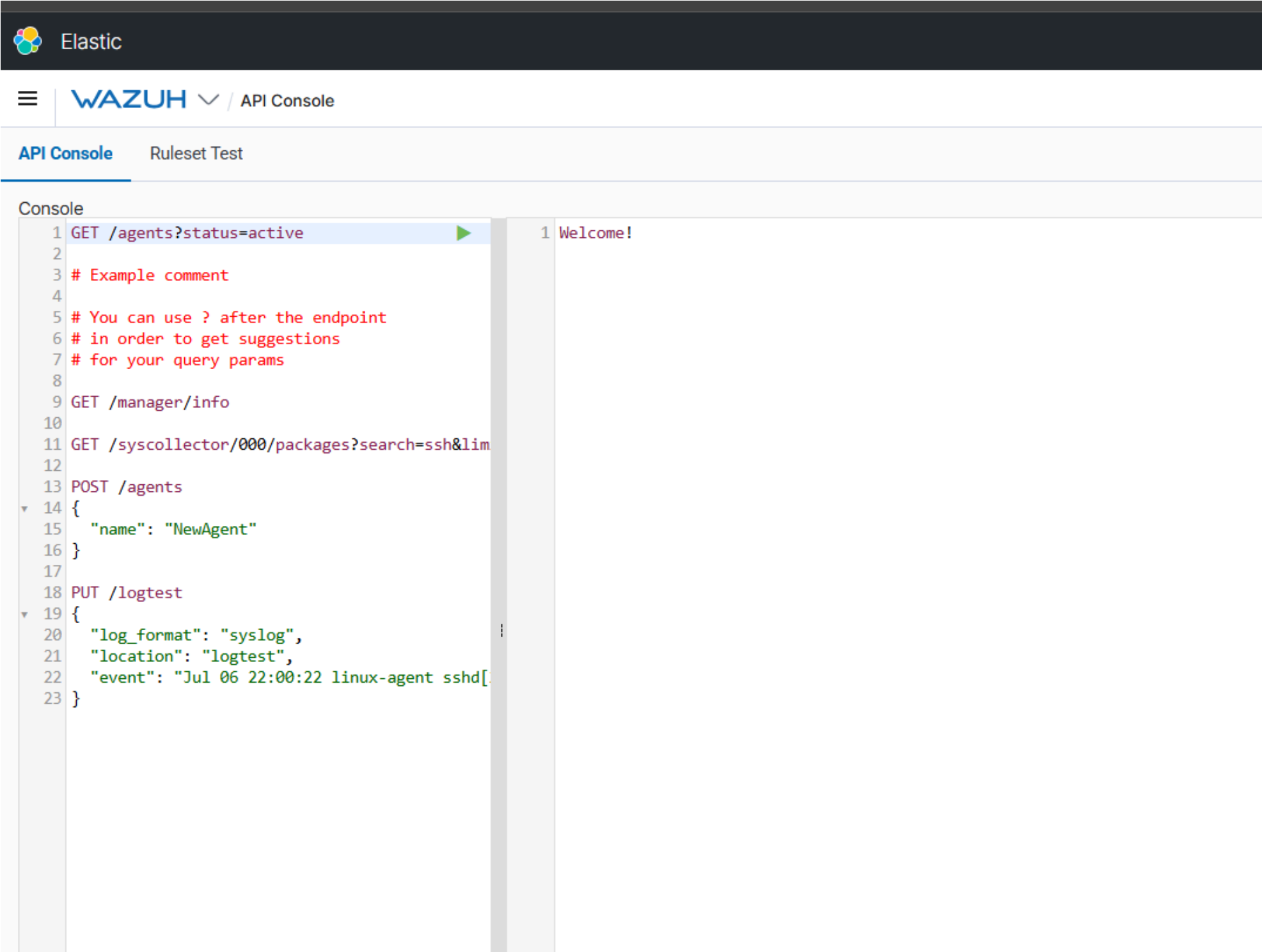
curl -k -X GET "https://10.10.141.81:55000/manager/status?pretty=true" -H "Authorization: Bearer $TOKEN"
```

GET
Entregar

¿Qué método HTTP usaríamos para realizar una acción en una API del servidor de administración de Wazuh?

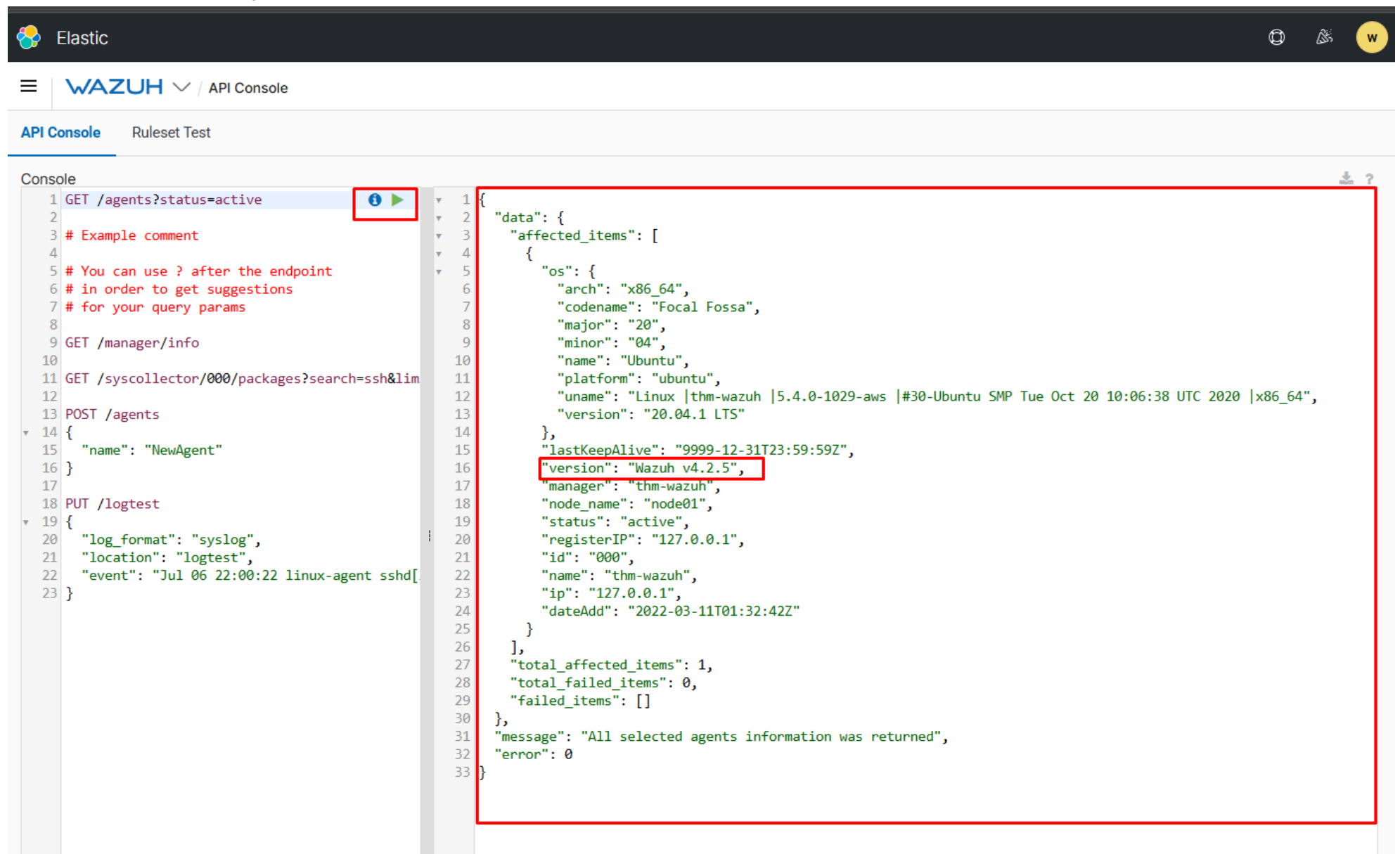
PUT
Entregar

Navegue a la consola API de Wazuh.



CompletoPista

Utilice la consola API para encontrar la versión del servidor Wazuh.



The screenshot shows the Wazuh API Console interface. On the left, the 'Console' tab is active, displaying a list of API requests. The first request, 'GET /agents?status=active', is highlighted. On the right, the JSON response for this request is shown. A red box highlights the 'version' field in the response, which is 'Wazuh v4.2.5'.

```
1 GET /agents?status=active
2
3 # Example comment
4
5 # You can use ? after the endpoint
6 # in order to get suggestions
7 # for your query params
8
9 GET /manager/info
10
11 GET /syscollector/000/packages?search=ssh&lim
12
13 POST /agents
14 {
15   "name": "NewAgent"
16 }
17
18 PUT /logtest
19 {
20   "log_format": "syslog",
21   "location": "logtest",
22   "event": "Jul 06 22:00:22 linux-agent sshd[
23 }
```

```
1 {
2   "data": {
3     "affected_items": [
4       {
5         "os": {
6           "arch": "x86_64",
7           "codename": "Focal Fossa",
8           "major": "20",
9           "minor": "04",
10          "name": "Ubuntu",
11          "platform": "ubuntu",
12          "uname": "Linux |thm-wazuh |5.4.0-1029-aws |#30-Ubuntu SMP Tue Oct 20 10:06:38 UTC 2020 |x86_64",
13          "version": "20.04.1 LTS"
14        },
15        "lastKeepAlive": "9999-12-31T23:59:59Z",
16        "version": "Wazuh v4.2.5",
17        "manager": "thm-wazuh",
18        "node_name": "node01",
19        "status": "active",
20        "registerIP": "127.0.0.1",
21        "id": "000",
22        "name": "thm-wazuh",
23        "ip": "127.0.0.1",
24        "dateAdd": "2022-03-11T01:32:42Z"
25      }
26    ],
27    "total_affected_items": 1,
28    "total_failed_items": 0,
29    "failed_items": []
30  },
31  "message": "All selected agents information was returned",
32  "error": 0
33 }
```

Nota: Deberá agregar el prefijo "v" al número de esta respuesta. Por ejemplo, v 1.2.3

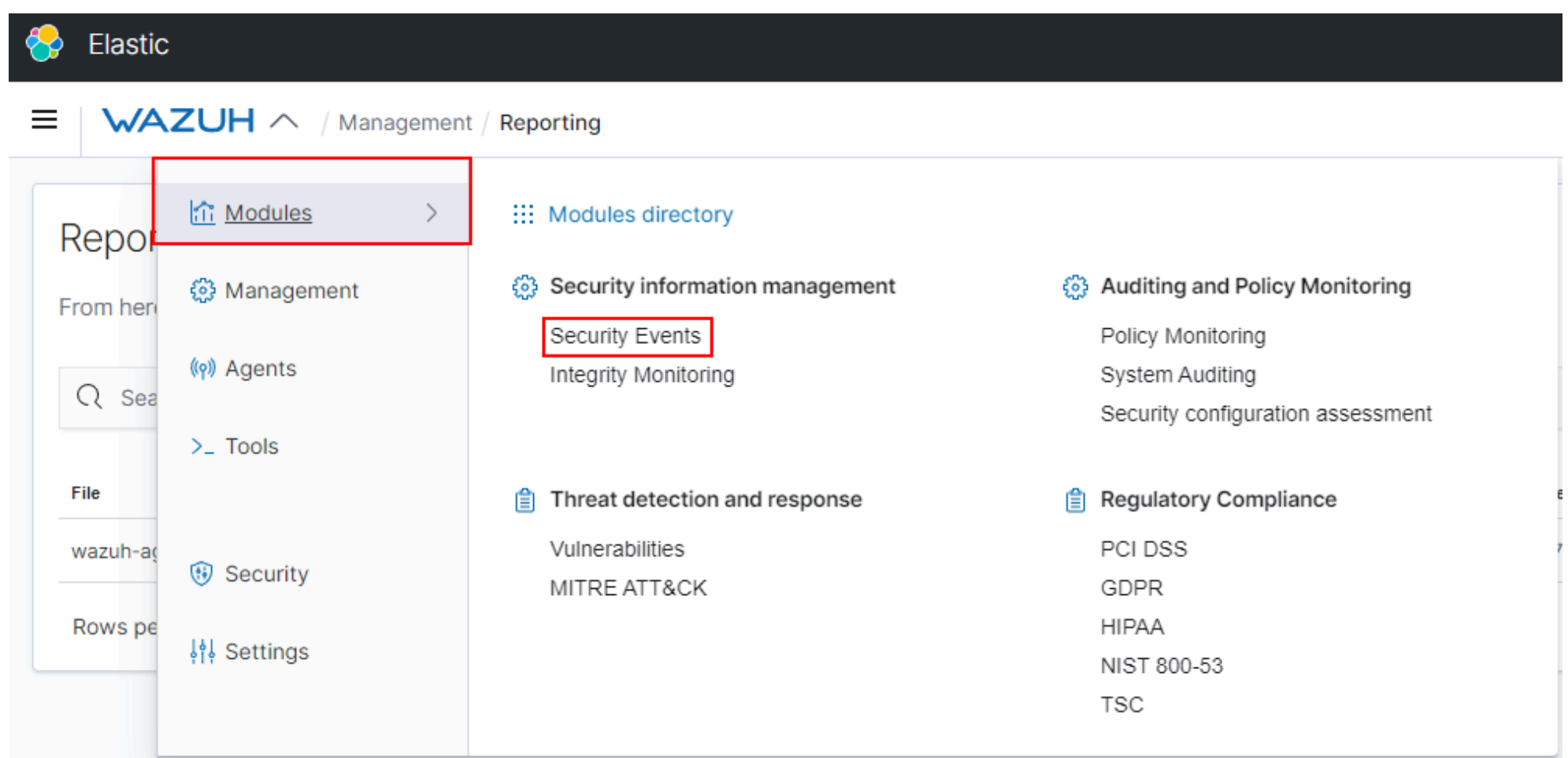
v4.2.5

Entregar

Generando informes con Wazuh

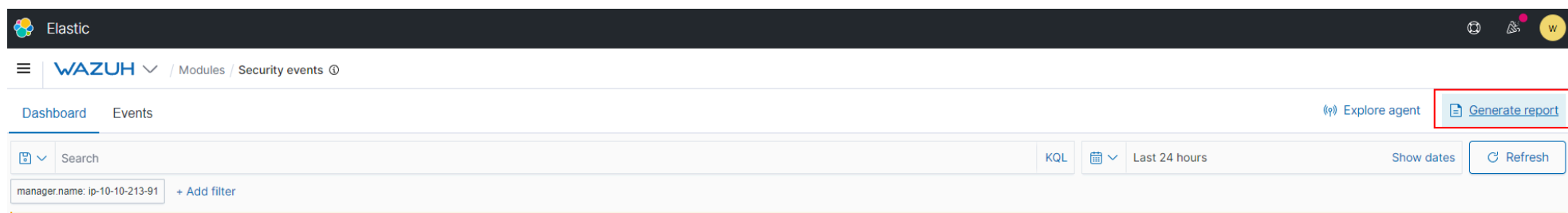
Wazuh cuenta con un módulo de informes que le permite ver un desglose resumido de los eventos que han ocurrido en un agente.

Primero, debemos seleccionar una vista para generar informes. En este ejemplo, quiero generar un informe de los eventos de seguridad de las últimas 24 horas. Para ello, debo abrir la vista: **1. Módulos -> 2. Eventos de seguridad**



The screenshot shows the Wazuh Management Reporting interface. The 'Modules' menu is open, and 'Security Events' is highlighted under the 'Security information management' section. The 'Security Events' section is also highlighted in the main content area.

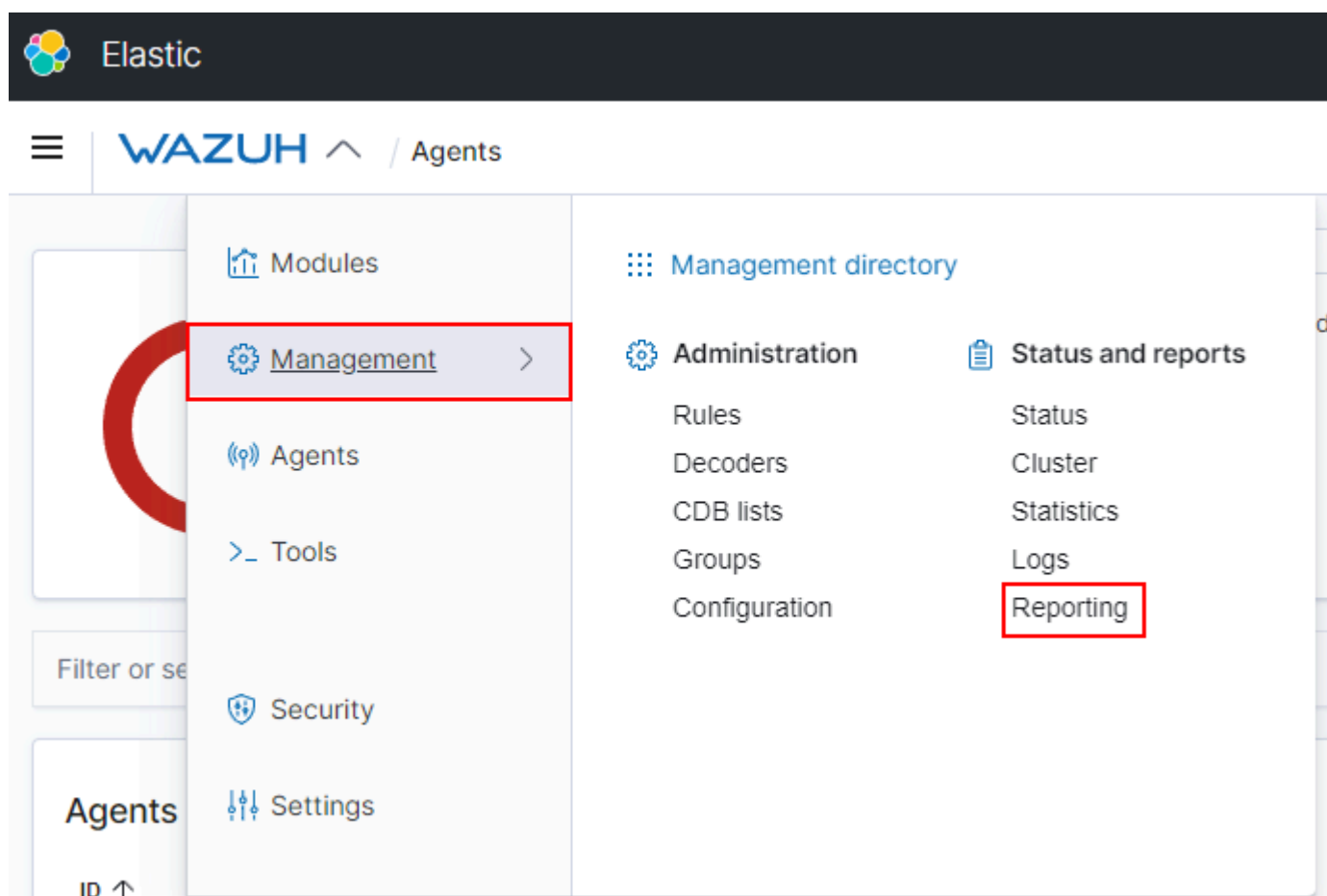
Ahora, si ha habido alertas en las últimas 24 horas, puedo generar un informe como el siguiente:



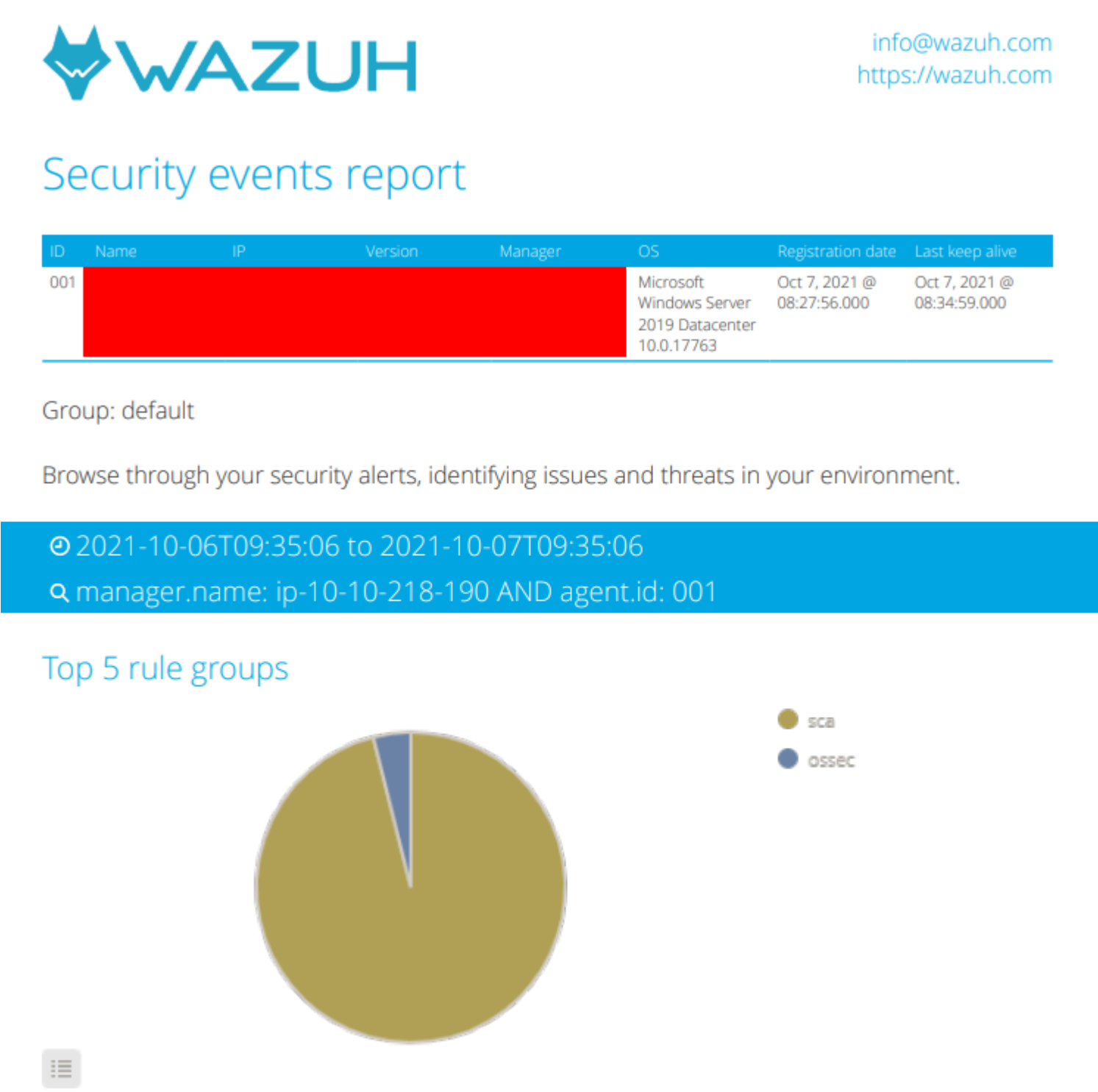
Nota: Si este botón está desactivado, no hay datos del informe, por lo que deberá cambiar su consulta o ampliar el rango de fechas.

El informe **puede** tardar entre un par de segundos y unos minutos en generarse (dependiendo de la cantidad de datos que se deban procesar). Tras un tiempo, accederemos al panel de resumen del informe en Wazuh .

Primero, presione el encabezado “ Wazuh ” en la parte superior de la pantalla y seleccione “ **Administración** ”, y luego haga clic en el texto “ **Informes** ” ubicado debajo del subtítulo “ **Estado e informes** ”:

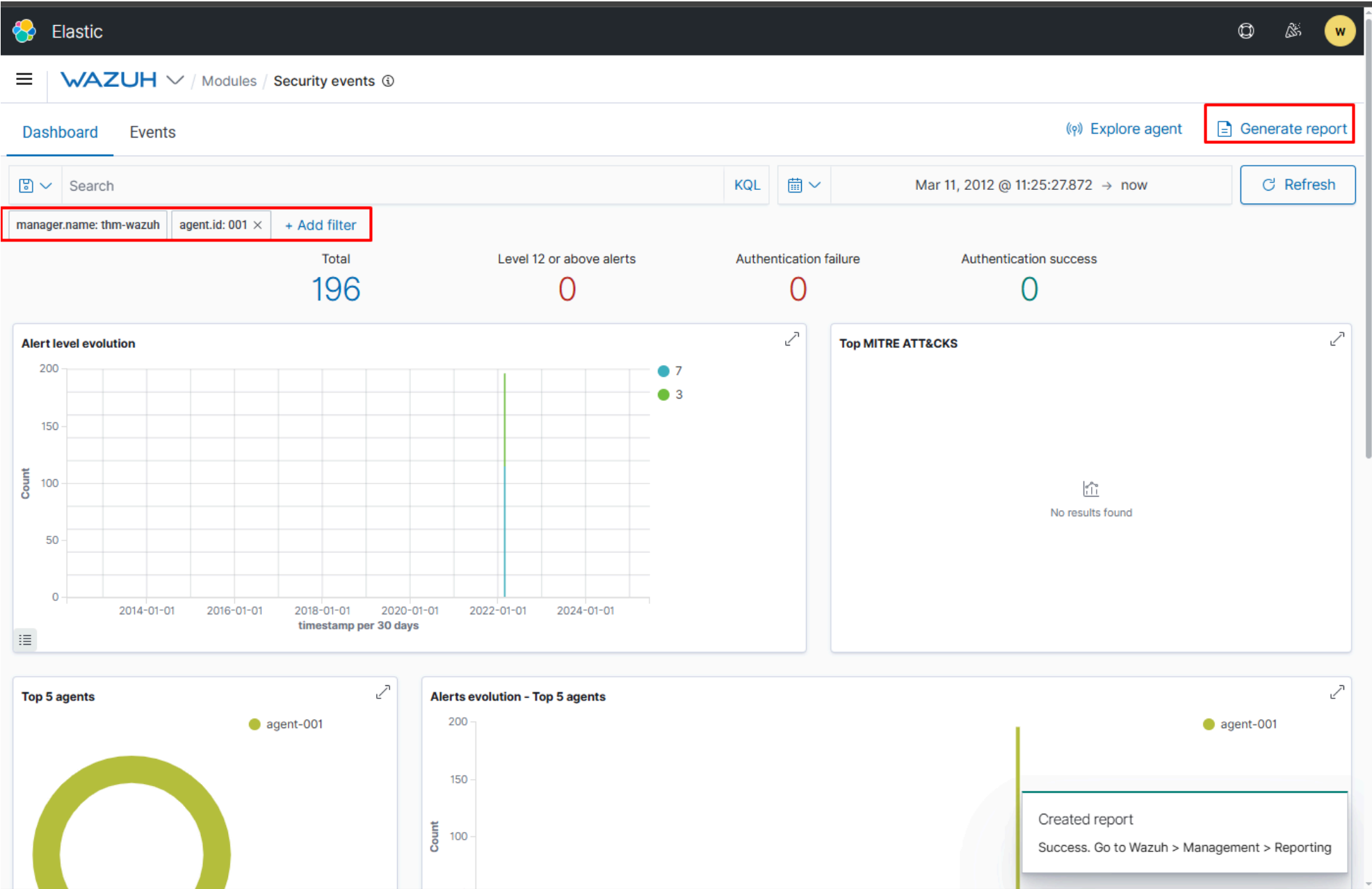


El panel de resumen de informes muestra todos los informes generados. Para descargar un informe, pulse el icono de guardar a la derecha, debajo del encabezado " **Acciones** ".



Responda las preguntas a continuación

Utilice la función "Informe" de Wazuh para generar un informe de un agente.



Completo

Vaya al [panel de "Informes" de Wazuh](#)

The screenshot shows the Wazuh Reporting page. It has a search bar and a table of reports. The table has columns for File, Size, Created, and Actions. The first row is highlighted with a red box.

File	Size	Created	Actions
wazuh-overview-general-1750352037.pdf	66.73KB	Jun 19, 2025 @ 11:53:59.652	Download Delete
wazuh-overview-general-1750351910.pdf	97.80KB	Jun 19, 2025 @ 11:51:53.336	Download Delete

Rows per page: 10

Completo

Analice el informe. ¿Cuál es el nombre del agente que generó más alertas?

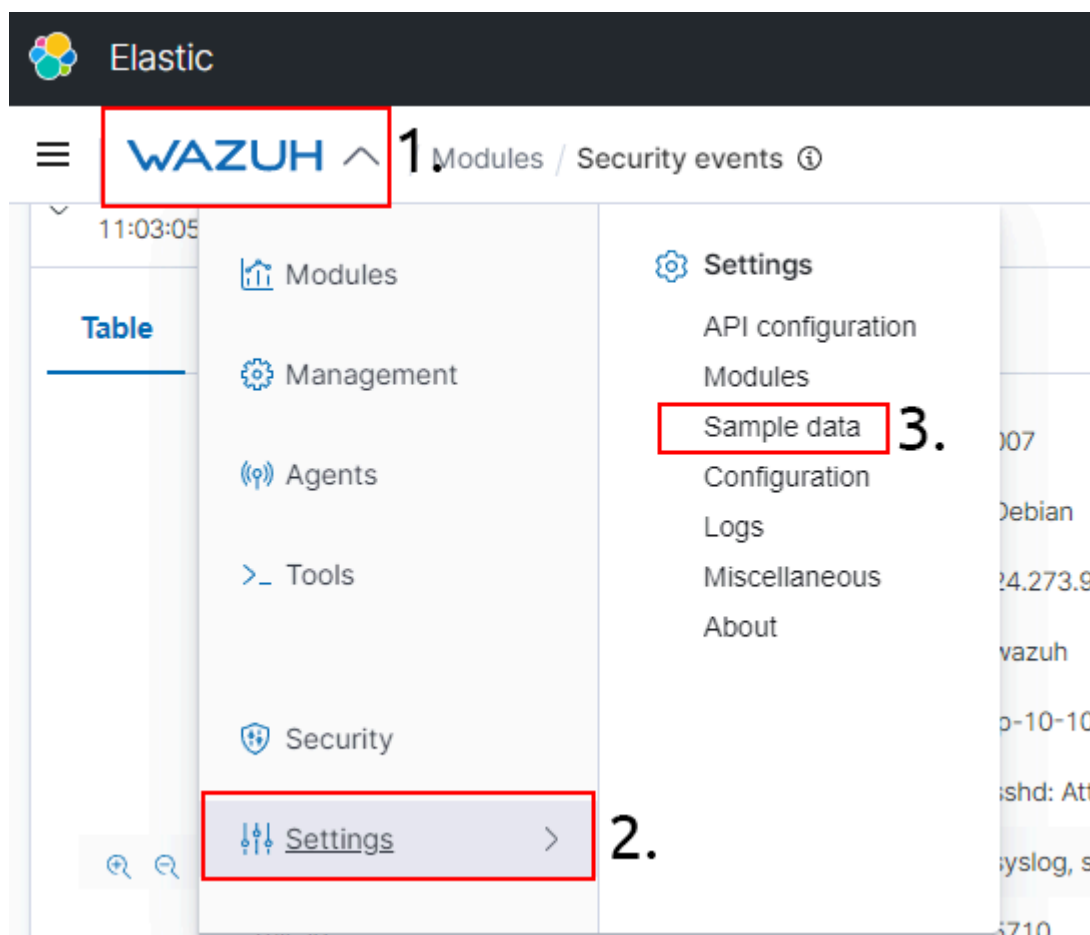
agent-001

Entregar

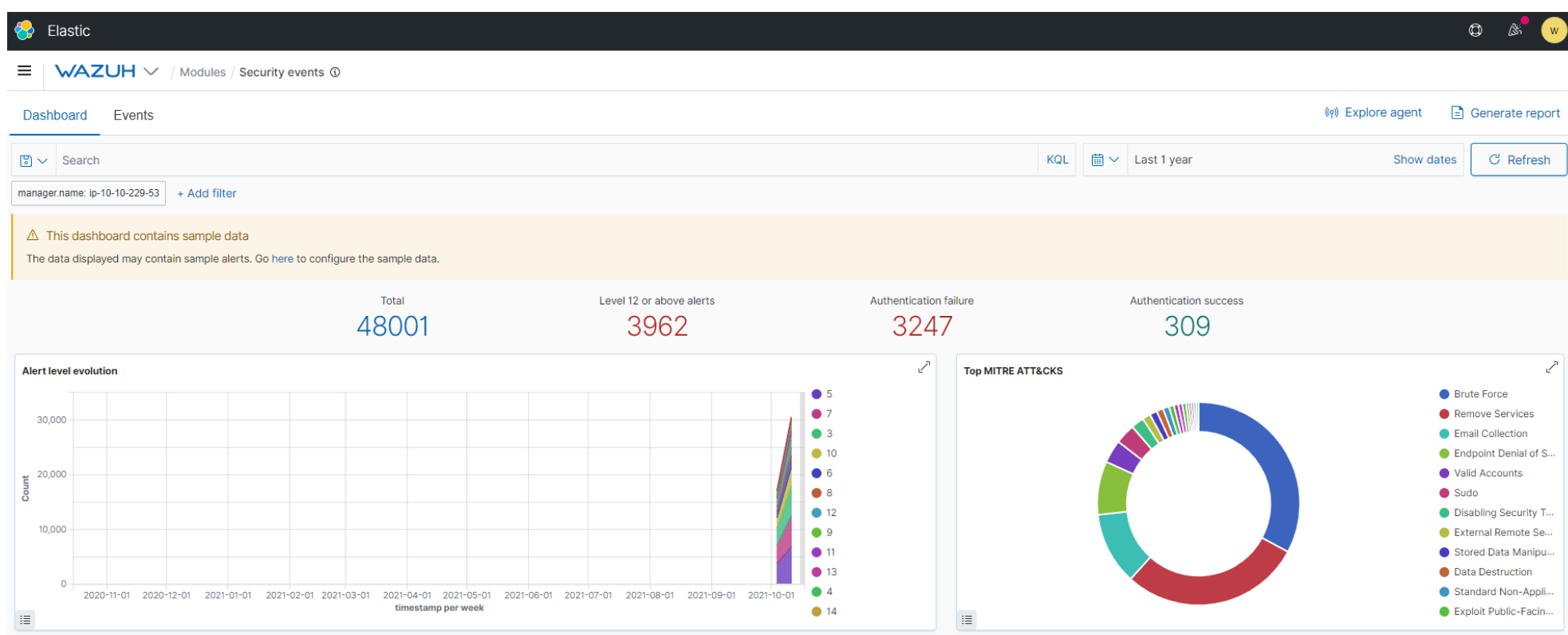
Cargando datos de muestra

El servidor de administración de Wazuh incluye datos de muestra con la instalación, que puede cargar cuando lo desee. No he habilitado esta opción por defecto para mejorar el rendimiento del servidor. Sin embargo, si desea importar muchos más datos para demostrar aún más la extensibilidad de Wazuh , siga los pasos a continuación. Vaya al módulo para cargar los datos de muestra:

1. Abra la pestaña " **Wazuh** " en el encabezado.
2. Resalte " **Configuración** " .
3. Seleccione el encabezado " **Datos de muestra** " .
4. Presione el botón " **Agregar datos** " en las tres tarjetas respectivas para importar los datos.



- Tenga en cuenta que esto puede tardar hasta un minuto. Vea la imagen animada a continuación como ejemplo. Los datos se habrán importado correctamente cuando el botón de la tarjeta indique "Eliminar datos".



Regrese al panel de Wazuh para ver los datos recién importados. Por ejemplo, ahora podemos ver que el módulo "Eventos de Seguridad" tiene muchísimos más datos para explorar.

Tenga en cuenta que deberá modificar el rango de fechas. El mínimo necesario para mostrar la muestra debe ser de más de los últimos 7 días. Para que esto aplique, actualice el panel.

KQL

1.

Calendar icon

▼

last 1 year

Show dates

3.

Refresh icon

Refresh

Quick select

Last

▼

1

▼

years

▼

Apply

Commonly used

Today

Last 24 hours

This week

Last 7 days

Last 15 minutes

Last 30 days

Last 30 minutes

Last 90 days

Last 1 hour

Last 1 year

Recently used date ranges

Last 1 year

Oct 15, 2021 @ 11:16:00.652 to Oct 15, 2022 @ 11:16:00.652

Last 7 weeks

Last 7 days

Refresh every

0

seconds

▼

▶ Start

Brute Force

Remove Services

Email Collection

Endpoint Denial of S...

Valid Accounts

Sudo

Disabling Security T...

External Remote Se...

Stored Data Manipu...

Data Destruction

Standard Non-Appli...