FACULTAD DE CIENCIA Y TECNOLOGÍA

CENTRO REGIONAL DE USULUTÁN

INGENIERÍA EN SISTEMAS Y REDES INFORMÁTICAS

SEGURIDAD INFORMÁTICA

CICLO II – 2025

ACTIVIDAD:

LABORATORIO 1 COMPUTO 2 – GUÍA DE INSTALACIÓN DE WAZUH.

DOCENTE:

ING. TIMOTEA GUADALUPE MENJIVAR.

ESTUDIANTES:

ANA PATRICIA GAITÁN HERNÁNDEZ	USSS017122
FREDY ADILSON CAMPOS HERNÁNDEZ	USSS017322
JOSUÉ GABRIEL CAMPOS ESCOBAR	USIS006316
LESLY CAROLINA BERMÚDEZ MEMBREÑO	USSS017722
ROMEO ALEXANDER GARCIA CASTILLO	USIS000313

FECHA DE ENTREGA:

USULUTÁN, 30 DE OCTUBRE DE 2025.

Wazuh es una plataforma de seguridad que proporciona protección XDR y SIEM unificada para terminales y cargas de trabajo en la nube. La solución está compuesta por un único agente universal y tres componentes centrales: el servidor Wazuh, el indexador Wazuh y el panel de Wazuh.

INSTALACIÓN DE WAZUH.

Recomendaciones generales.

1. Servidor de Ubuntu server 24.04, con mínimo de almacenamiento de 80 gb. (Por cache, por logs).
2. Dirección IP estática.
3. Configuración de la interfaz gráfica.
4. Configuración del resolv.conf
5. Conexión a internet.

Configuración básica para su correcto funcionamiento.

Agentes	UPC	RAM	Almacenamiento (90 días)
1–25	4 vCPU	8 GiB	50 GB
25–50	8 vCPU	8 GiB	100 GB
50–100	8 vCPU	8 GiB	200 GB

Actualizamos el Ubuntu server, utilizando el siguiente comando.

sudo apt update && sudo apt upgrade

```
root@server01:/home/ragc# sudo apt update && sudo apt upgrade
```

```
root@server01:/home/ragc# sudo apt update && sudo apt upgrade
Hit:1 http://sv.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://sv.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://sv.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://sv.archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
```

Verificamos que el cortafuegos este activo.

ufw status numbered

```
root@server01:/home/ragc# ufw status numbered
Status: inactive
root@server01:/home/ragc#
```

Habilitamos los servicios de firewall.

ufw enable

```
root@server01:/home/ragc# ufw enable
Firewall is active and enabled on system startup
root@server01:/home/ragc#
```

Se verifica que este activo en el servidor.

ufw status numbered

```
root@server01:/home/ragc# ufw status numbered
Status: active
root@server01:/home/ragc#
```

Habilitamos los puertos necesarios de escucha con el comando:

Sudo ufw allow 22/tcp

```
root@server01:/home/ragc# sudo ufw allow 22/tcp
Rule added
Rule added (v6)
root@server01:/home/ragc# sudo ufw allow 443/tcp
Rule added
Rule added (v6)
root@server01:/home/ragc# sudo ufw allow 1514/tcp
Rule added
Rule added (v6)
root@server01:/home/ragc# sudo ufw allow 1515/tcp
Rule added
Rule added (v6)
root@server01:/home/ragc# sudo ufw allow 55000/tcp
Rule added
Rule added (v6)
root@server01:/home/ragc# sudo ufw allow 5601/tcp
Rule added
Rule added (v6)
root@server01:/home/ragc#
```

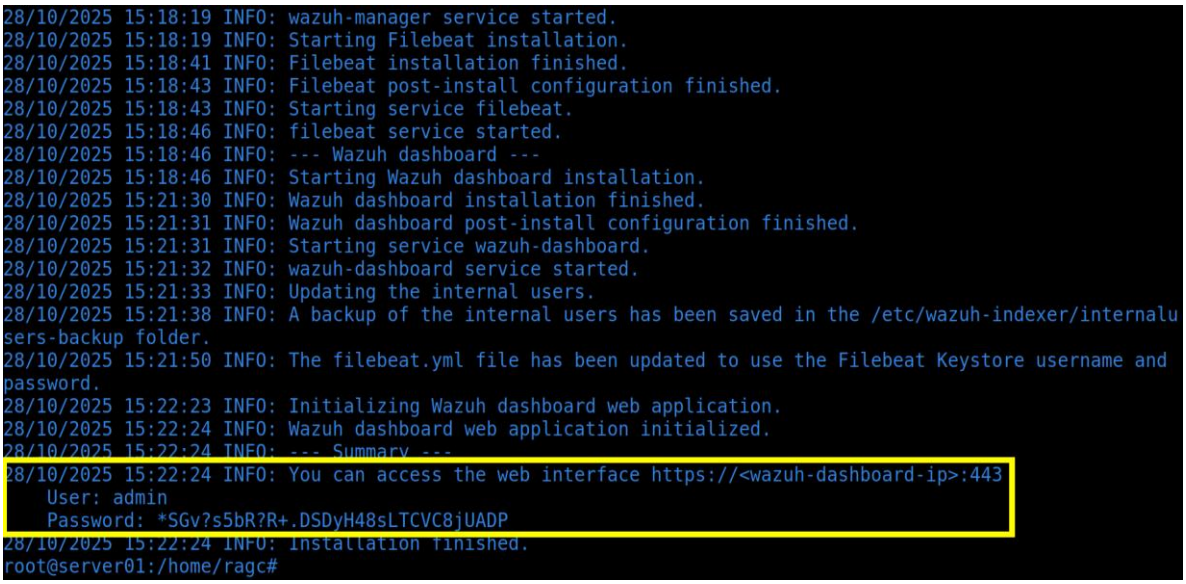
Ya que hemos activado los puertos necesarios, Podemos comenzar con la instalación de Wazuh.

Descargamos y ejecutamos el asistente de instalación de Wazuh.

curl -sO https://packages.wazuh.com/4.11/wazuh-install.sh && sudo bash ./wazuh-install.sh -a

```
root@server01:/home/ragc# curl -sO https://packages.wazuh.com/4.11/wazuh-install.sh && sudo bash ./wazuh-in
stall.sh -a
27/10/2025 22:20:12 INFO: Starting Wazuh installation assistant. Wazuh version: 4.11.2 (x86_64/AMD64)
27/10/2025 22:20:12 INFO: Verbose logging redirected to /var/log/wazuh-install.log
27/10/2025 22:20:16 INFO: Verifying that your system meets the recommended minimum hardware requirements.
27/10/2025 22:20:16 INFO: Wazuh web interface port will be 443.
27/10/2025 22:20:16 WARNING: The system has UFW enabled. Please ensure that traffic is allowed on these por
ts: 1515, 1514, 443.
```

Este proceso tarda bastante, debemos recordar que si no contamos con los requerimientos necesarios no se instalará, posteriormente mostrará el medio de acceso a la interfaz web de Wazuh, algo que es bueno mencionar es que debemos percatarnos de no dejar salto de línea en el comando anterior, porque de ser así no se instalará.



En este caso la rut del dashboard, usuario y contraseña:

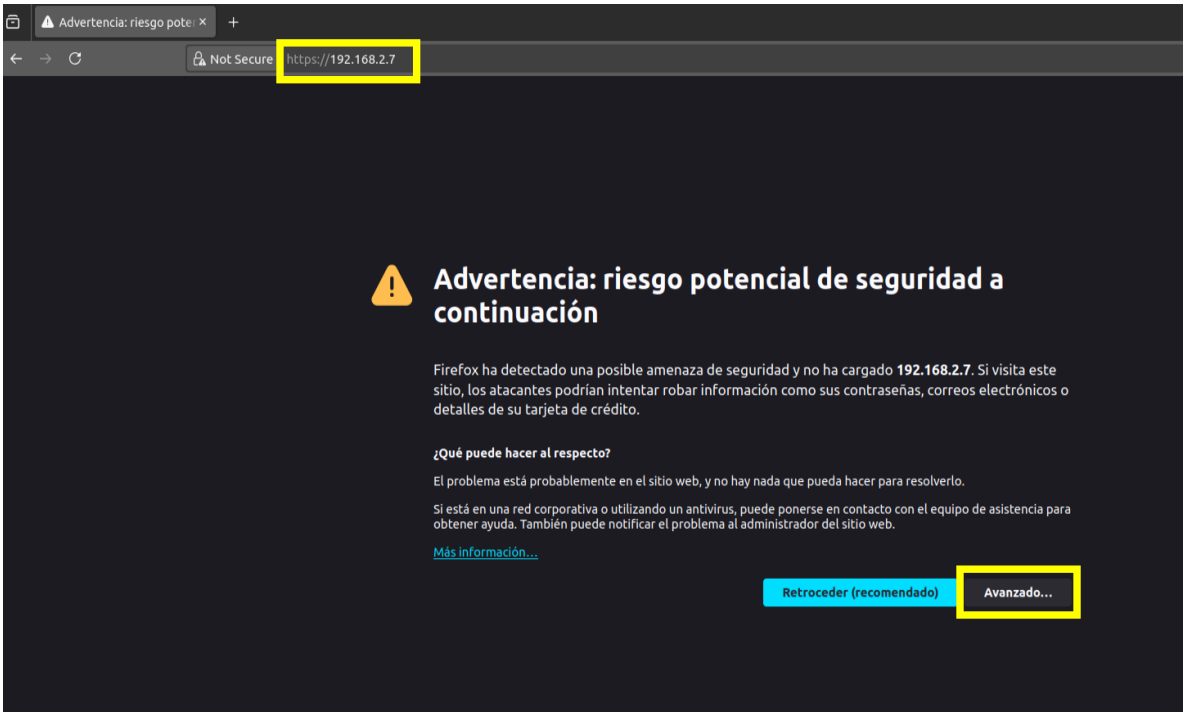
https://<wazuh-dashboard-ip>:443
User: admin
Password: *SGv?s5bR?R+.DSDyH48sLTCVC8jUADP

Hay que tomar en cuenta que estos datos varían, así que debemos guardar los que proporciona wazuh en la instalación que realicemos.

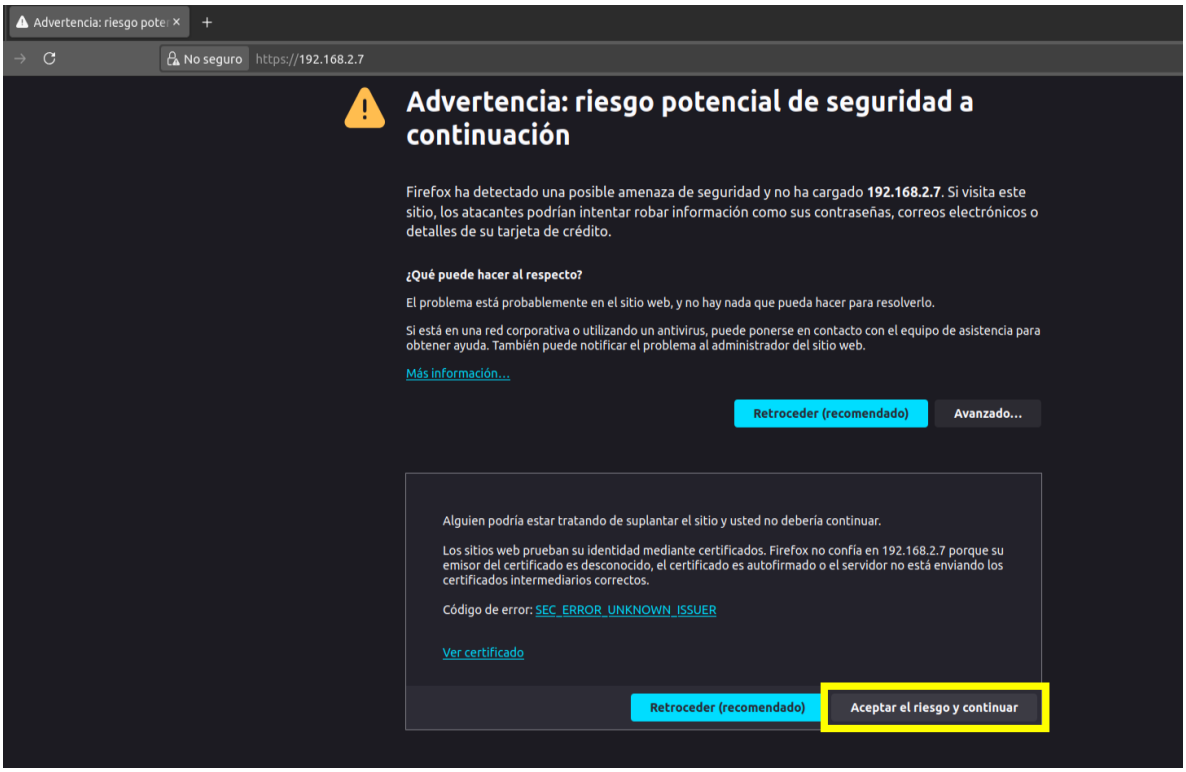
CARGANDO EL DASHBOARD.

Debemos poner la ip de Nuestro servidor en el navegador, en este caso es la 192.168.2.7, se colocará de la siguiente manera:

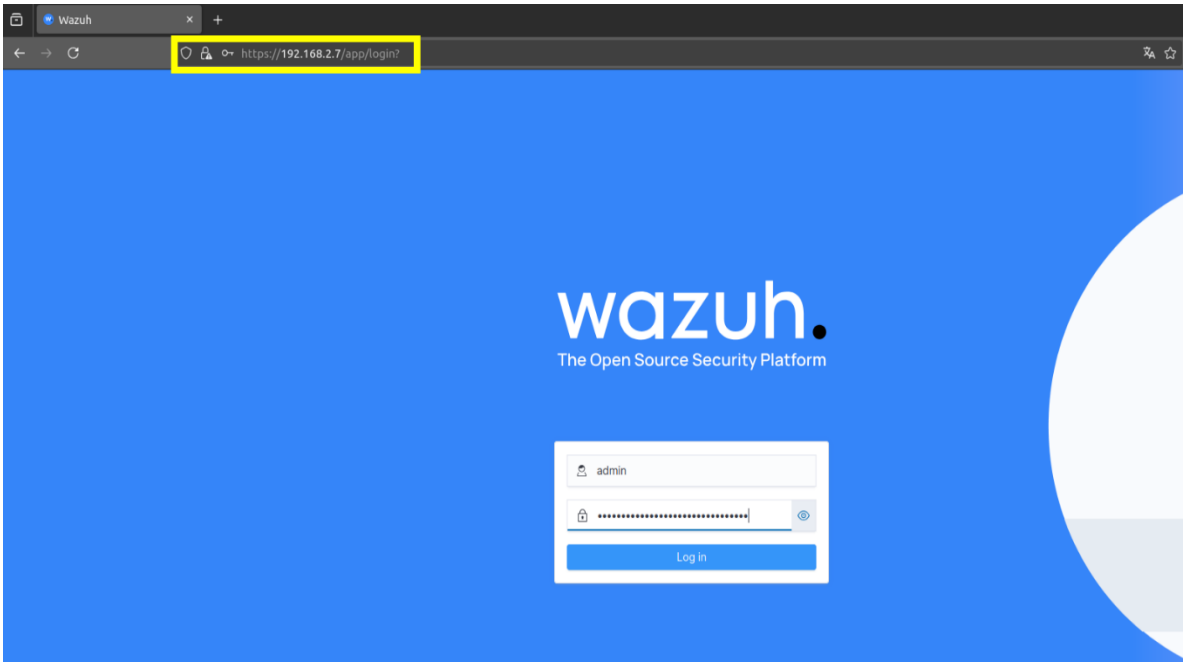
https:/192.168.2.7:443



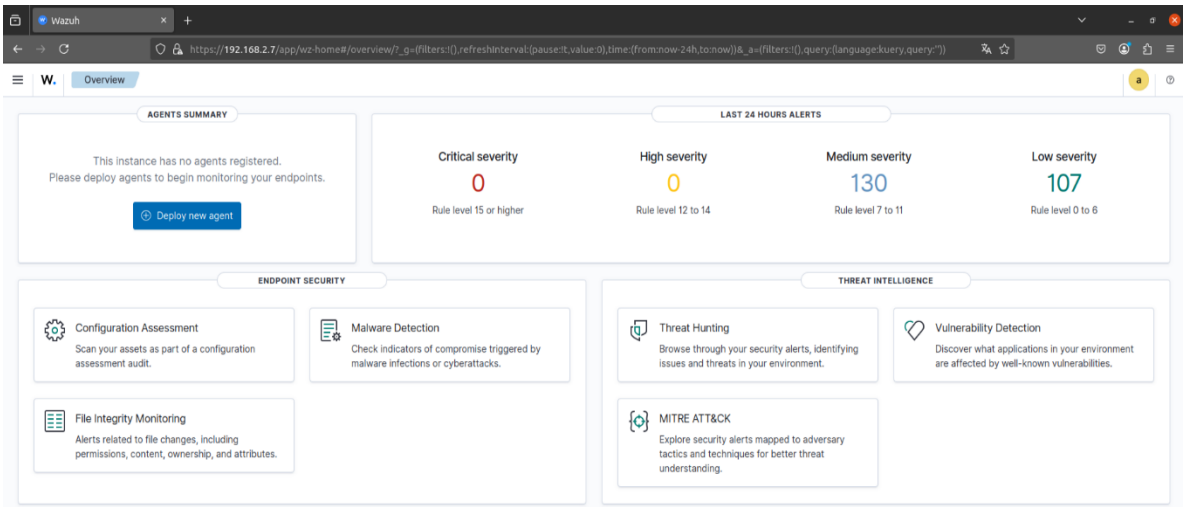
Debemos seleccionar la opción que dice “Avanzado”, luego nos mostrará la siguiente ventana advirtiendo el riesgo de seguir, seleccionamos la opción “Aceptar el riesgo y continuar”.



Una vez que inicie, nos pedirá loguearnos, para ello debemos ingresar los datos proporcionados anteriormente por la herramienta.



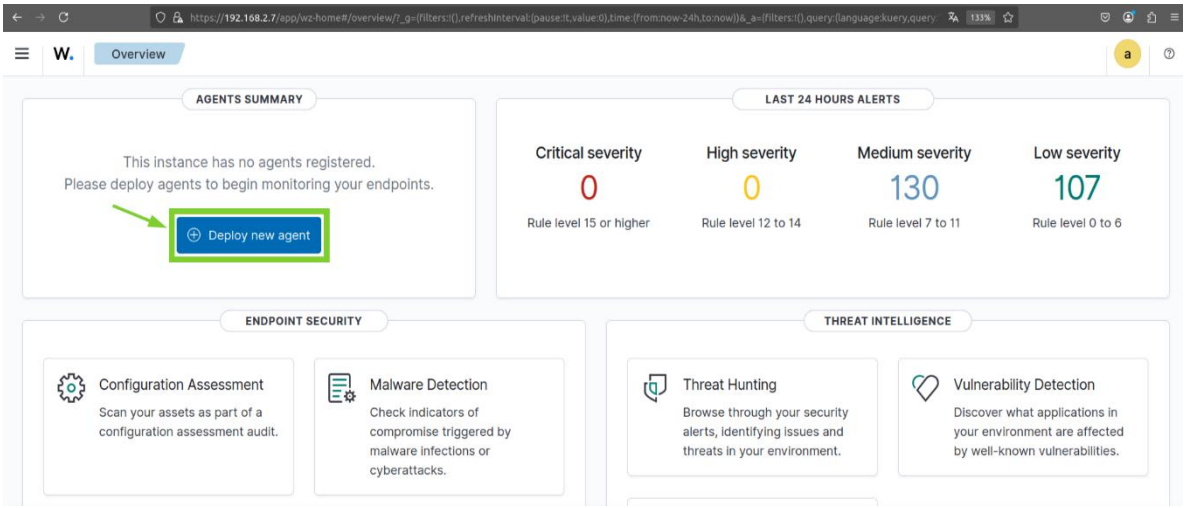
Finalmente, tenemos acceso al panel de administración:



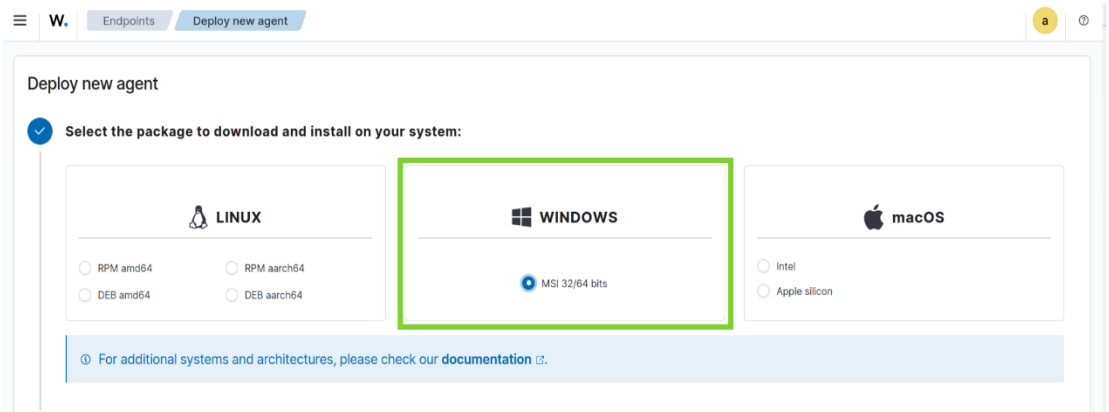
AGREGANDO UN NUEVO AGENTE.

Agregando un agente Windows.

Seleccionamos la opción “Desplegar nuevo agente”.



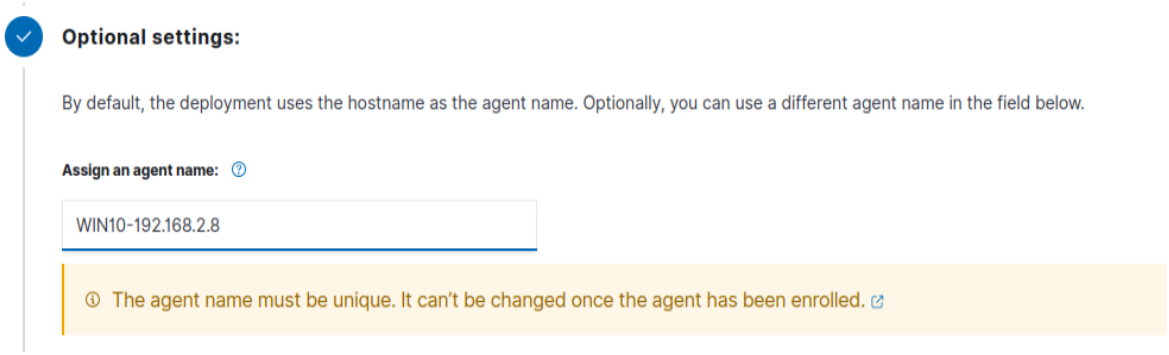
Seleccionamos la distribución del nuevo agente, en este caso será windows.



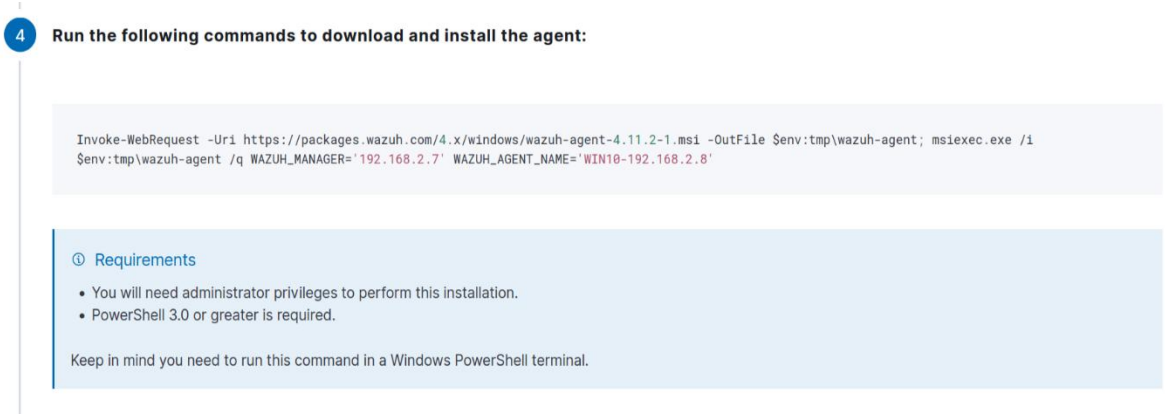
Llenamos los datos del servidor:

Asignamos un nombre al agente windows, uno que nos permita reconocerlo fácilmente y además de su ip estática que le hemos configurado, siendo este el nombre:

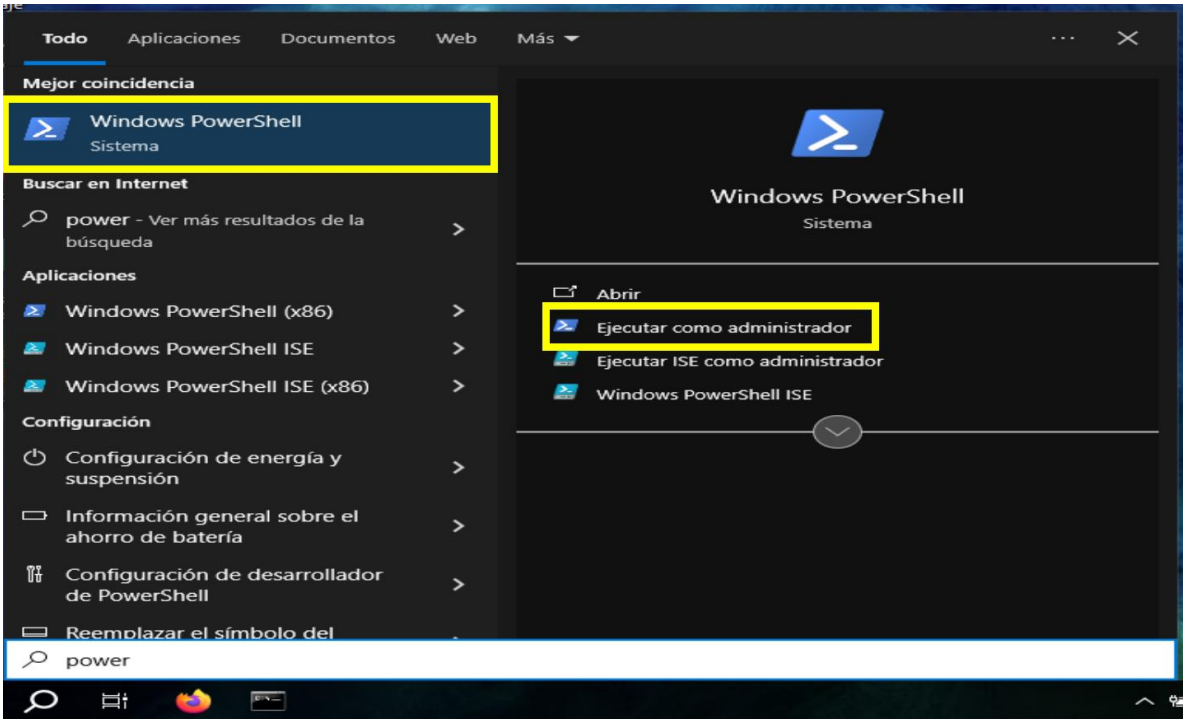
WIN10-192.168.2.8



Lo siguiente es copiar el código de configuración que debemos ejecutar en powershell en la máquina cliente, es decir en el agente windows.

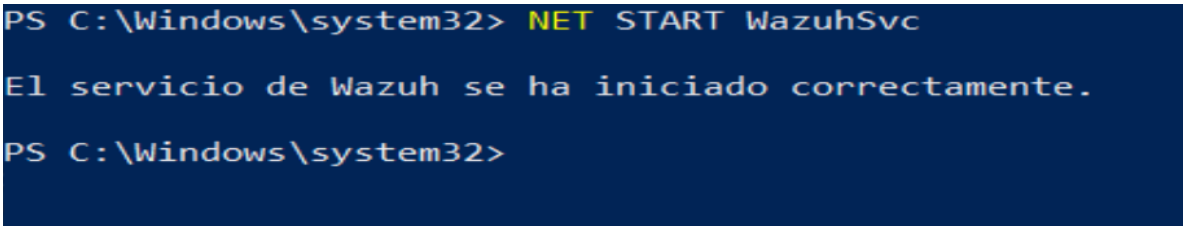


El último paso es abrir el powershell de la máquina windows como administrador.

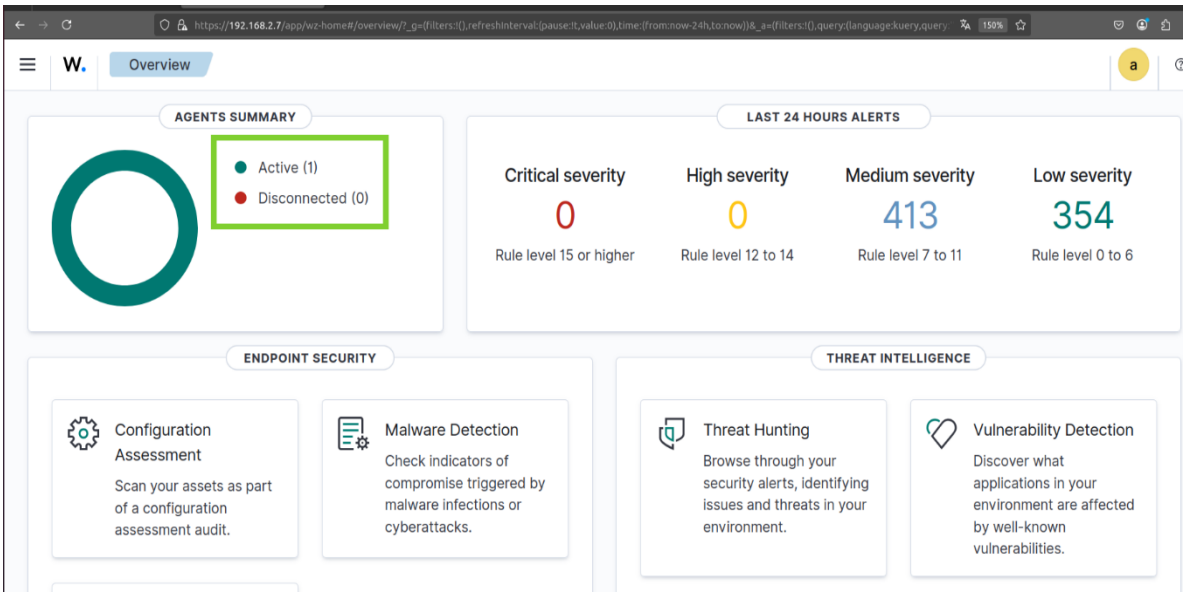


Ahora ejecutamos el comando de inicio del agente:

NET START WazuhSvc



Podemos notar que el agente se instaló correctamente:

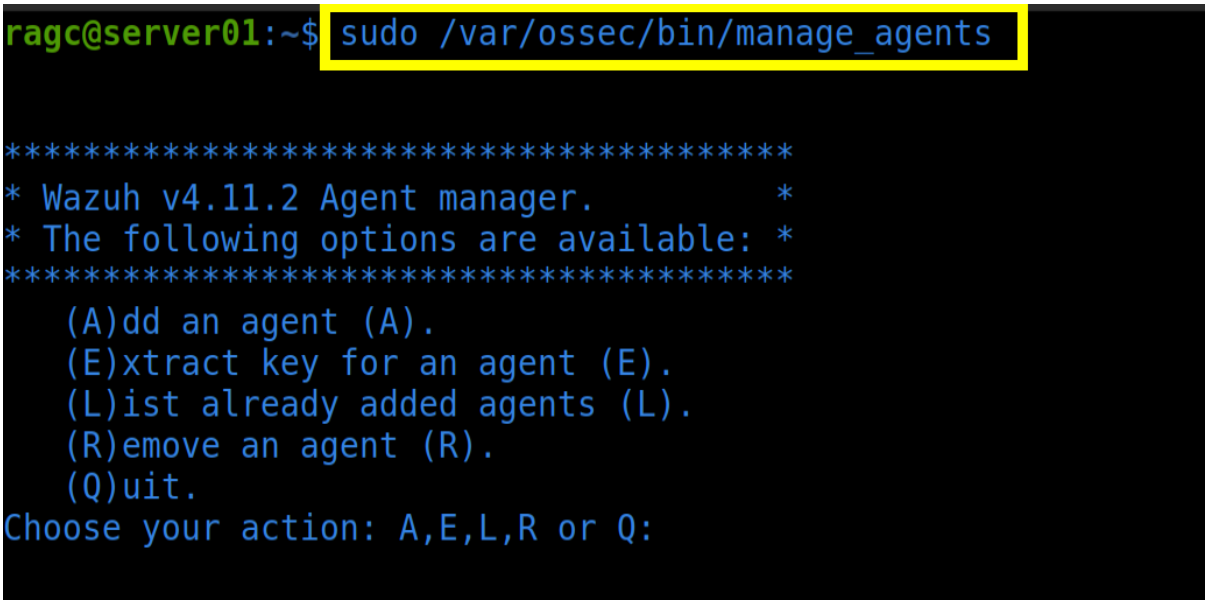


Método alternativo si no aparece después del comando: NET START WazuhSvc:

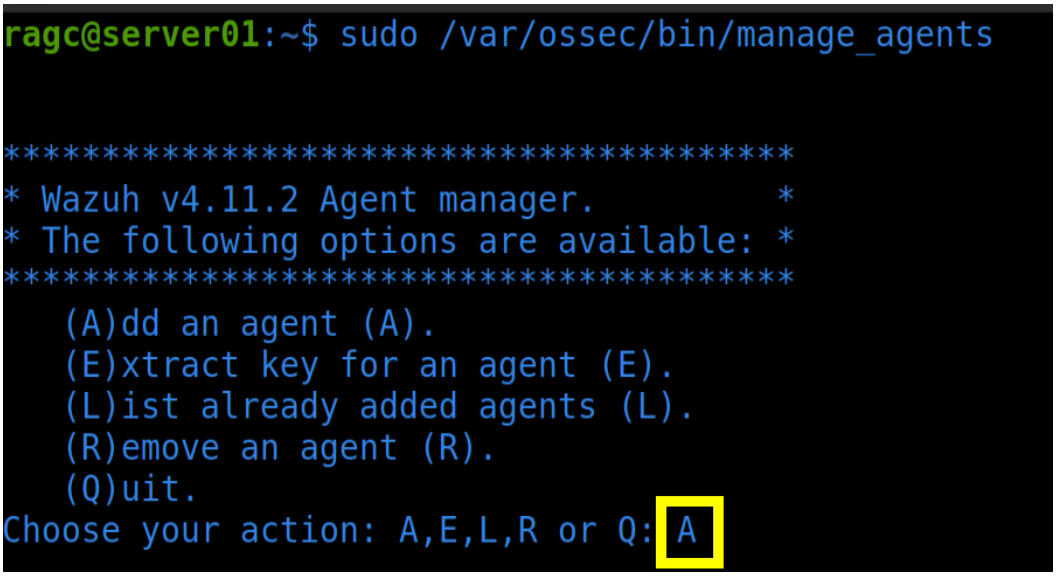
Si no nos aparece hasta este punto el agente, entonces debemos agregarlo con la terminal de nuestro Ubuntu server.

Para ello debemos abrir nuestra terminal y escribir el siguiente comando:

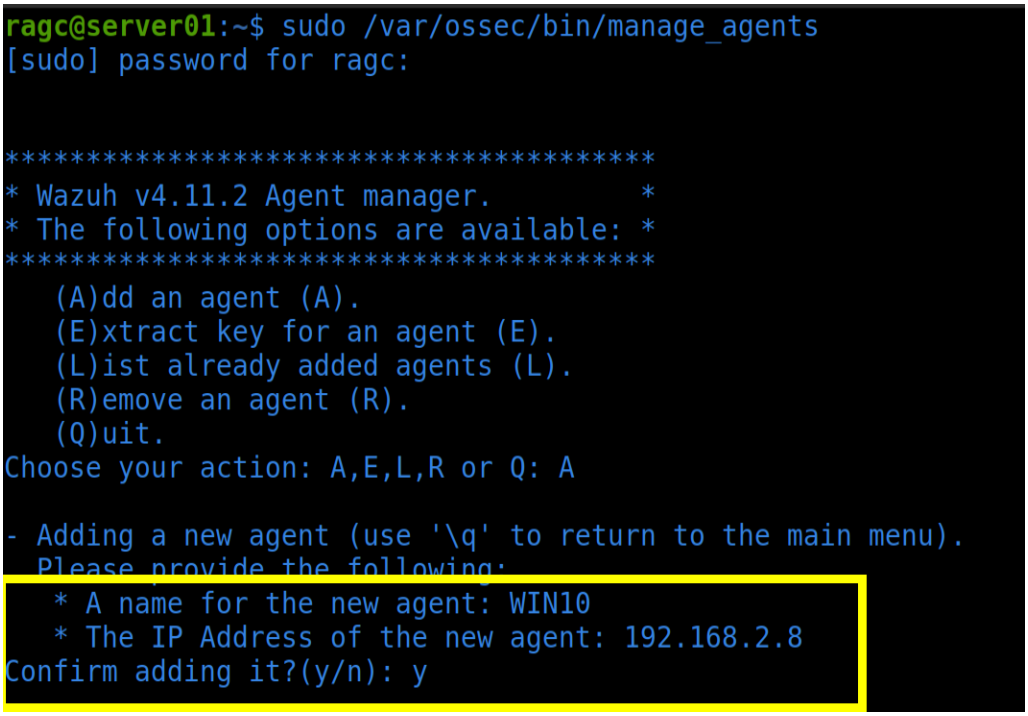
sudo /var/ossec/bin/manage_agents



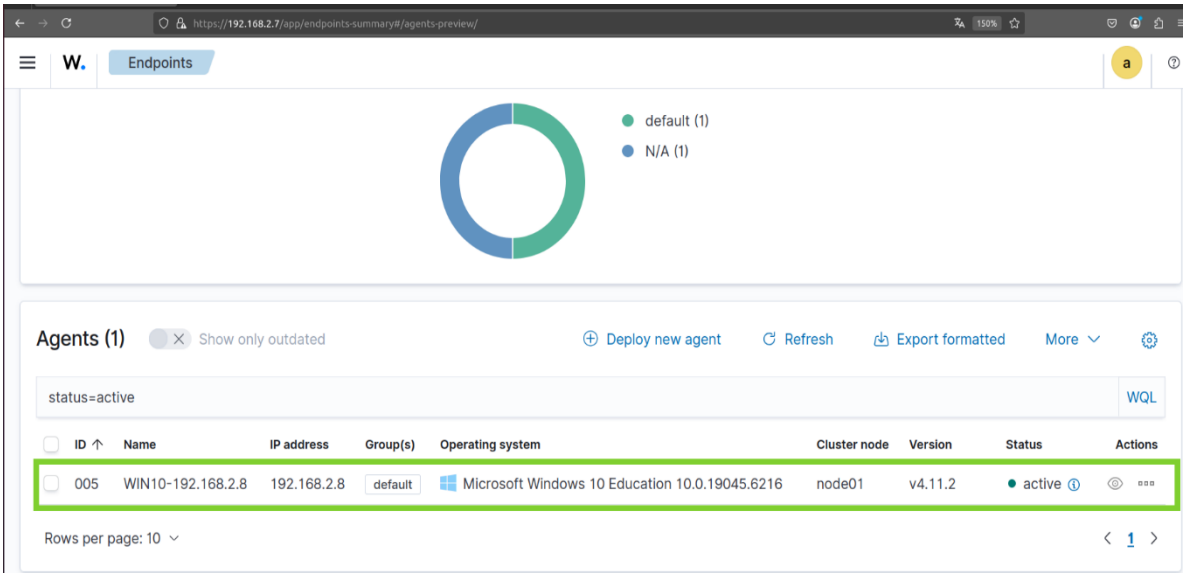
Se abre un menú de opciones y debemos seleccionar la letra “A” (Add), para agregar el agente:



Nos pedirá el nombre de nuestro agente, para lo cual le pondremos WIN10 y la ip del agente, que en este caso es la 192.168.2.8 (máquina windows) y por último confirmamos el hecho de agregar el agente con la letra “y”, damos enter y listo hemos agregado el agente windows:



Ya Podremos notar que se ha agregado el agente:



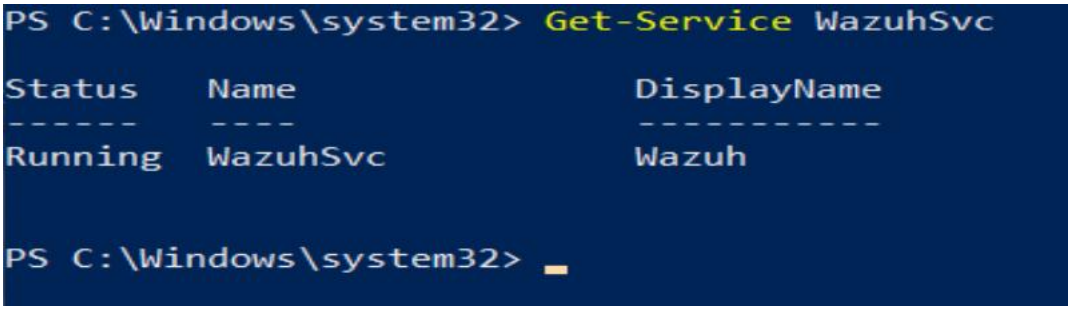
Pasos que debemos tener en cuenta antes de apagar nuestro agente windows.

PASO 1:

Verifica que el servicio esté en buen estado (opcional):

En PowerShell (Administrador):

Get-Service WazuhSvc

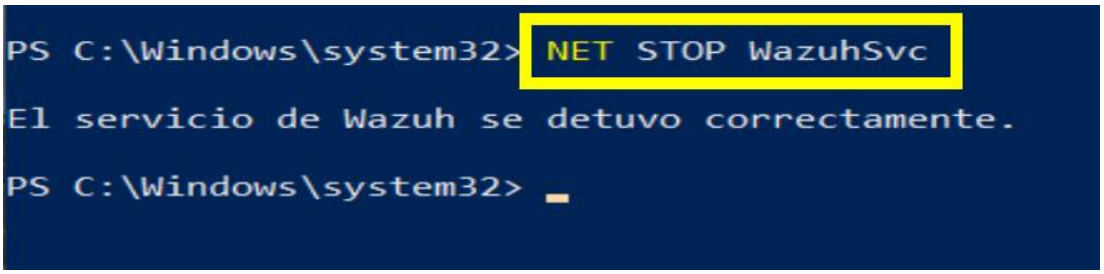


Si dice Running, está funcionando correctamente.

PASO 2:

Detenemos el servicio manualmente. Utilizamos el comando, de esta forma desconectamos limpiamente el agente del servidor.

NET STOP WazuhSvc

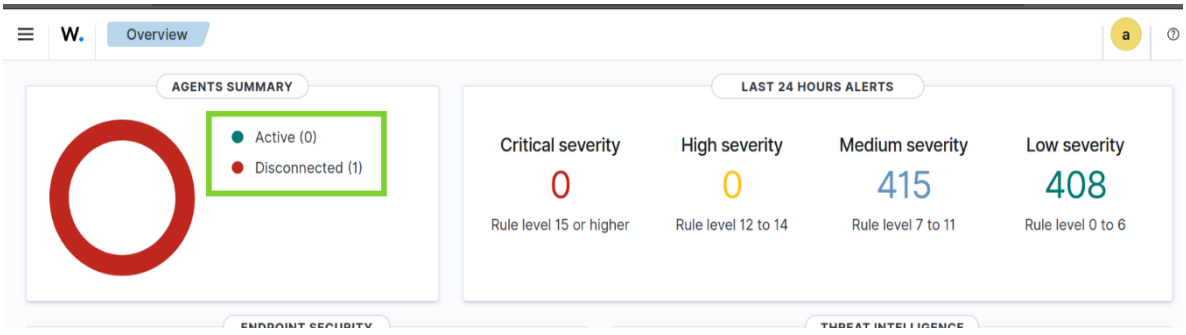


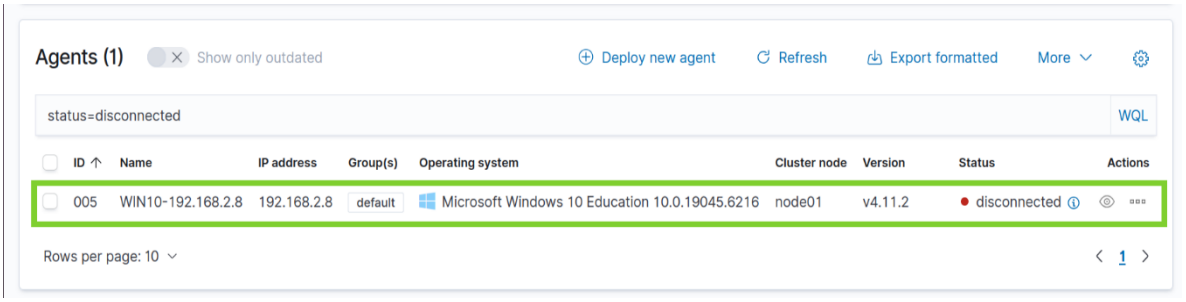
PASO 3:

Apagamos windows normalmente:

shutdown /s /t 0

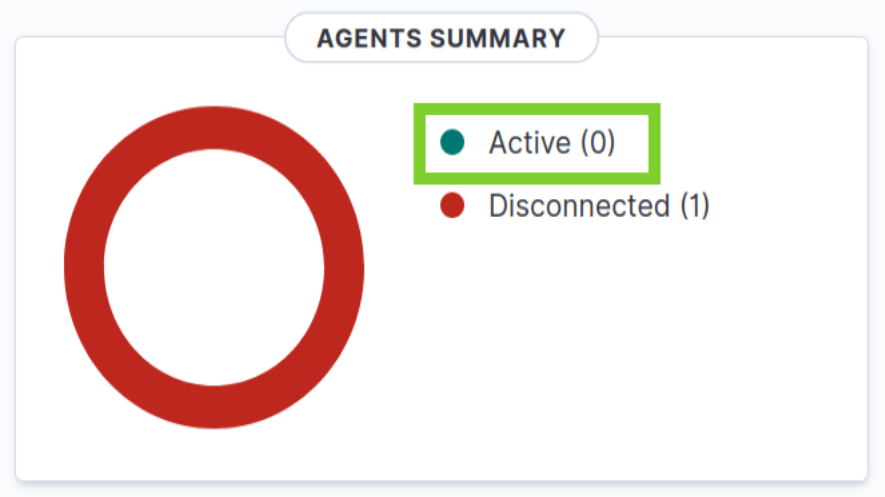
Notaremos que ahora no aparece la máquina windows 10 en “Active”, sino en “Disconected(1)”.



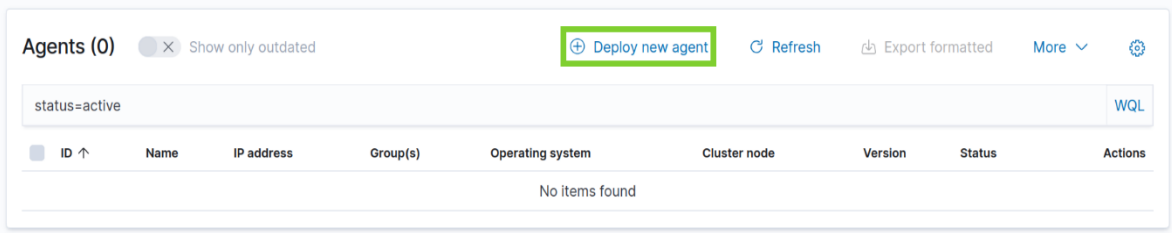


Agregando un agente Libre.

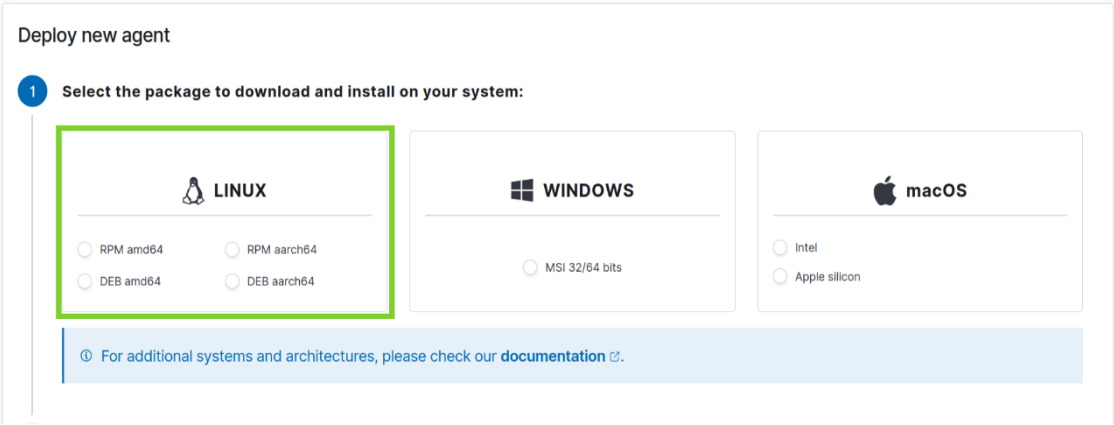
Debemos dar clic en la parte de Wazuh, donde dice “Active”, esto nos mostrará el panel de los agentes activos.



Una vez estando en ese panel, debemos dar clic en “Deploy new agent”, con el objetivo de agregar un nuevo agente.



Nos desplegará el panel donde vamos a seleccionar el tipo de arquitectura que usaremos, en este caso es un S.O libre. Por lo tanto usamos esa opción:



Ahora debemos saber la arquitectura de nuestro sistema operativo para que podamos elegir la opción correcta, para ello en la terminal de nuestro agente debian 13 (agente), vamos a poner el siguiente comando:

uname -a

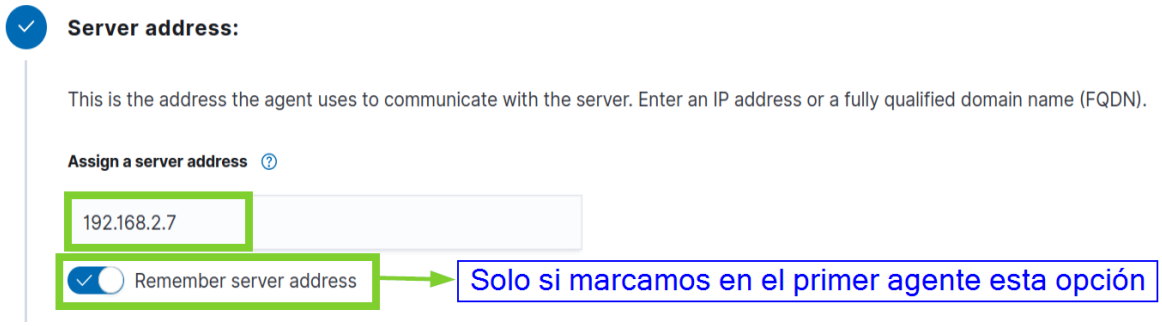
```
ragc@romeoDebian:~$ uname -a
Linux romeoDebian 6.12.48+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.48
-1 (2025-09-20) x86_64 GNU/Linux
ragc@romeoDebian:~$
```

Aquí lo importante es ver la parte de la arquitectura:

```
ragc@romeoDebian:~$ uname -a
Linux romeoDebian 6.12.48+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.48
-1 (2025-09-20) x86_64 GNU/Linux
ragc@romeoDebian:~$
```

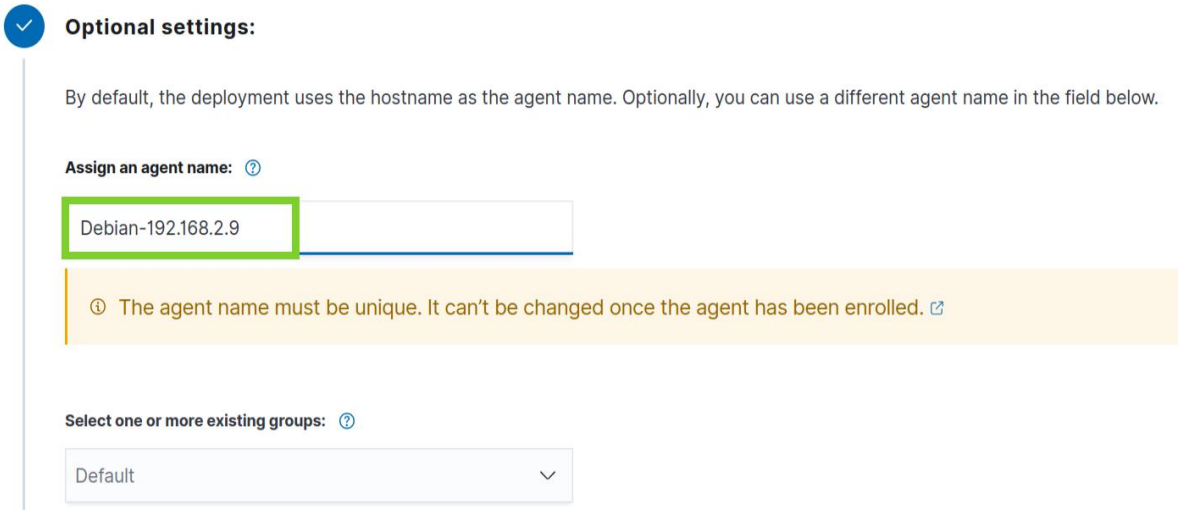
Ahora que ya tenemos la arquitectura de nuestro debian, podemos seleccionar la opción correcta en nuestro Wazuh:

Una vez seleccionada la opción correcta, pasamos a la siguiente opción, aunque debemos aclarar que, ya que es el segundo agente, entonces automáticamente estará disponible la ip de nuestro servidor.



El siguiente paso es darle un nombre con el que podamos identificar fácilmente la máquina, en este caso le pondremos:

Debian-192.168.2.9



Ahora lo siguiente, antes de pasar a pegar el comando de descarga e instalación del agente, debemos instalar las siguientes dependencias en Debian 11, 12 y 13 con el objetivo de obtener primero las llaves que el sistema necesita para poder **descargar y verificar con seguridad** el paquete del agente Wazuh desde internet.

Los comandos de las dependencias son, aunque es posible que ya esten instalados por defecto en el sistema:

sudo apt-get install gnupg apt-transport-https -y

```
ragc@romeoDebian:~$ sudo apt-get install gnupg apt-transport-https -y
[sudo] contraseña para ragc:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
gnupg ya está en su versión más reciente (2.4.7-21).
apt-transport-https ya está en su versión más reciente (3.0.3).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
ragc@romeoDebian:~$
```

sudo apt-get install curl -y

```
ragc@romeoDebian:~$ sudo apt-get install curl -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
curl ya está en su versión más reciente (8.14.1-2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
ragc@romeoDebian:~$
```

Ahora copiamos el comando de instalación del agente y lo pegamos en la terminal de debian 13.


```
ragc@romeoDebian:~$ sudo wget https://packages.wazuh.com/4.x/apt/pool/main/w/waz
uh-agent/wazuh-agent_4.11.2-1_amd64.deb && sudo WAZUH_MANAGER='192.168.2.7' WAZU
H_AGENT_NAME='Debian-192.168.2.9' dpkg -i ./wazuh-agent_4.11.2-1_amd64.deb
[sudo] contraseña para ragc:
```

```
[sudo] contraseña para ragc:
--2025-10-29 20:24:29--  https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-ag
ent/wazuh-agent_4.11.2-1_amd64.deb
Resolviendo packages.wazuh.com (packages.wazuh.com)... 18.173.166.9, 18.173.166.
107, 18.173.166.17, ...
Conectando con packages.wazuh.com (packages.wazuh.com)[18.173.166.9]:443... cone
ctado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 11075686 (11M) [application/vnd.debian.binary-package]
Grabando a: «wazuh-agent_4.11.2-1_amd64.deb»

wazuh-agent_4.11.2- 100%[===================>]  10.56M  15.3MB/s    en 0.7s

2025-10-29 20:24:30 (15.3 MB/s) - «wazuh-agent_4.11.2-1_amd64.deb» guardado [110
75686/11075686]

Seleccionando el paquete wazuh-agent previamente no seleccionado.
(Leyendo la base de datos ... 142593 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar .../wazuh-agent_4.11.2-1_amd64.deb ...
Desempaquetando wazuh-agent (4.11.2-1) ...
Configurando wazuh-agent (4.11.2-1) ...
ragc@romeoDebian:~$
```

Una vez configurado lo anterior, debemos ingresar los siguientes comandos para el inicio del agente.

PRIMER COMANDO:

sudo systemctl daemon-reload

Este comando nos permite recargar la lista de servidores para detectar el nuevo agente wazuh que acabamos de instalar.

```
ragc@romeoDebian:~$ sudo systemctl daemon-reload
ragc@romeoDebian:~$
```

SEGUNDO COMANDO:

sudo systemctl enable wazuh-agent

Este comando nos permite ejecutar automáticamente cada vez que encienda el debian.

```
ragc@romeoDebian:~$ sudo systemctl enable wazuh-agent
Created symlink '/etc/systemd/system/multi-user.target.wants/wazuh-agent.service
' → '/usr/lib/systemd/system/wazuh-agent.service'.
ragc@romeoDebian:~$
```

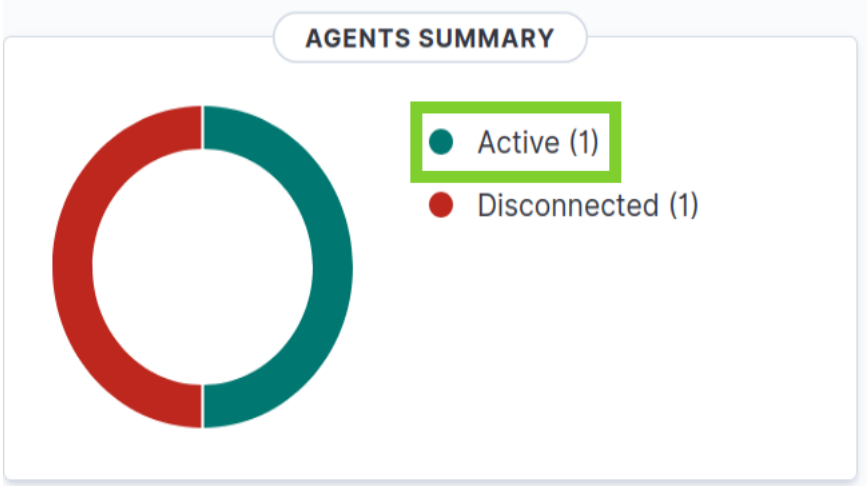
TERCER COMANDO:

sudo systemctl start wazuh-agent

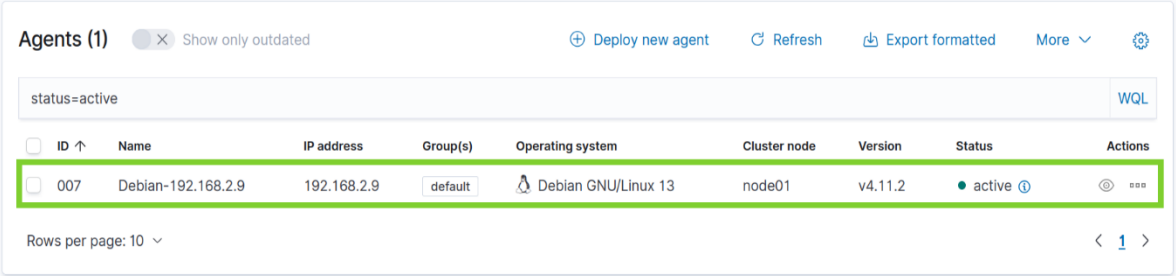
Este comando enciende el servicio inmediatamente, sin necesidad de reiniciar.

```
ragc@romeoDebian:~$ sudo systemctl start wazuh-agent
ragc@romeoDebian:~$
```

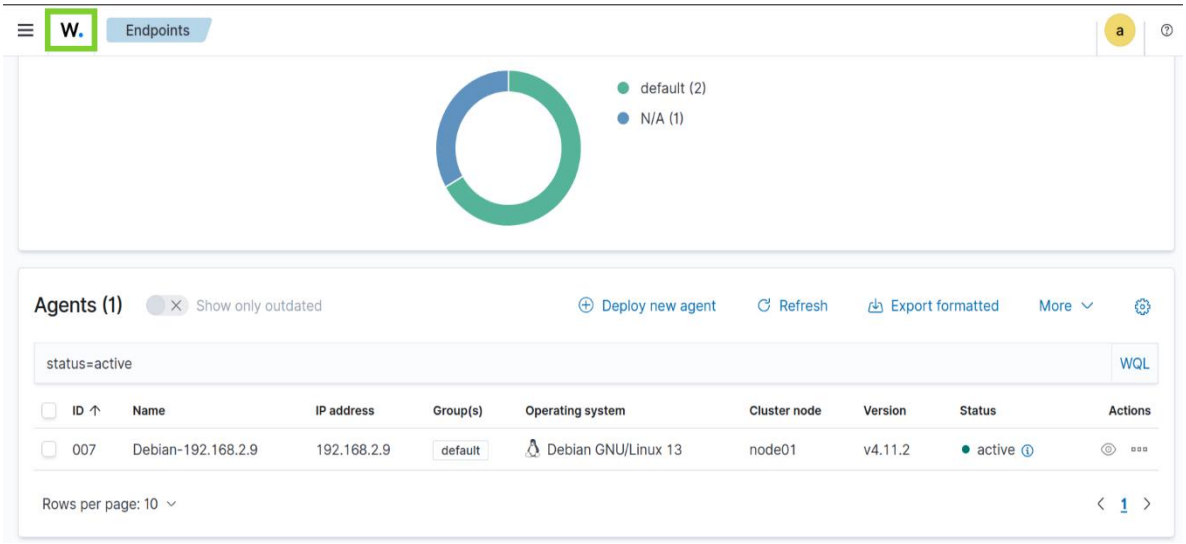
Ahora solo falta recargar el dashboard de wazuh para que se cargue el nuevo agente y el estado activo del mismo, notaremos que ya nos muestra uno activo y uno desectivado, recordemos que solo teníamos conectado el agente windows:



Damos clic en el que dice “Active” y este nos llevara a la ventana donde están todos los agentes activos, pero que en este momento solo es uno, mostrándonos el agente libre, el cual es Debian 13.



Si queremos volver a la página principal de Wazuh, basta con presionar el botón superior izquierdo que tiene una “W”.



MÓDULOS DE WAZUH.

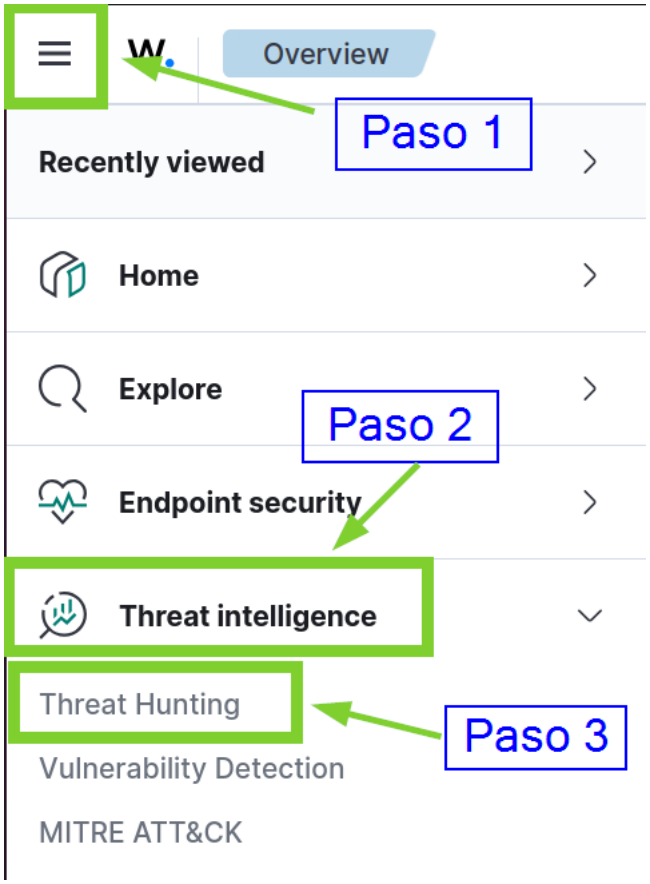
Módulo 1 – Análisis de logs (recolección y normalización).

Lo realiza el agente y el manager al recibir los logs. Los decodificadores (decoders) y las reglas que transforman y normalizan los eventos, están en el conjunto de reglas (ruleset).

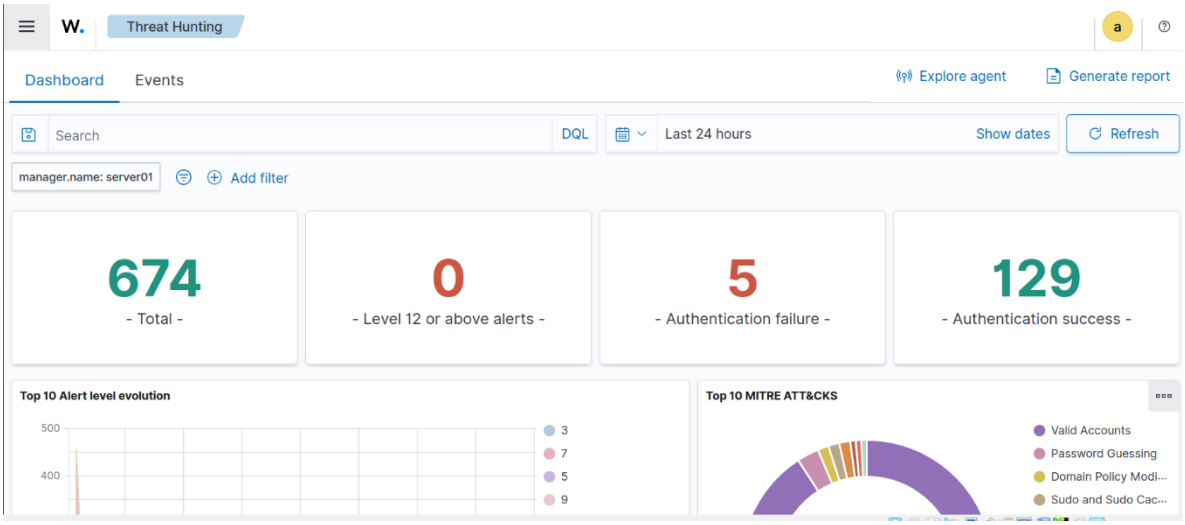
En resumen, este módulo se encarga de recolectar, normalizar y almacenar los logs que envían los agentes.

Pasos para entrar al módulo 1 (Análisis de logs).

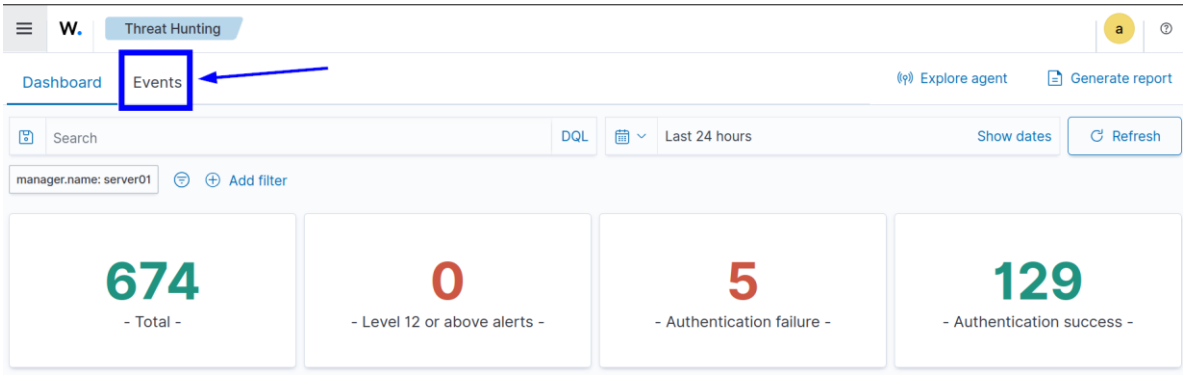
Primero abrimos el panel, dando clic en el botón superior izquierdo. Luego buscamos el módulo Treat Hunting, Aquí es donde se visualizan los resultados combinados del **Módulo 1 (Análisis de logs)**, **Módulo 2 (Detección de intrusos)** y **Módulo 3 (Análisis de seguridad).**



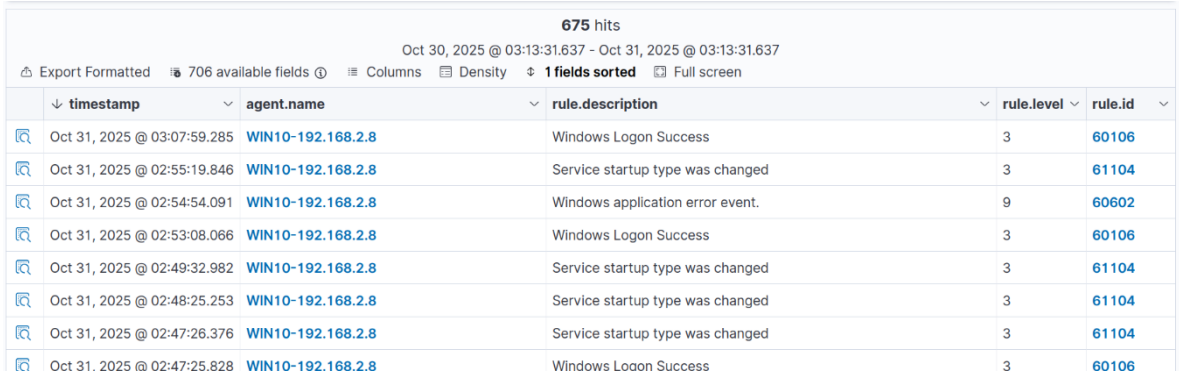
Ahora, nos mostrará el siguiente panel:



Lo que debemos hacer ahora es dar clic en la etiqueta que dice “Eventos”. Ahí podremos ver los logs que Wazuh esta recibiendo de los agentes.



Al dar clic ahí podremos ver los logs que son enviados a Wazuh.



Módulo 2 – Detección de intrusos.

El Módulo 2 de Wazuh se centra en la detección de intrusos dentro del sistema y la red. Su función principal es identificar actividades sospechosas o maliciosas mediante el análisis continuo de los registros y eventos que provienen de los agentes instalados en los equipos monitorizados.

En resumen, este módulo es la alarma del sistema:

- Analiza continuamente los registros de los agentes.
- Detecta patrones de intrusión conocidos y comportamientos anómalos.
- Correlaciona eventos para detectar ataques complejos.
- Ayuda a responder rápidamente ante incidentes de seguridad.

En otras palabras, se encarga de analizar los logs que ya procesó el **Módulo 1**, compararlos con sus **reglas de detección**, y generar alertas según el nivel de riesgo.

Pasos para entrar al módulo 2 (Detección de intrusos).

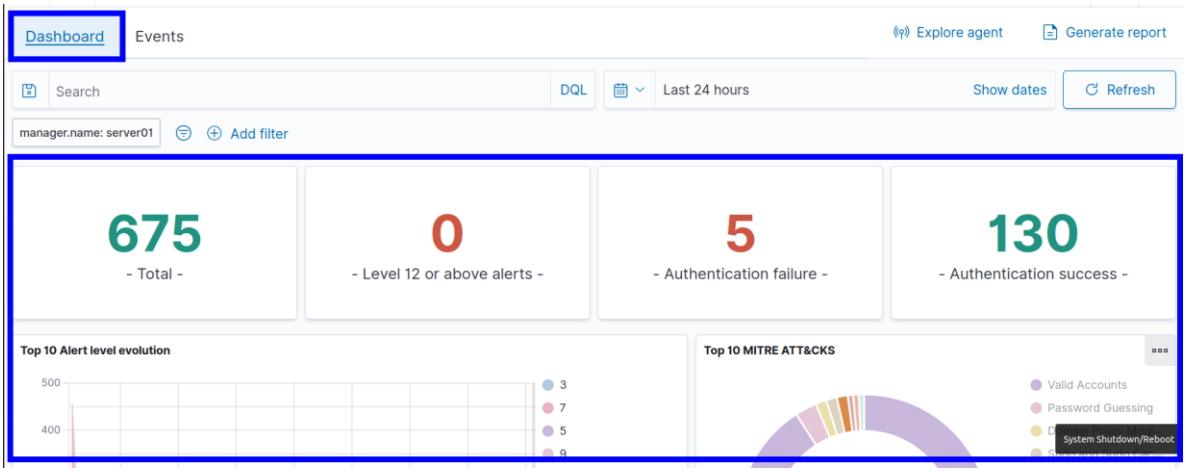
Nos quedamos siempre en el apartado del módulo 1, es decir en la ruta:

Threat Hunting → Dashboard.

Ahí debemos ver los cuadros que dicen:

- Authentication success.
- Authentication failure.
- Level 12 or above alerts
- Total.

Tal como podemos ver en la imagen:



Si nosotros hacemos clic en un número o gráfico, Wazuh abre los eventos que dispararon esas alertas.

Módulo 3 – Análisis de seguridad.

Se enfoca en el **análisis de seguridad integral** del sistema. Su propósito es **evaluar el estado general de la seguridad** de los equipos y servidores monitorizados, verificando configuraciones, políticas, vulnerabilidades y cumplimiento de normas.

Este módulo **recopila y correlaciona los datos de los agentes** para ofrecer una visión completa del nivel de protección y los riesgos existentes.

El análisis de seguridad ayuda a los administradores a **prevenir ataques antes de que ocurran**, reforzando las defensas del sistema mediante acciones correctivas basadas en los reportes generados por Wazuh.

En resumen, este módulo actúa como un auditor de seguridad automatizado que revisa la postura general de los sistemas:

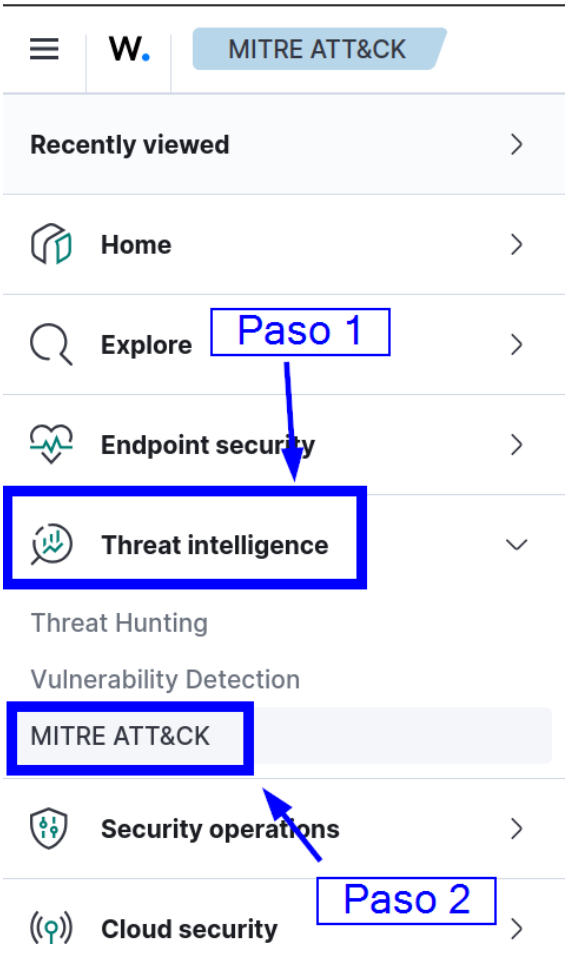
- Evalúa configuraciones y políticas de seguridad.
- Detecta vulnerabilidades y fallos en el sistema.
- Correlaciona eventos para medir el nivel de riesgo.
- Verifica el cumplimiento de estándares y normativas.

Evalúa vulnerabilidades y cumplimiento normativo.

Pasos para entrar al módulo 3 – Análisis de seguridad.

Debemos ir a la ruta:

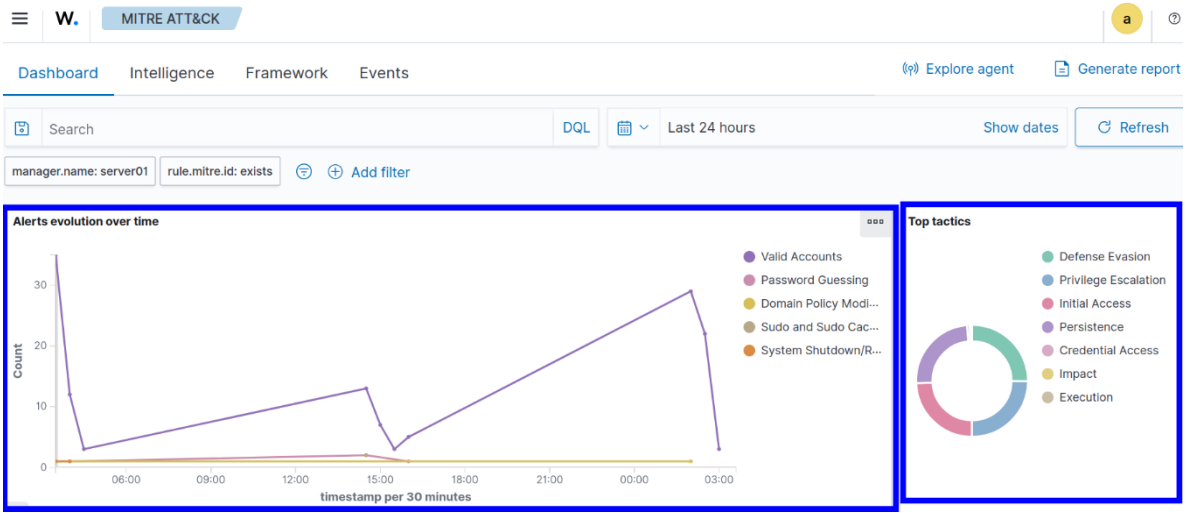
Threat Intelligence → MITRE ATT&CK.



El **gráfico de la izquierda** (evolución) muestra **cuántas alertas** relacionadas con cada *técnica* ocurrieron a lo largo del tiempo (por ejemplo *Valid Accounts*, *Password Guessing*).

El gráfico circular / Top tactics muestra las tácticas (ej. Credential Access, Privilege Escalation) y la proporción de alertas asociadas.

Estos datos vienen de las reglas que incluyen campos rule.mitre.* (p. ej. rule.mitre.id, rule.mitre.tactic, rule.mitre.technique) — por eso ves rule.mitre.id: exists en el filtro.



Módulo 4 – Detección de vulnerabilidades.

Este módulo analiza los paquetes y programas instalados en tus agentes (Windows, Linux, etc.) y los compara con la base de datos de vulnerabilidades (CVE). Cuando detecta que un software tiene una versión vulnerable, genera una alerta.

Por ejemplo si nuestro agente tiene *OpenSSL 1.0.2* y existe una CVE crítica para esa versión, Wazuh mostrará esa vulnerabilidad aquí, con su nivel de riesgo.

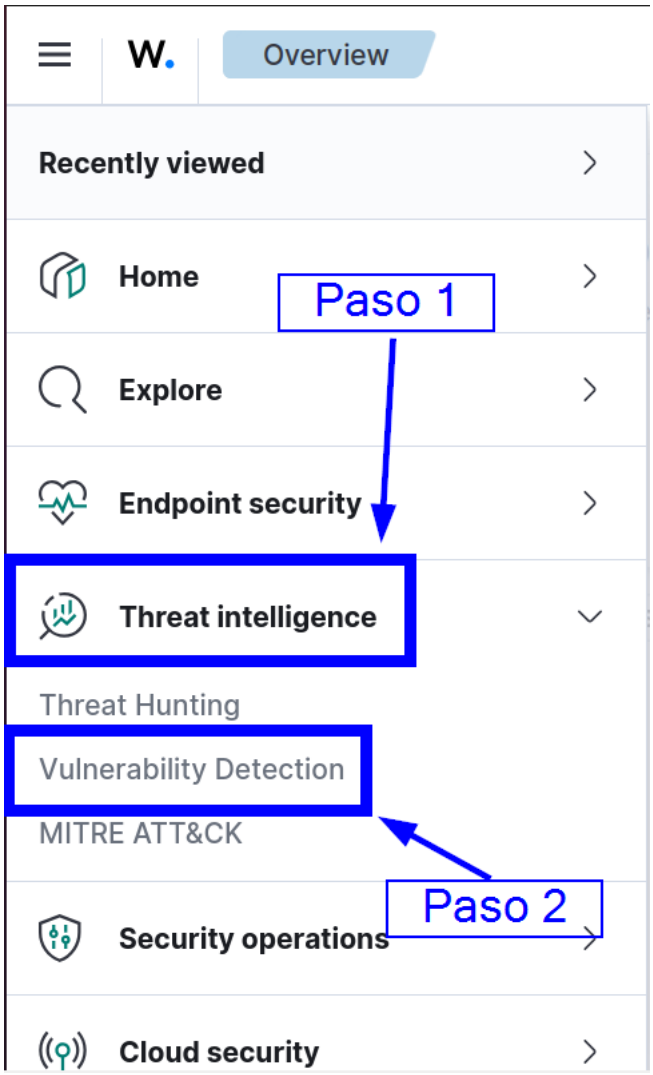
Aquí podemos ver:

- Listado de vulnerabilidades detectadas.
- CVE ID (por ejemplo, CVE-2024-1234).
- Severidad (Critical, High, Medium, Low).
- Nombre del paquete afectado y su versión.

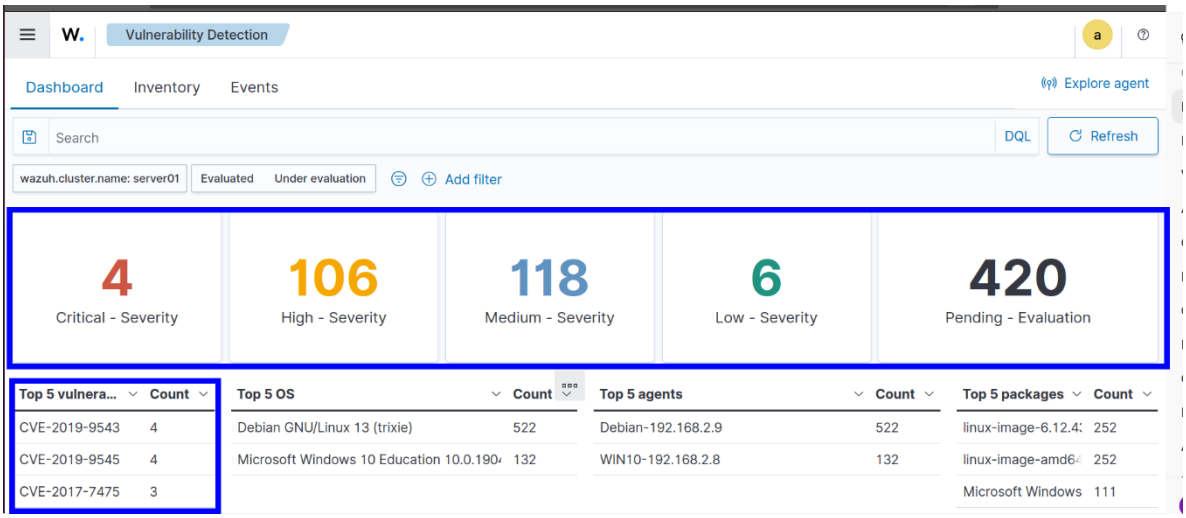
Pasos para entrar al módulo 4 – Detección de vulnerabilidades.

En el menú lateral, debemos seleccionar:

Threat Intelligence → Vulnerability Detection.



Nos abrirá el dashboard donde podremos ver todas las vulnerabilidades detectadas en los agentes, desde las más leves hasta las más graves.



Módulo 5 – Respuesta ante incidentes.

El módulo Active Response ejecuta acciones inmediatas y automáticas cuando se genera una alerta o se detecta un comportamiento peligroso.

Estas acciones pueden incluir:

- Bloquear una IP atacante (por ejemplo, tras un ataque de fuerza bruta).
- Eliminar procesos maliciosos.
- Desconectar un usuario o detener un servicio sospechoso.
- Restaurar configuraciones seguras automáticamente.

Estas respuestas se definen mediante scripts de acción almacenados en el servidor Wazuh (por defecto en /var/ossec/active-response/bin/).