



**Certificado de Profesionalidad en
Seguridad Informática
IronHack - SOC**

**Módulo 3
Gestión de Incidentes
Práctica 2 - Wazuh**

Alumno: Julián Gordon

Indice

| | |
|---|-----------|
| Indice..... | 2 |
| Enunciado..... | 3 |
| Introducción..... | 4 |
| Registro de usuario en Wazuh..... | 5 |
| Despliegue de Agentes..... | 7 |
| Activación de servicios en Kali Linux..... | 16 |
| Verificación de Eventos en Wazuh Cloud..... | 17 |
| Generación de nuevos eventos de Seguridad..... | 19 |
| Conclusiones..... | 24 |

Enunciado

1. Registro en Wazuh Cloud

Accede a la página de registro de Wazuh Cloud. Utiliza una cuenta de correo temporal para completar el registro. Verifica tu cuenta a través del correo recibido y accede al portal de Wazuh Cloud.

2. Creación del Entorno en Wazuh

Inicia sesión en Wazuh Cloud. Crea un nuevo entorno desde el dashboard principal. Configura el entorno según las especificaciones necesarias.

3. Despliegue de Agentes

Máquina Virtual Windows: Crea un agente desde el portal de Wazuh Cloud e invoca con el comando necesario y desde la máquina virtual la instalación del agente de Wazuh. Inicia el servicio.

Máquina Virtual Kali: Crea un agente desde el portal de Wazuh Cloud e invoca con el comando necesario y desde la máquina virtual la instalación del agente de Wazuh. Inicia el servicio.

4. Activación de Servicios en Kali

Verifica y activa los servicios de seguridad previamente instalados: Snort: Configura y activa el servicio. Suricata: Configura y activa el servicio.

WAF (mod-security): Configura y activa el servicio en el servidor web.

5. Verificación de Eventos en Wazuh Cloud

Accede al dashboard de Wazuh Cloud y verifica que los eventos de seguridad de las máquinas virtuales se reflejan correctamente.

6. Generación de Nuevos Eventos de Seguridad

Máquina Windows: Genera varios intentos de inicio de sesión fallidos.

Máquina Kali: Realiza un intento de inyección de código en un formulario web protegido por WAF.

Ejecuta acciones para activar reglas de Snort y Suricata generando tráfico sospechoso.

7. Documentación del Proceso

Compila todas las capturas de pantalla en un documento. Redacta una breve descripción para cada paso, explicando el proceso seguido y los resultados obtenidos. Asegúrate de incluir cualquier problema encontrado y cómo se resolvió.

Opción Alternativa: Instalación de Wazuh mediante OVA

Descarga la OVA oficial de Wazuh desde el sitio web de Wazuh. Importa la OVA en tu software de virtualización preferido (por ejemplo, VirtualBox, VMware).

Sigue las instrucciones de instalación y configuración para desplegar Wazuh en una máquina virtual. Repite los pasos 3-6 para el despliegue de agentes y generación de eventos. Documento PDF con todas las capturas de pantalla y descripciones detalladas.

Introducción

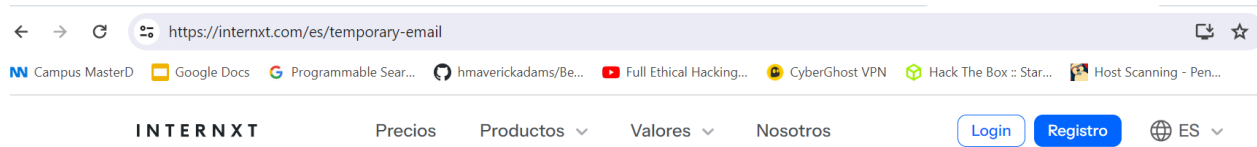
En el ámbito de la ciberseguridad, la capacidad de detectar, analizar y responder a amenazas de manera eficaz es crucial para proteger la integridad y confidencialidad de los sistemas de información. Wazuh es una plataforma integral de monitoreo de seguridad que permite gestionar eventos de seguridad y garantizar la protección de los activos tecnológicos de una organización. Este trabajo práctico se centra en la implementación de Wazuh Cloud, una solución basada en la nube que ofrece una gestión centralizada de la seguridad.

El objetivo de este ejercicio es guiarnos a través del proceso completo de registro en Wazuh Cloud, la creación y configuración de un entorno de seguridad, el despliegue de agentes en máquinas virtuales con sistemas operativos Windows y Kali Linux, y la activación de diversos servicios de seguridad en Kali Linux. Además, se generan y verifican eventos de seguridad para demostrar la capacidad de Wazuh para identificar y gestionar incidentes de seguridad.

Durante el desarrollo del trabajo, se documenta detalladamente cada paso y procedimiento, proporcionando una guía práctica y útil para futuros usuarios de Wazuh Cloud. A través de capturas de pantalla y descripciones detalladas, se asegura una comprensión clara de cada fase del proceso, destacando la facilidad de uso y la robustez de la plataforma para la gestión de la seguridad informática.

Registro de usuario en Wazuh

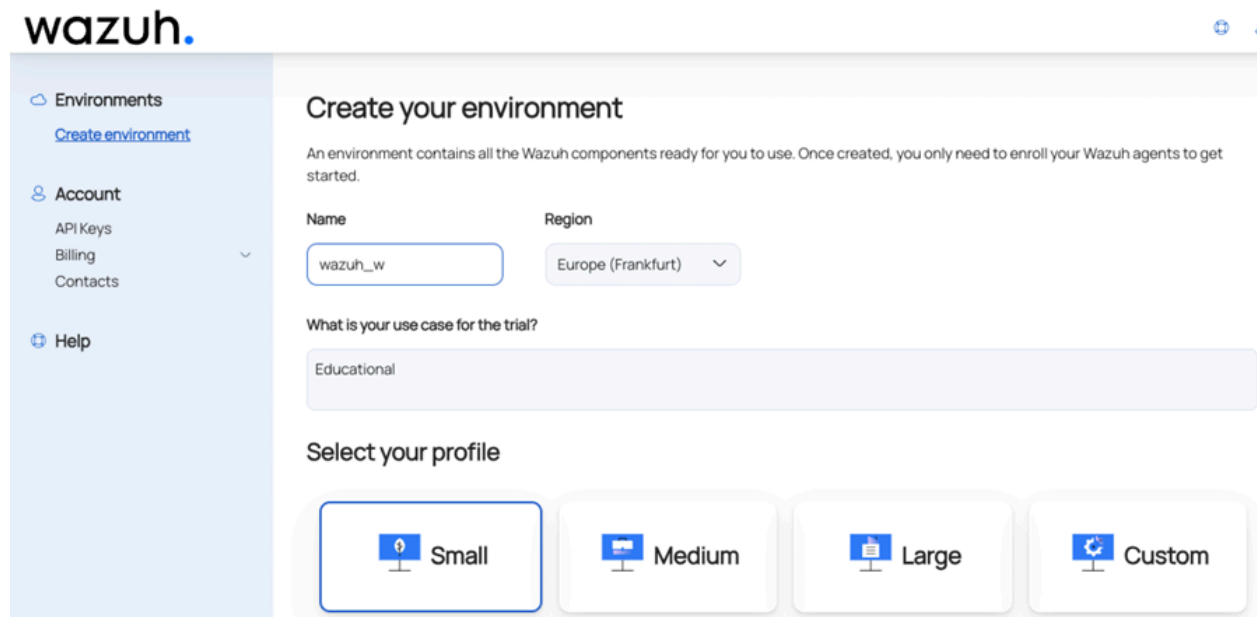
Comenzaremos esta práctica registrando en el sitio web de Wazuh. Para ello crearemos un mail temporal desde la página web <https://internxt.com/es/temporary-email>.



Luego nos registramos en la página web de Wazuh <https://console.cloud.wazuh.com/sign-up>.

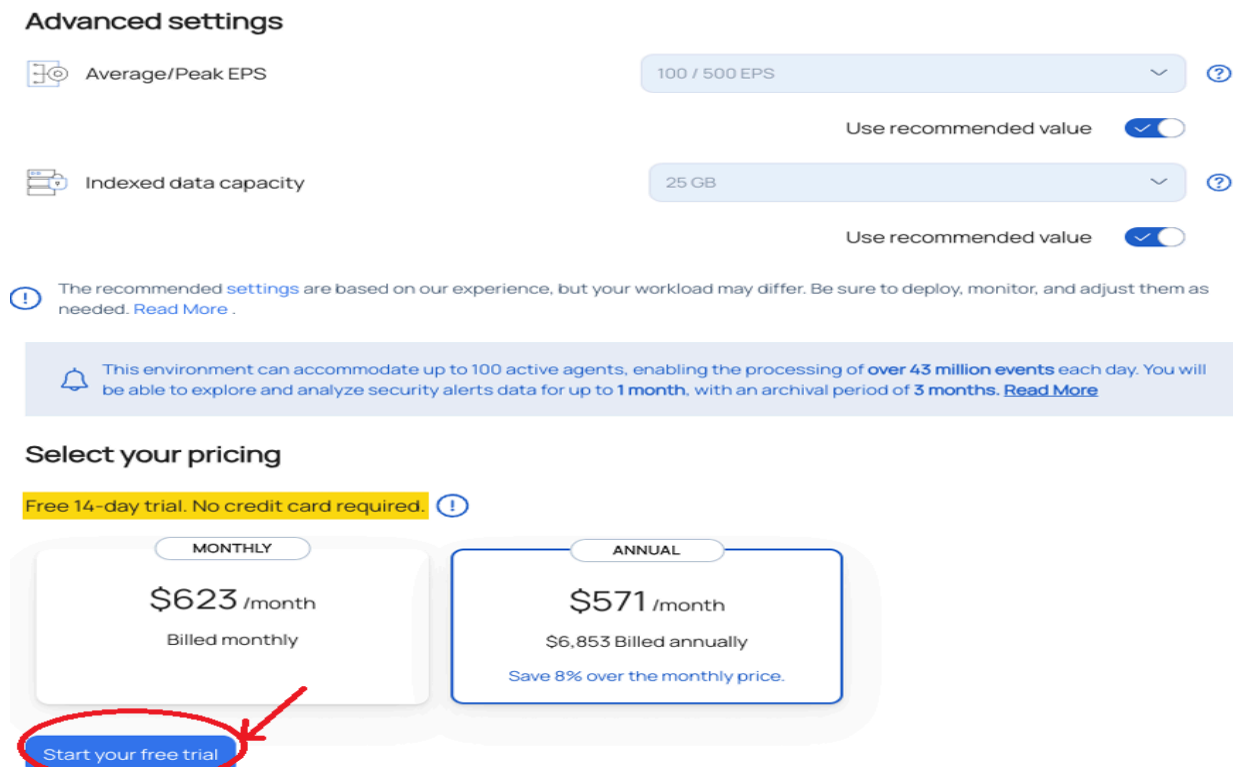
A screenshot of the Wazuh Cloud sign-up page. The browser address bar shows 'https://console.cloud.wazuh.com/sign-up'. The page has a dark header with various links. The main content area is titled 'Create your account' and includes a link 'Already have an account? Log in'. On the left, there is promotional text about 'Wazuh Cloud' and a 'Start your Free Trial' section. On the right, there is a registration form with fields for 'First name', 'Family name', 'Business email', 'Phone number', 'Password', 'Company', and 'Country'. A 'Create account' button is at the bottom right.

Ahora accedemos y crearemos un entorno desde el dashboard principal.



The image shows the 'Create your environment' page in the Wazuh dashboard. On the left is a sidebar with 'Environments', 'Account', and 'Help'. The main area has a title 'Create your environment' and a description. Below this are input fields for 'Name' (wazuh_w) and 'Region' (Europe (Frankfurt)). A 'What is your use case for the trial?' section has 'Educational' selected. A 'Select your profile' section shows four options: 'Small', 'Medium', 'Large', and 'Custom', with 'Small' selected.

Le damos un nombre y le asignamos la región Europe (Frankfurt). Asignamos según nuestros recursos los tipos de planes que deseamos contratar. En nuestro caso iremos por la versión gratis y seleccionamos free trial.

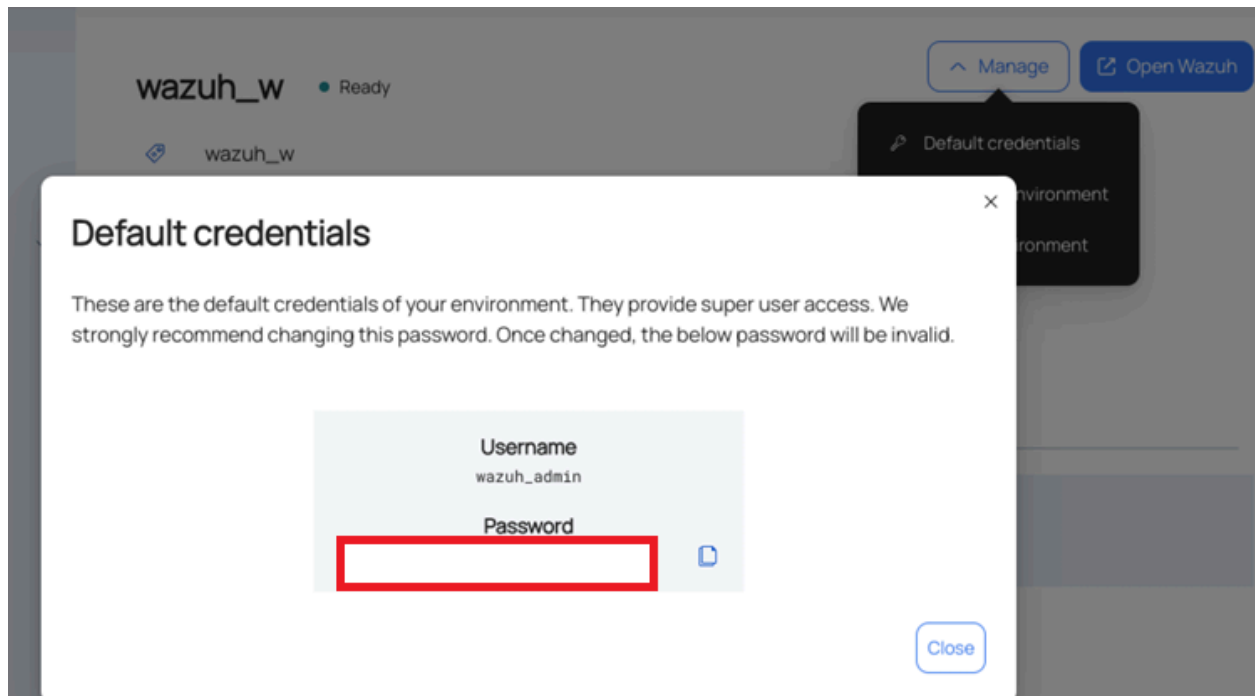


This section contains 'Advanced settings' and 'Select your pricing'. Under 'Advanced settings', there are two rows: 'Average/Peak EPS' set to '100 / 500 EPS' and 'Indexed data capacity' set to '25 GB', both with 'Use recommended value' toggled on. A note below states that recommended settings are based on experience. A blue box below that states the environment can accommodate up to 100 active agents. The 'Select your pricing' section features a yellow banner for a 'Free 14-day trial. No credit card required.' Below are two pricing cards: 'MONTHLY' at \$623/month and 'ANNUAL' at \$571/month. A red circle and arrow point to a 'Start your free trial' button at the bottom left.

Despliegue de Agentes

El siguiente paso en este ejercicio es crear agentes para registros de nuestras máquinas virtuales de Windows y de Kali Linux.

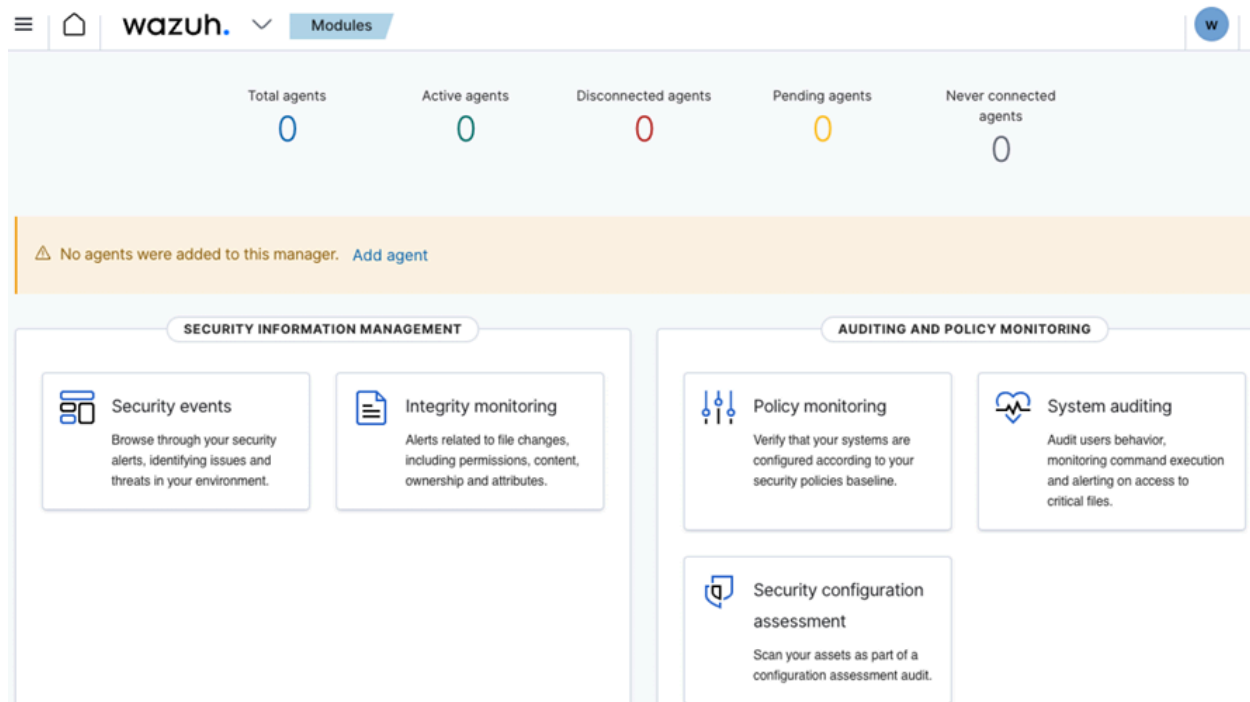
Vamos a la pestaña Manage y luego debajo veremos Default Credentials. Allí estará nuestro usuario y password que debemos utilizar para activar Wazuh.



Ahora que ya sabemos nuestras credenciales, iremos al icono de Open Wazuh.



Una vez dentro podemos ver el panel de control.



Crearemos ahora un agente para la máquina de Kali Linux.

En primer lugar le daremos un nombre.



Optional settings:

By default, the deployment uses the hostname as the agent name. Optionally, you can use a different agent name in the field below.


Assign an agent name: ?

Kali


ⓘ The agent name must be unique. It can't be changed once the agent has been enrolled. ↗

Deploy new agent


✓ Select the package to download and install on your system:


LINUX

☐ RPM amd64
 ☐ RPM aarch64
 ☒ DEB amd64
 ☐ DEB aarch64


WINDOWS

☐ MSI 32/64 bits


macOS

☐ Intel
 ☐ Apple silicon

Seleccionamos DEB amd64 que es el que corresponde a nuestro Kali Linux que está basado en Debian.

Ahora nos dará la dirección del Server.

✓ Server address:

This is the address the agent uses to communicate with the server. Enter an IP address or a fully qualified domain name (FDQN).

Assign a server address: ?

6v0pu385fbs.cloud.wazuh.com

Luego debemos ejecutar este comando en nuestra consola de Kali Linux.

✓ Run the following commands to download and install the agent:

```
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.3-1_amd64.deb &&
sudo WAZUH_MANAGER='6v0pu385fbs.cloud.wazuh.com'
WAZUH_REGISTRATION_PASSWORD='$' WAZUH_AGENT_NAME='Kali' dpkg -i
./wazuh-agent_4.7.3-1_amd64.deb
```

```

(root@kali)-[/home/kali]
# wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.3-1_amd64.deb
2024-05-14 10:26:13-- https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.3-1
Resolving packages.wazuh.com (packages.wazuh.com) ... 18.154.48.50, 18.154.48.93, 18.154.48.95, ...
Connecting to packages.wazuh.com (packages.wazuh.com)|18.154.48.50|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 9362524 (8.9M) [binary/octet-stream]
Saving to: 'wazuh-agent_4.7.3-1_amd64.deb'

wazuh-agent_4.7.3-1_amd64.deb  100%[=====] 8.93M 2
2024-05-14 10:26:18 (2.27 MB/s) - 'wazuh-agent_4.7.3-1_amd64.deb' saved [9362524/9362524]

Selecting previously unselected package wazuh-agent.
(Reading database ... 420448 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.7.3-1_amd64.deb ...
Unpacking wazuh-agent (4.7.3-1) ...

```

Una vez que termina la instalación, iniciaremos el agente, abriendo una terminal y ejecutando este comando:



Start the agent:

```

sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent

```

```

(root@kali)-[/home/kali]
# sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /usr/lib/systemd/system/wazuh-agent.service.

```

Para confirmar que el agente está activo y funcionando usamos el comando:

`systemctl status wazuh-agent`

```
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enab>
   Active: active (running) since Sat 2024-05-18 12:27:22 CEST; 9min>
     Tasks: 32 (limit: 4610)
   Memory: 516.9M (peak: 612.4M)
      CPU: 1min 26.925s
   CGroup: /system.slice/wazuh-agent.service
           └─20855 /var/ossec/bin/wazuh-execd
             └─20863 /var/ossec/bin/wazuh-agentd
               └─20885 /var/ossec/bin/wazuh-syscheckd
                 └─20906 /var/ossec/bin/wazuh-logcollector
                   └─20915 /var/ossec/bin/wazuh-modulesd

May 18 12:27:18 kali systemd[1]: Starting wazuh-agent.service - Wazuh >
May 18 12:27:18 kali env[20454]: Starting Wazuh v4.7.3 ...
May 18 12:27:18 kali env[20454]: Started wazuh-execd ...
May 18 12:27:18 kali env[20454]: Started wazuh-agentd ...
May 18 12:27:19 kali env[20454]: Started wazuh-syscheckd ...
May 18 12:27:19 kali env[20454]: Started wazuh-logcollector ...
May 18 12:27:20 kali env[20454]: Started wazuh-modulesd ...
May 18 12:27:22 kali env[20454]: Completed.
May 18 12:27:22 kali systemd[1]: Started wazuh-agent.service - Wazuh a>
lines 1-22/22 (END)
```

Ahora debemos crear el agente para Windows 10, siguiendo los mismos pasos.





Agents (1)

[⊕ Deploy new agent](#)

[↻ Refresh](#)


[📄 Export formatted](#)

Ahora seleccionamos Windows MSI 32/64 bits




Agents


W

 Refresh

Deploy new agent



Select the package to download and install on your system:




LINUX

☐ RPM amd64

☐ RPM aarch64

☐ DEB amd64

☐ DEB aarch64



WINDOWS

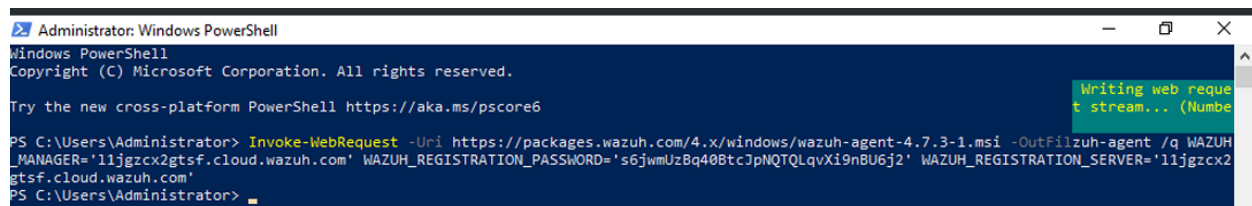
☒ MSI 32/64 bits

Ahora ejecutaremos el comando que nos indica en una terminal de Powershell en nuestra máquina virtual de Windows 10.

4

Run the following commands to download and install the agent:

```
Invoke-WebRequest -Uri
https://packages.wazuh.com/4.x/windows
/wazuh-agent-4.7.3-1.msi -OutFile
${env.tmp}\wazuh-agent; msixec.exe /i
${env.tmp}\wazuh-agent /q
WAZUH_MANAGER='l1jgzc2gtsf.cloud.wazuh.com'
WAZUH_REGISTRATION_PASSWORD='*****
*****'
WAZUH_REGISTRATION_SERVER='l1jgzc2gtsf.clou
d.wazuh.com'
```



The screenshot shows a Windows PowerShell terminal window titled "Administrator: Windows PowerShell". The command prompt shows the execution of the `Invoke-WebRequest` command with the following parameters: `-Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.7.3-1.msi -OutFile ${env.tmp}\wazuh-agent; msixec.exe /i ${env.tmp}\wazuh-agent /q WAZUH_MANAGER='l1jgzc2gtsf.cloud.wazuh.com' WAZUH_REGISTRATION_PASSWORD='s6jwmUzBq40Btc3pNQTLqvX19nBU6j2' WAZUH_REGISTRATION_SERVER='l1jgzc2gtsf.cloud.wazuh.com'`. The command is executed successfully, and the prompt returns to `PS C:\Users\Administrator>`. A green tooltip on the right side of the terminal window reads "Writing web request stream... (Number)".

Seguimos el siguiente paso que nos indica y ejecutamos el comando:

NET START WazuhSvc

5

Start the agent:

```
NET START WazuhSvc
```

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> NET START WazuhSvc
The Wazuh service is starting.
The Wazuh service was started successfully.
PS C:\Users\Administrator>
```

Para verificar si todo está correcto y el agente está activo utilizamos el comando:

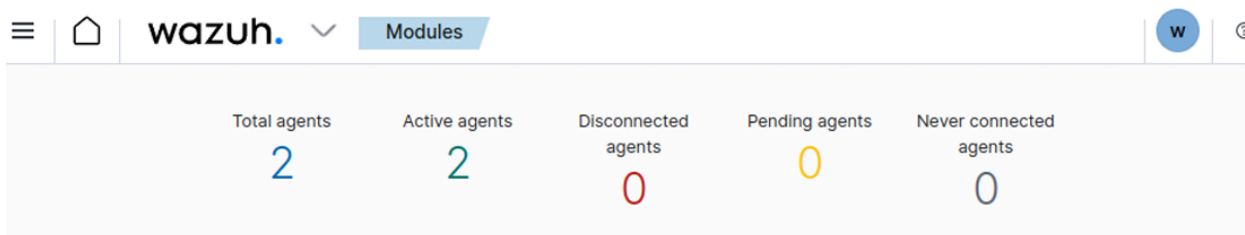
Get-Service -Name WazuhSvc

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Get-Service -Name WazuhSvc

Status      Name      DisplayName
-----
Running     WazuhSvc  Wazuh

PS C:\Users\Administrator>
```

Ya tenemos los agentes de Windows 10 y Kali Linux configurados correctamente y activos. Podemos verificar esto desde el navegador.



Activación de servicios en Kali Linux

El siguiente paso será activar servicios para verificar que Wazuh puede detectar estos servicios y eventos. Para ello vamos a activar varios servicios que utilizamos anteriormente como apache2, mariadb, snort, etc.

```
(root@kali)-[/home/kali]
# systemctl start apache2
```

```
(root@kali)-[/home/kali]
# systemctl start mariadb
```

```
(root@kali)-[/home/kali]
# a2enmod security2
```

Considering dependency unique_id for security2:
Module unique_id already enabled
Module security2 already enabled

```
(root@kali)-[/home/kali]
# a2enmod headers
```

Module headers already enabled

```
(root@kali)-[/home/kali]
# systemctl start snort
```

Verificación de Eventos en Wazuh Cloud

Ahora vamos a verificar estos eventos en Wazuh, para ello volvemos al panel de control en el navegador y vamos al apartado Agentes y veremos que tenemos los 2 activos.

Agents (2)

id!=000 and

status=active

WQL

Refresh

ID ↑

Name

IP address

Group(s)

Operating system

Cluster node

Version

Status


Actions

001

kali

10.0.3.15


default


 Kali GNU/Linux 2024.1

wazuh-manager-master-0

v4.7.3

?






002

W10

10.0.3.15


default


 Microsoft Windows 10 10.0.19045.4412

wazuh-manager-master-0

v4.7.3

?





Seleccionamos la máquina de Kali Linux.

Navigation: [wazuh.](#) [Agents](#) [kali](#) [w](#) [🔍](#)

Modules [Inventory data](#) [Stats](#) [Configuration](#)

| ID | Status | IP address | Version | Groups | Operating system | Cluster node |
|-----|----------------------------------|------------|--------------|-------------------------|-----------------------|--------------------|
| 001 | ● ? | 10.0.3.15 | Wazuh v4.7.3 | default | Kali GNU/Linux 2024.1 | wazuh-manager-m... |

Registration date: May 18, 2024 @ 12:27:19.000 | Last keep alive: May 18, 2024 @ 15:40:25.000

MITRE

Top Tactics

- Defense Evasion: 39
- Privilege Escalation: 38
- Initial Access: 25
- Persistence: 25

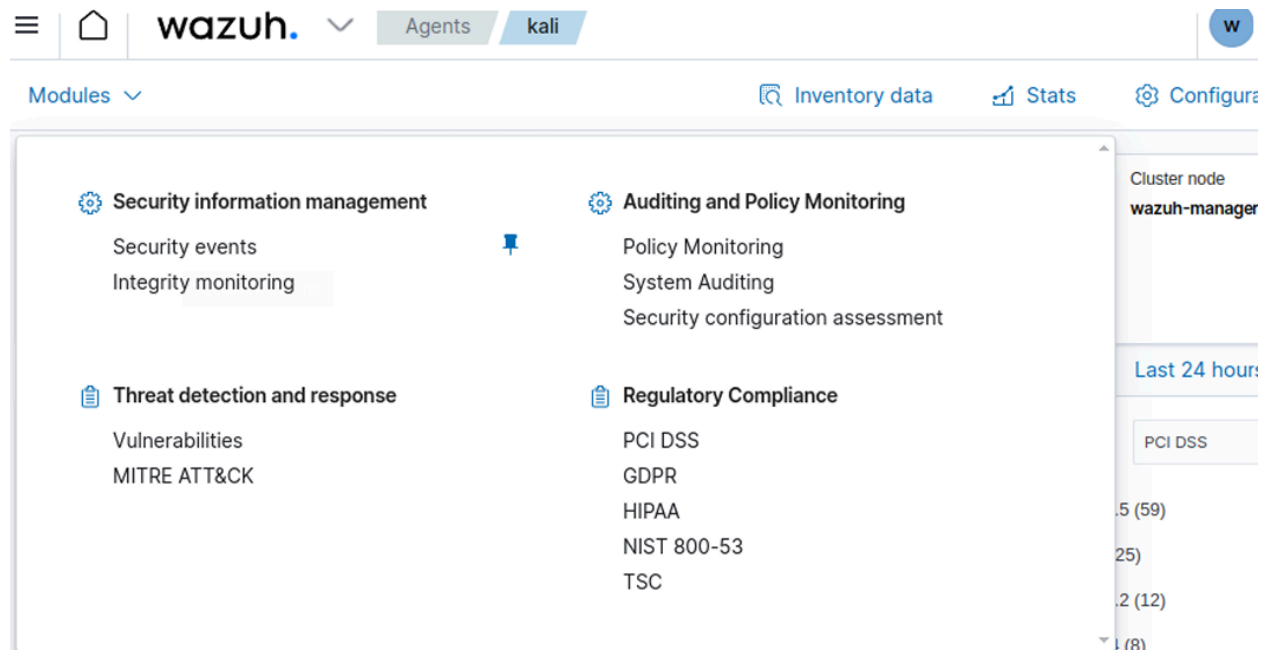
Compliance

PCI DSS

| Version | Count |
|---------|-------|
| 10.2.5 | 59 |
| 2.2 | 25 |
| 10.2.2 | 12 |
| 2.2.4 | 8 |
| 10.6.1 | 7 |

Podemos observar a la izquierda que Wazuh nos proporciona visibilidad detallada sobre amenazas y vulnerabilidades alineadas con la matriz MITRE ATT&CK, permitiéndonos identificar, rastrear y mitigar técnicas y tácticas utilizadas por actores maliciosos.

Podemos observar entrando al apartado Module.

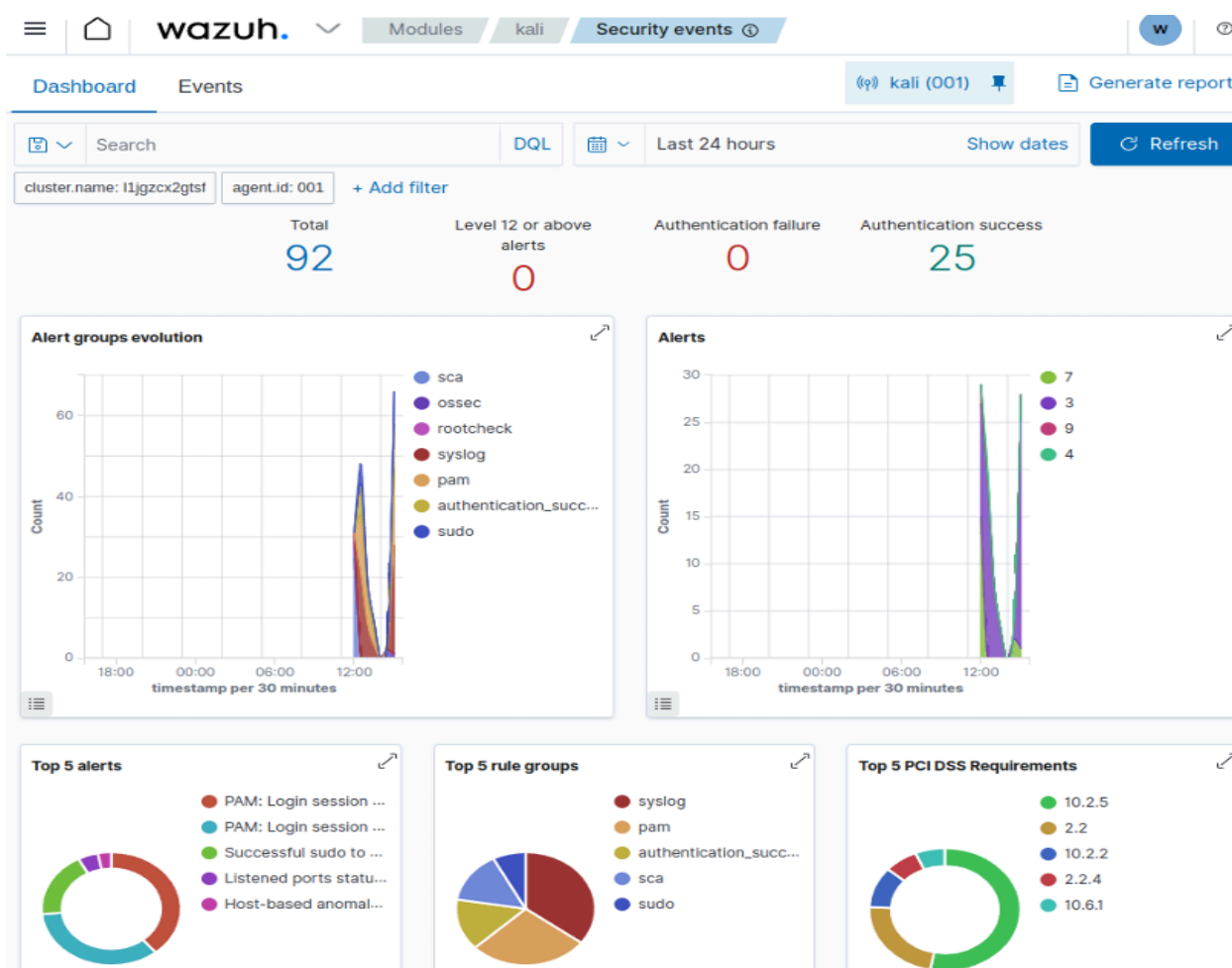


Este apartado muestra información específica sobre los diferentes módulos de seguridad que están activos y operando en el sistema. Estos módulos abarcan diversas funciones, como la detección de intrusiones, la supervisión de la integridad de archivos, el análisis de vulnerabilidades y la recopilación de registros, proporcionando un resumen detallado de las actividades y eventos de seguridad gestionados por cada módulo.

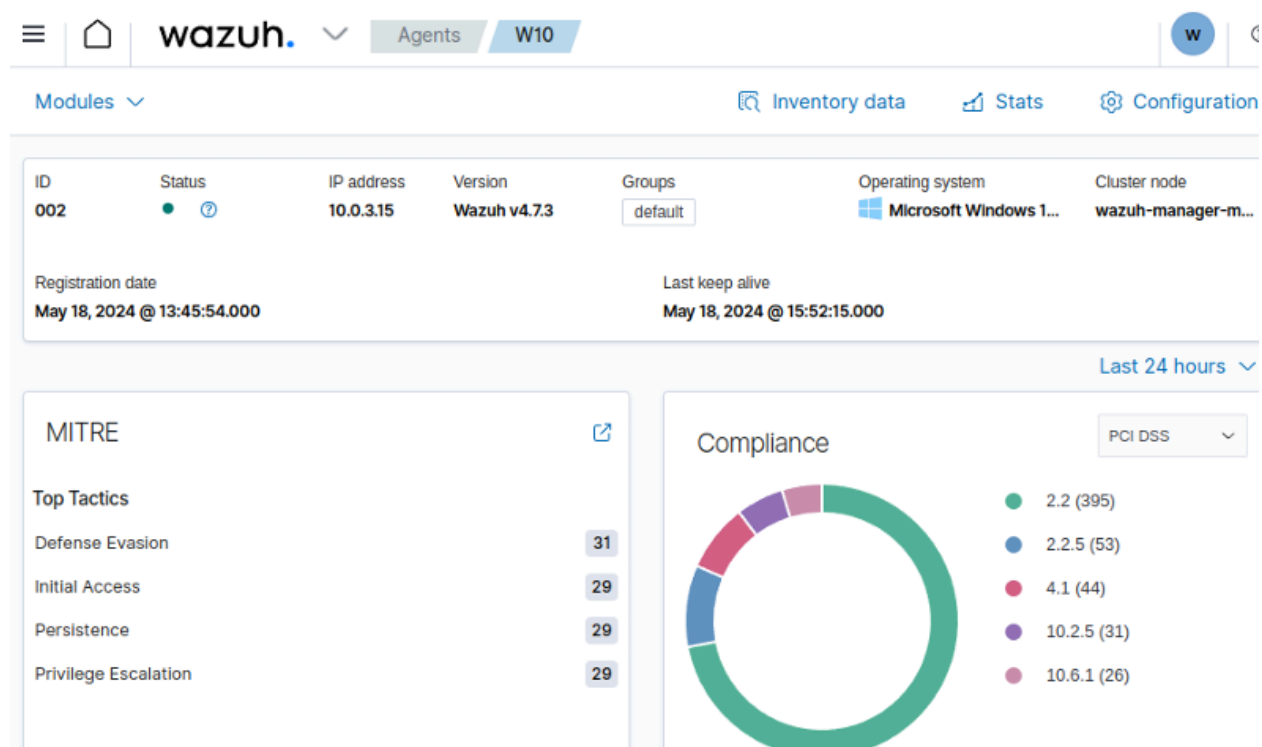
Generación de nuevos eventos de Seguridad

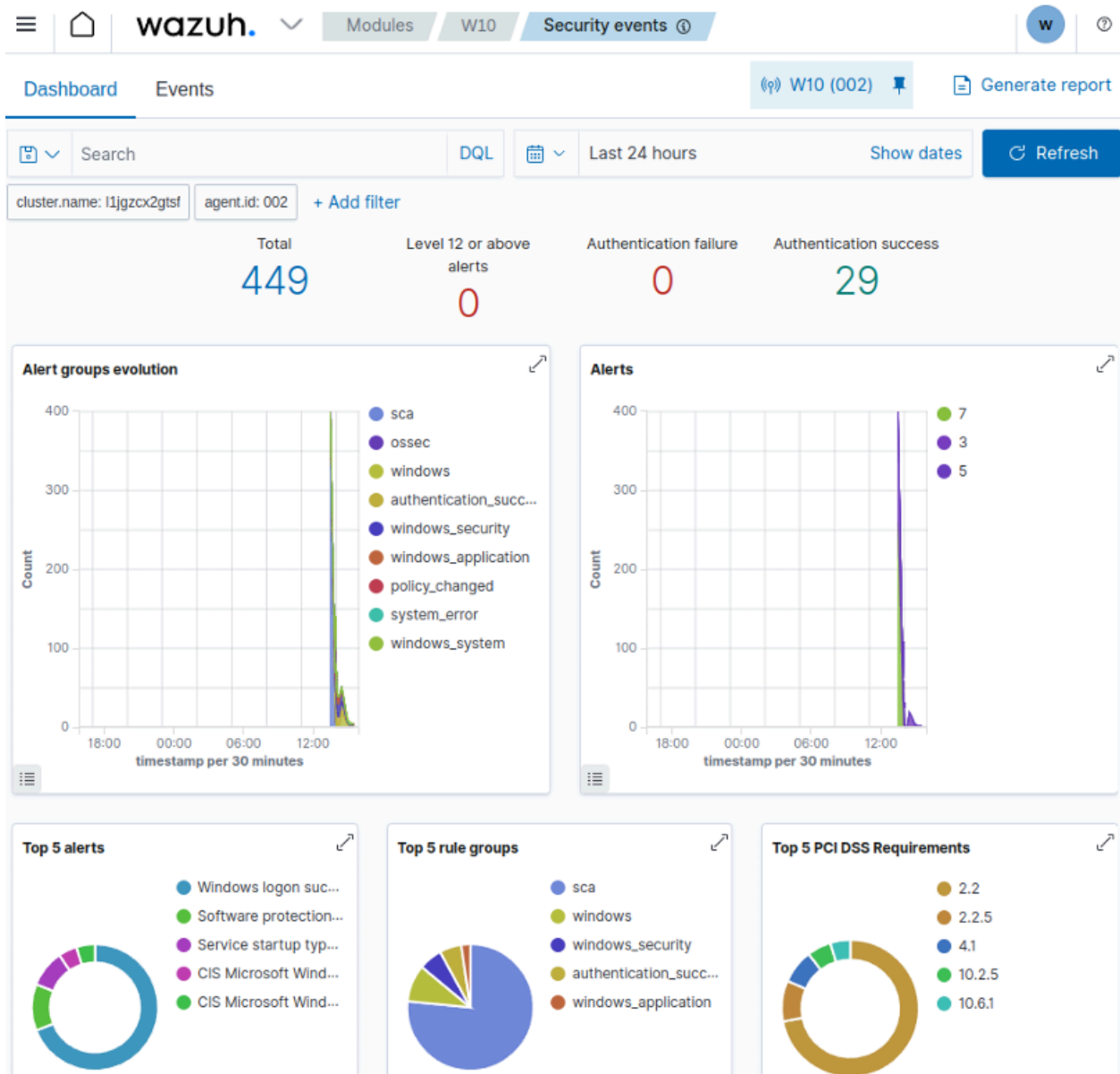
Si accedemos al apartado Events, podemos ver todos los eventos de seguridad que se generaron, con fecha, descripción, nivel y número de identificación. Para este paso hemos realizado algunos intentos de Login con contraseñas incorrectas y con contraseñas correctas dentro de nuestra máquina de Kali Linux.

| Time ↓ | Technique(s) | Tactic(s) | Description | Level | Rule ID |
|-------------------------------|--------------|--|---|-------|---------|
| > May 14, 2024 @ 16:32:36.938 | T1110.001 | Credential Access | PAM: User login failed. | 5 | 5503 |
| > May 14, 2024 @ 16:31:52.960 | T1110.001 | Credential Access | PAM: User login failed. | 5 | 5503 |
| > May 14, 2024 @ 16:31:44.842 | T1078 | Defense Evasion, Persistence, Privilege Escalation, Initial Access | PAM: Login session opened. | 3 | 5501 |
| > May 14, 2024 @ 16:31:44.697 | T1078 | Defense Evasion, Persistence, Privilege Escalation, Initial Access | PAM: Login session opened. | 3 | 5501 |
| > May 14, 2024 @ 16:31:44.697 | T1078 | Defense Evasion, Persistence, Privilege Escalation, Initial Access | PAM: Login session opened. | 3 | 5501 |
| > May 14, 2024 @ 16:31:42.794 | | | Systemd: Service exited due to a failure. | 5 | 40704 |

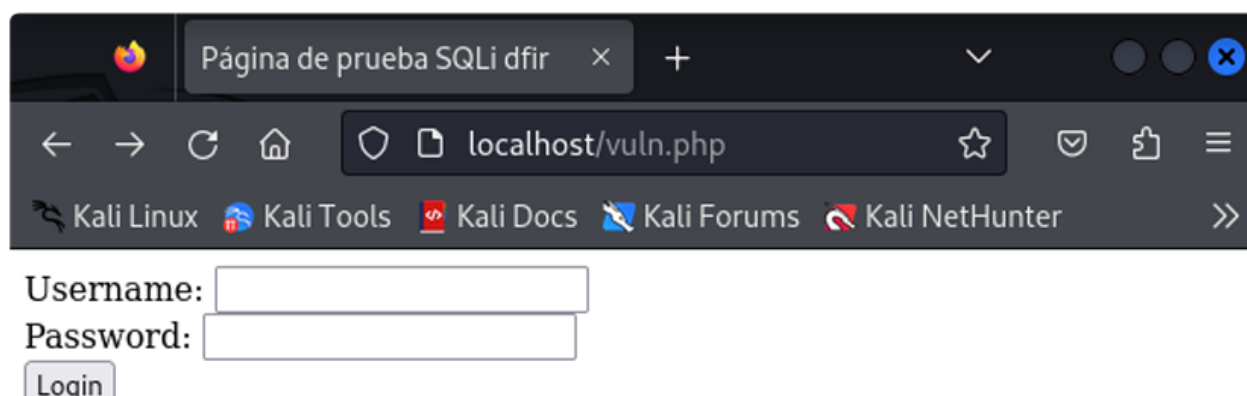


Podemos observar igualmente la misma información pero ahora en Windows10.

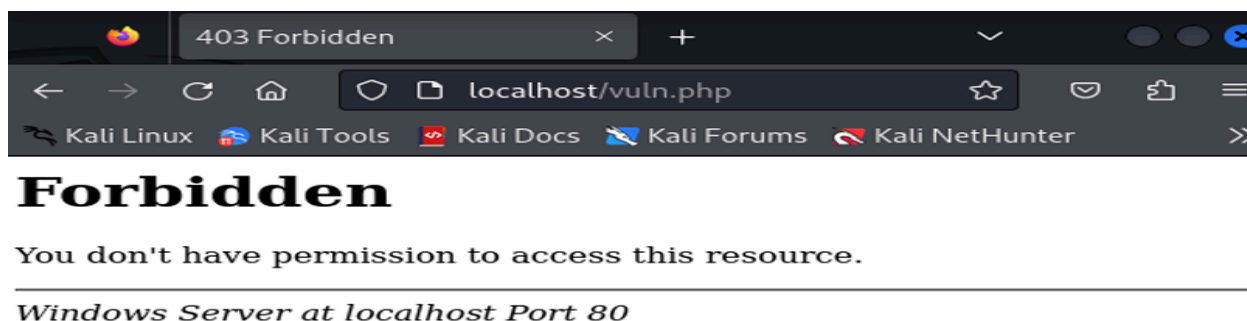
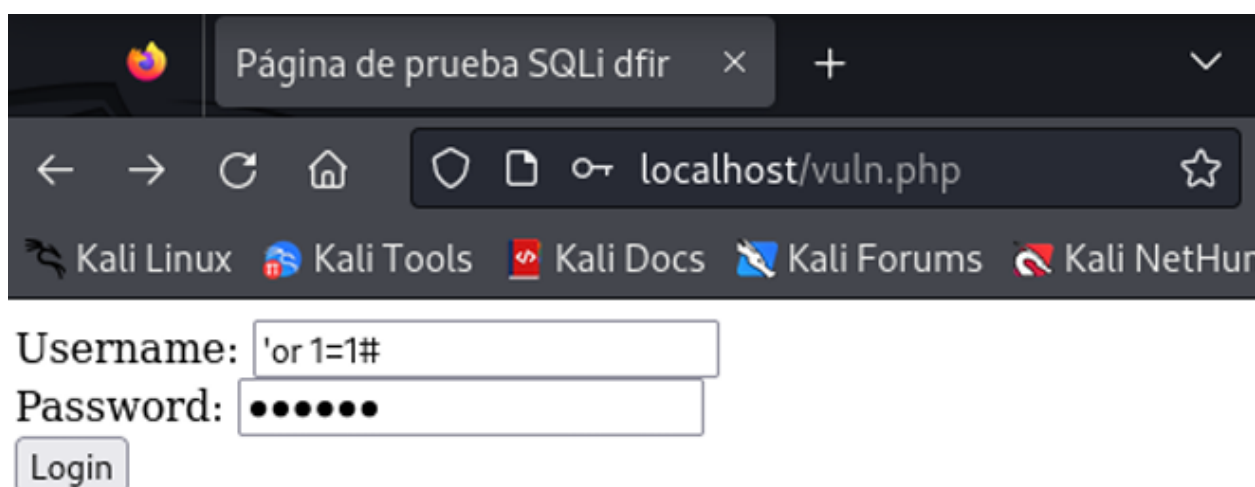




El siguiente paso que haremos será una prueba de inyección de código en un formulario web protegido por WAF. En el examen práctico del módulo anterior, creamos una servidor apache2 y una base de datos mariadb. Además creamos un fichero llamado vuln.php que tiene un formulario para login. Usaremos esto para demostrar el funcionamiento en esta práctica. Como ya tenemos activados y corriendo los servicios, entraremos en nuestro navegador desde Kali Linux a la página web localhost/vuln.php .



Ahora vamos a intentar inyectar código, escribiendo en Username: 'or 1=1#. Verificamos que no nos dejará, ya que tenemos activado el mod-security.



Ahora dentro de Wazuh, podemos verificar que se generó un evento de seguridad rechazando la inyección SQL.

| | | | | | | |
|--------------------------|-----------------------------|-------------------------------|------------|---------|---|---|
| ≡ | 🏠 | wazuh. ▾ | Modules | kali | W | ? |
| Security events ⓘ | | | | | | |
| Timestamp per 60 minutes | | | | | | |
| | Time ▾ | rule.description | rule.level | rule.id | | |
| > | May 18, 2024 @ 23:55:09.299 | Web server 400 error code. | 5 | 31101 | | |
| > | May 18, 2024 @ 23:55:09.291 | ModSecurity: Rejected a query | 7 | 30411 | | |

Conclusiones

En este trabajo práctico, logramos realizar una implementación completa de Wazuh Cloud, incluyendo la creación de un entorno de seguridad y la configuración de agentes en máquinas virtuales con sistemas operativos Windows y Kali Linux. A través de este proceso, hemos demostrado la capacidad de Wazuh para gestionar y monitorizar eventos de seguridad de manera efectiva.

1. **Registro y Configuración Inicial:** El registro en Wazuh Cloud fue sencillo y directo, utilizando una cuenta de correo temporal para acceder al portal. La creación del entorno y la configuración inicial se completaron sin complicaciones, lo que subraya la accesibilidad y facilidad de uso de la plataforma para nuevos usuarios.
2. **Despliegue de Agentes:** La instalación y activación de agentes en las máquinas virtuales fueron exitosas, permitiendo que ambos sistemas (Windows y Kali Linux) fueran monitorizados por Wazuh. Los comandos y procedimientos utilizados garantizaron que los agentes se integraran correctamente y comenzaran a reportar datos al entorno de Wazuh Cloud.
3. **Activación de Servicios en Kali Linux:** La configuración y activación de servicios de seguridad como WAF (mod-security) en Kali Linux demostraron la capacidad de Wazuh para interactuar con múltiples herramientas de seguridad. Esto facilitó la detección y registro de eventos relevantes, mejorando la cobertura de seguridad.
4. **Verificación y Generación de Eventos de Seguridad:** Al verificar los eventos en Wazuh Cloud, se pudo confirmar que los agentes estaban reportando correctamente. La generación de nuevos eventos, como intentos de inicio de sesión fallidos y pruebas de inyección de código, mostraron cómo Wazuh detecta y documenta actividades sospechosas, proporcionando visibilidad detallada de las amenazas.
5. **Documentación y Resolución de Problemas:** La documentación detallada del proceso, junto con capturas de pantalla y descripciones claras, permitió un seguimiento preciso de cada paso. Cualquier problema encontrado fue resuelto

de manera efectiva, lo que refuerza la importancia de una buena práctica de registro y solución de problemas en implementaciones de seguridad.

Este ejercicio práctico ha demostrado que Wazuh Cloud es una herramienta poderosa y versátil para la gestión de seguridad en entornos diversos. La plataforma no solo facilita la detección y mitigación de amenazas, sino que también proporciona una estructura clara para la administración de eventos de seguridad, mejorando la capacidad de respuesta ante incidentes.