

Incidentes de ciberseguridad

SOCwAttack Lab

Hecho por Rubén

INDICE:

Punto 0. AVISO IMPORTANTE.	2
1. Declaración general	2
2. Ámbito y límites	2
3. Naturaleza de las acciones realizadas	2
Punto 1. Configuración del laboratorio.	3
Punto 1º Configuración de las máquinas virtuales.	4
Punto 1.1 Configuración de la red del entorno.	4
Punto 1.1.2. Configuración del directorio de vm.	5
Punto 1.2 Soc.Lab.	6
Punto 1.3 Máquina virtual victim.lab	9
Punto 2º Instalación y configuración del SOC (Wazuh + ELK)	10
④ Ejemplo práctico	11
Punto 2.1 Wazuh	12
Paso 2.1.2 Creación de usuario.	18
Punto 2.2 Configuración de los agentes.	28
Punto 2.3 Información de los agentes.	33
Punto 3º Pruebas de monitorización y detección de incidentes.	36
Declaración de responsabilidad, alcance y limitaciones	36
Objetivo de este apartado	37
Paso 3.1 Asegurar que Victim.lab genera logs de error.	40
Paso 3.1.1 RDP.	40
Paso 3.1.2 Ataque y resultados.	41
Paso 4º Implementación de Virustotal en WAZUH	45
Paso 4.1 Configurar Victim.lab.	51
Paso 4.1.2 Prueba de implantación.	52
CONCLUSIÓN:	55

Punto 0. AVISO IMPORTANTE.

1. Declaración general

El presente proyecto realiza **únicamente una simulación controlada de un intento de ataque** con fines exclusivamente **académicos, didácticos y de práctica segura** en materia de respuesta a incidentes. En ningún caso se desarrollará, distribuirá, reproducirá ni utilizará software malicioso real, ni se realizarán acciones que puedan dañar sistemas ajenos, privacidad de terceros, infraestructuras públicas o privadas fuera del entorno de laboratorio definido.

2. Ámbito y límites

- **Ámbito:** Las pruebas y experimentos se llevarán a cabo únicamente en el entorno de laboratorio local provisionado en VMware, en **red Host-Only** (VMnet1 o equivalente), sin acceso a Internet ni a sistemas externos.
- **Máquinas involucradas:** solamente las VMs listadas en este proyecto (por ejemplo soc.lab, victim.lab, attacker.lab, cowrie.lab).
- **Límites:** no se realizará ningún escaneo, explotación ni propagación fuera del laboratorio. **No se intentará vulnerar sistemas de terceros ni realizar pruebas en redes públicas o de la institución sin autorización expresa y por escrito.**

3. Naturaleza de las acciones realizadas

- Todas las herramientas que se usarán no se desvela las acciones llevadas a cabo, para evitar problemas de ética.



Punto 1. Configuración del laboratorio.

A continuación se detallan los requisitos para llevar a cabo esta simulación controlada.

#	Hostname	Rol	SO sugerido	vCPU	RAM	Disco	NIC	IP estática
1	soc.lab	SOC:Wazuh	Ubuntu 24.04 LTS	2	8 GB	70 GB	<ul style="list-style-type: none"> • 1 Host-Only (red del laboratorio) • 1 NAT (solo durante instalación) 	192.168.75.20
2	victim.lab	Endpoint víctima	Windows 10 / Ubuntu	1	2 GB	20 GB	1 Host only 1 NAT en un paso intermedio de instalación	192.168.75.137 (DHCP)
3	attacker.lab	Máquina atacante: Kali / Ubuntu con herramientas	Kali Linux	1	2 GB	20 GB	1 Host only	192.168.75.130

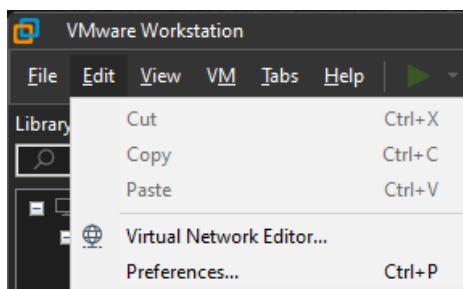
Punto 1º Configuración de las máquinas virtuales.

En este primer paso, vamos a montar las respectivas máquinas virtuales, para ello, no se detalla todo el proceso de creación de cada máquina, se da por hecho que quien lea esto, debe tener ciertos conocimientos sobre virtualización.

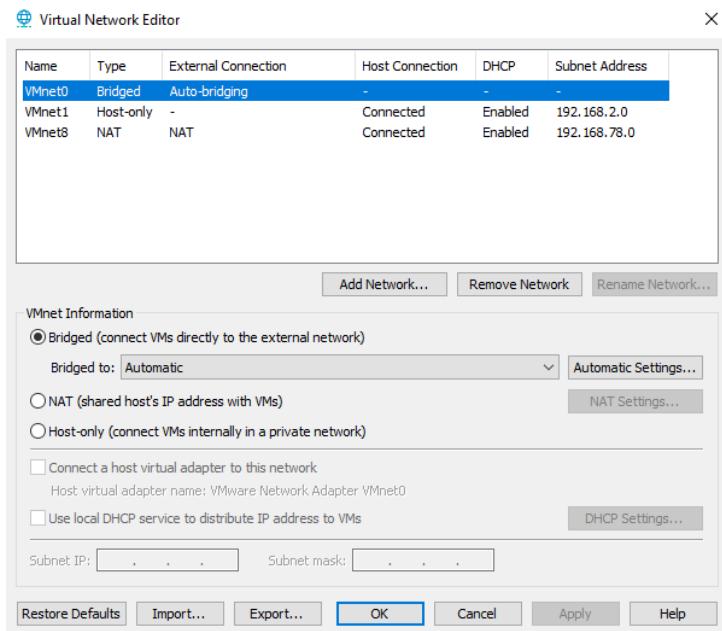
Punto 1.1 Configuración de la red del entorno.

En este breve paso, se detalla cómo va a ser la red a usar.

Yo usaré vmware pro, pero es aplicable, si usamos cualquier otro sistema de virtualización.



Vamos al editor de redes...



Damos en add network.

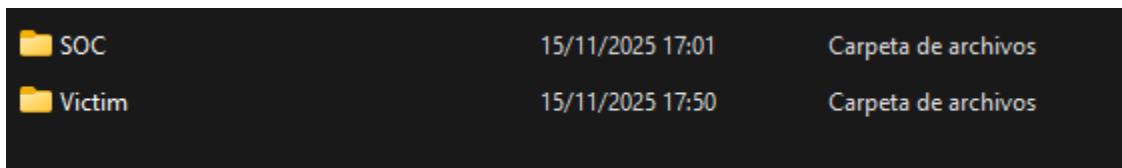
Y seleccionamos una vmnet....yo elegí la 4.

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet1	Host-only	-	Connected	Enabled	192.168.75.0

Será host only, aunque para ciertos momentos, como ubuntu server con el tema de paquetes y dependencias.....pues será necesario usar NAT, o que tenga doble interfaz, pero en cuánto esté todo listo...será un entorno aislado totalmente.

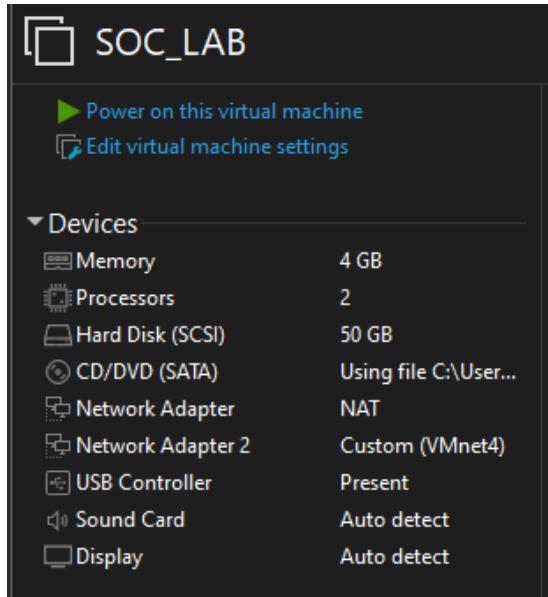
Punto 1.1.2. Configuración del directorio de vm.

En este breve paso, sólo destacar que es necesario tener bien definida dónde tendremos las máquinas virtuales.



Punto 1.2 Soc.Lab.

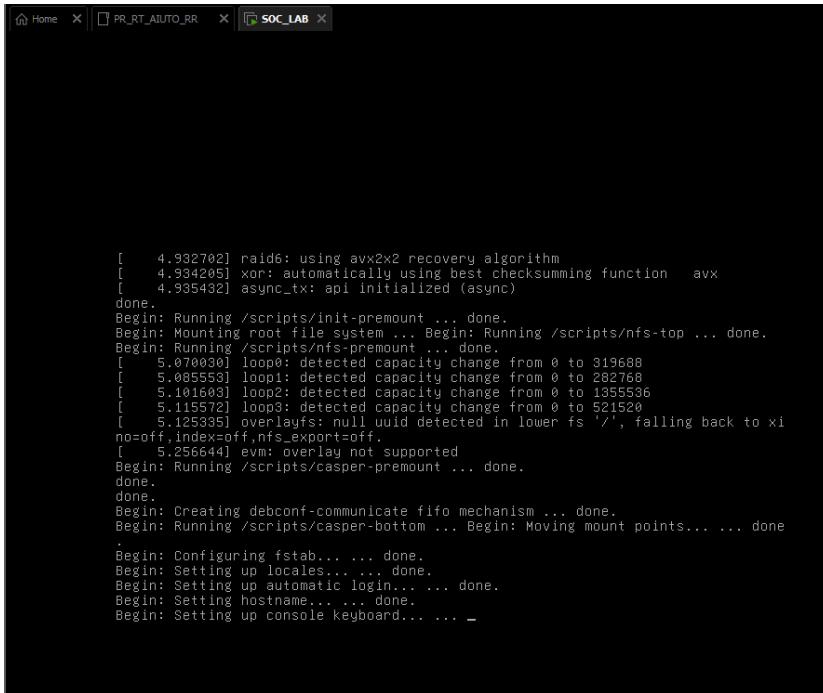
En este paso, vamos a configurar y crear la máquina virtual de Ubuntu Server.



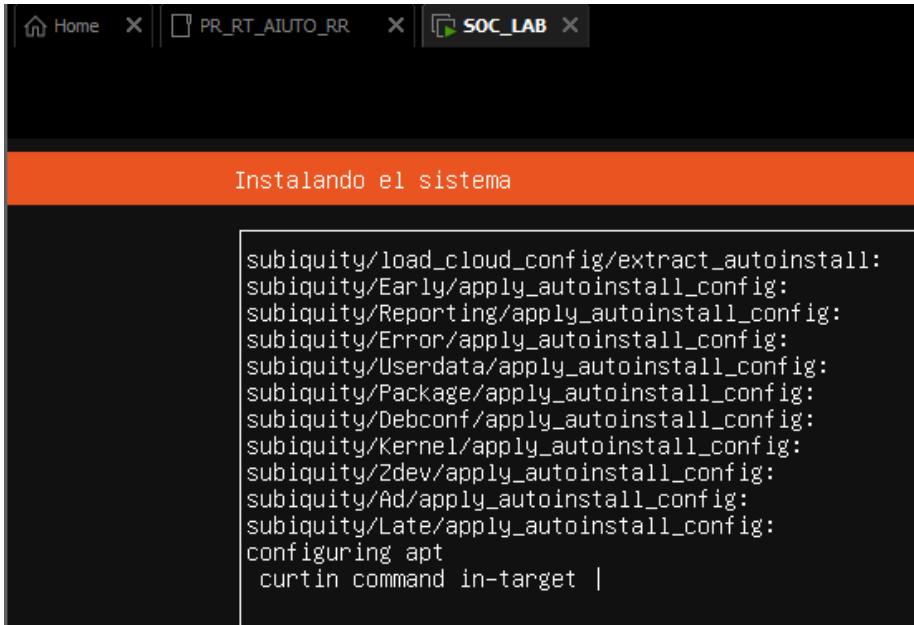
Aquí la tenemos creada, la máquina virtual dónde tendremos el “núcleo” de operación.

Iniciamos la máquina virtual....





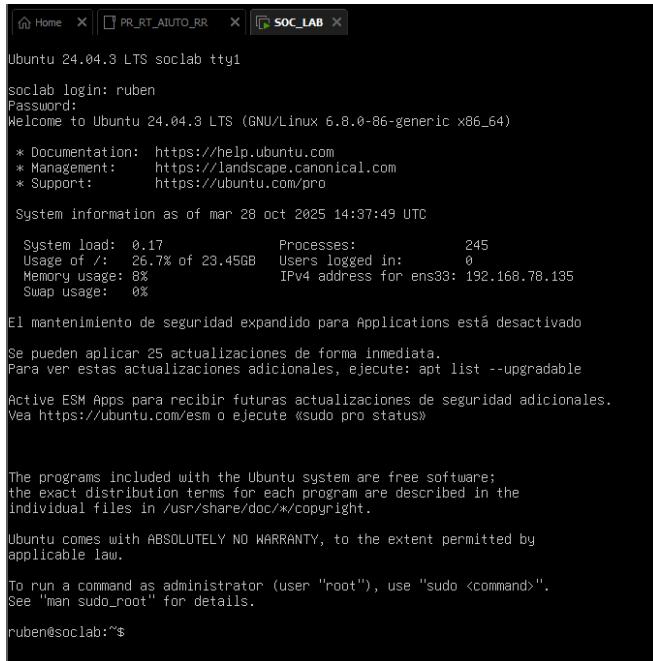
```
[    4.932702] raid6: using avx2x2 recovery algorithm
[    4.934205] xor: automatically using best checksumming function   avx
[    4.935432] async_tx: api initialized (async)
done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/nfs-top ... done.
Begin: Running /scripts/nfs-premount ... done.
[    5.070030] loop0: detected capacity change from 0 to 319688
[    5.085553] loop1: detected capacity change from 0 to 282768
[    5.101603] loop2: detected capacity change from 0 to 1355536
[    5.115572] loop3: detected capacity change from 0 to 521520
[    5.125335] overlayfs: null uid detected in lower fs '/', falling back to xi
no-off, index=off, nfs_export=off.
[    5.256644] evm: overlay not supported
Begin: Running /scripts/casper-premount ... done.
done.
done.
Begin: Creating debconf-communicate fifo mechanism ... done.
Begin: Running /scripts/casper-bottom ... Begin: Moving mount points... ... done
.
Begin: Configuring fstab... ... done.
Begin: Setting up locales... ... done.
Begin: Setting up automatic login... ... done.
Begin: Setting hostname... ... done.
Begin: Setting up console keyboard... ... _
```



```
subiquity/load_cloud_config/extract_autoinstall:
subiquity/Early/apply_autoinstall_config:
subiquity/Reporting/apply_autoinstall_config:
subiquity/Error/apply_autoinstall_config:
subiquity/Userdata/apply_autoinstall_config:
subiquity/Package/apply_autoinstall_config:
subiquity/Debconf/apply_autoinstall_config:
subiquity/Kernel/apply_autoinstall_config:
subiquity/2dev/apply_autoinstall_config:
subiquity/Ad/apply_autoinstall_config:
subiquity/Late/apply_autoinstall_config:
configuring apt
  curtin command in-target |
```

Instalandose.....

Una vez terminereiniciamos la vm....



The screenshot shows a terminal window titled "SOC LAB" with the following text:

```
Ubuntu 24.04.3 LTS soclab tty1
soclab login: ruben
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-86-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of mar 28 oct 2025 14:37:49 UTC

 System load: 0.17      Processes: 245
 Usage of /: 26.7% of 23.45GB  Users logged in: 0
 Memory usage: 8%
 Swap usage: 0%

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 25 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ruben@soclab:~$
```

Ya lo tenemos listo.

*Nota

Debemos de cambiar la zona horaria, ya que por defecto es UTF +0

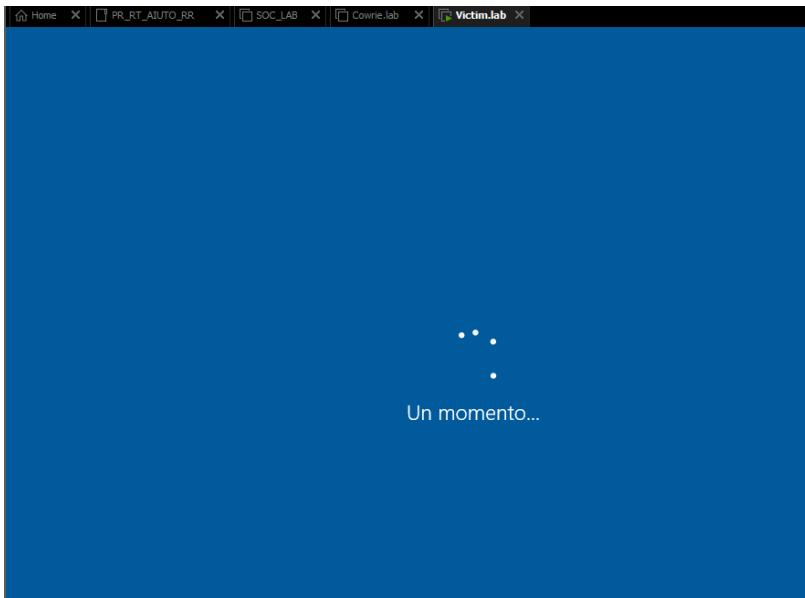
Tenemos que poner la zona de Madrid.

sudo timedatectl set-timezone Europe/Madrid

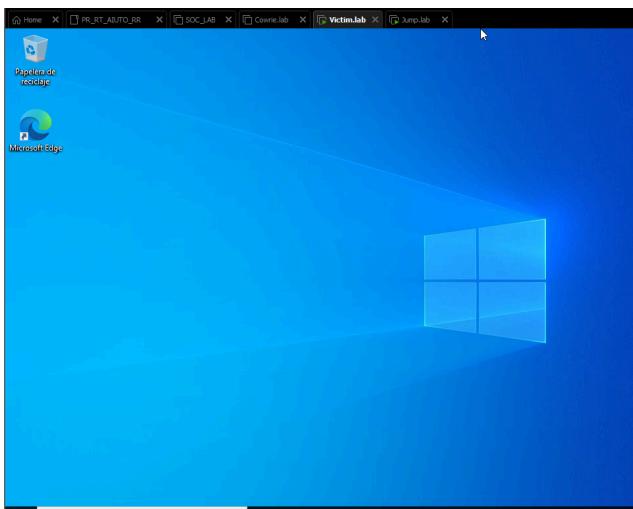
Punto 1.3 Máquina virtual victim.lab

Vamos con la máquina víctima, montaré un Windows 10.

Para ello, creamos la vm lo primero de todo....



En el proceso de configuración...



Ya tenemos la máquina virtual lista para cuándo la necesitemos.

Punto 2º Instalación y configuración del SOC (Wazuh)

El servidor SOC (soc.lab) actuará como centro de monitorización y detección de amenazas, utilizando Wazuh como SIEM, el cual incluye Elasticsearch y Kibana integrados en una única solución gestionada.

A continuación se presenta la explicación técnica y los pasos ejecutados para desplegar el **SOC** del laboratorio utilizando **Wazuh** (gestor/analizador de seguridad) con su **indexer** (Elasticsearch) y **dashboard** (Kibana) integrados. Este apartado está pensado para incluirse directamente en la memoria práctica: contiene objetivos, comandos principales, comprobaciones y notas de diseño/recursos.



4 Ejemplo práctico

- **Con ELK puro:**

- Recibes logs del sistema.
- Debes crear filtros y dashboards para verlos.
- Si quieres alertas, debes programar tus propias reglas.

- **Con Wazuh:**

- El agente del cliente envía los eventos automáticamente.
- Wazuh los analiza con su motor de correlación.
- Si detecta, por ejemplo, un intento de fuerza bruta, genera una alerta de seguridad y la almacena en Elasticsearch.

👉 En resumen:

🔒 **ELK** es una herramienta de análisis.

⚠️ **Wazuh** es una herramienta de **detección, análisis y respuesta** (usa ELK como base).

- Wazuh Indexer = Elasticsearch
- Filebeat = Logstash
- Wazuh Dashboard = Kibana



Punto 2.1 Wazuh

En este apartado se preparará el servidor **soc.lab** para la instalación de **Wazuh Manager**, que actuará como **núcleo del SOC**, recolectando, procesando y almacenando los logs y eventos de los agentes instalados en los endpoints.

Wazuh se integrará con **ElasticSearch y Kibana** para:

- **Almacenamiento:** todos los eventos y logs se indexan en ElasticSearch.
- **Visualización:** los dashboards y alertas se presentan en la interfaz web de Kibana/Wazuh Dashboard.



Primero, vamos a instalar unas dependencias.

```
usuario@soc:~$ sudo apt install vim curl apt-transport-https unzip wget libcap2-bin software-properties-common lsb-release gnupg2
[sudo] password for usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
vim ya está en su versión más reciente (2:9.1.0016-lubuntu7.9).
fijado vim como instalado manualmente.
curl ya está en su versión más reciente (8.5.0-2ubuntu10.6).
fijado curl como instalado manualmente.
wget ya está en su versión más reciente (1.21.4-lubuntu4.1).
fijado wget como instalado manualmente.
libcap2-bin ya está en su versión más reciente (1:2.66-5ubuntu2.2).
fijado libcap2-bin como instalado manualmente.
software-properties-common ya está en su versión más reciente (0.99.49.3).
fijado software-properties-common como instalado manualmente.
lsb-release ya está en su versión más reciente (12.0-2).
fijado lsb-release como instalado manualmente.
Paquetes sugeridos:
  zip
Se instalarán los siguientes paquetes NUEVOS:
  apt-transport-https gnupg2 unzip
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 183 kB de archivos.
Se utilizarán 454 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] ■
```

Cambiaremos el nombre del host a...

```
usuario@soc:~$ sudo hostnamectl set-hostname soc.lab
usuario@soc:~$ hostnamectl
  Static hostname: soc.lab
    Icon name: computer-vm
      Chassis: vm ■
    Machine ID: d2975d17e26241648d11035ae3e5c72e
      Boot ID: 7ca2a12762b04e43a93108e2ele9656b
  Virtualization: vmware
Operating System: Ubuntu 24.04.3 LTS
      Kernel: Linux 6.8.0-87-generic
  Architecture: x86-64
  Hardware Vendor: VMware, Inc.
  Hardware Model: VMware Virtual Platform
Firmware Version: 6.00
  Firmware Date: Thu 2020-11-12
  Firmware Age: 4y 11month 3w 5d
usuario@soc:~$ ■
```

Tras hacer el cambio, vamos a instalar wazuh.

Para ello, debemos de instalarlo con un script automatico de la propia Wazuh.

The screenshot shows the Wazuh website homepage. At the top, there is a navigation bar with links for Blog, Community, Contact us, and various social media icons. Below the navigation bar, the Wazuh logo is displayed, followed by a dropdown menu for Platform, Cloud, CTI, Documentation, Services, Partners, Company, and two buttons: "Install Wazuh" and "Login". The main content area features a large heading "The Open Source Security Platform" and a subtext "Unified XDR and SIEM protection for endpoints and cloud workloads." Below this are two buttons: "Install Wazuh" and "Free Cloud Trial". To the right of the text, there is a large screenshot of the Wazuh Threat Hunting dashboard, which includes a chart of alerts level evolution, a pie chart of top 5 agents, and a bar chart of alerts evolution for top 5 agents. The dashboard also displays statistics for total alerts (176957), Level 12 or above alerts (9), Authentication failure (33882), and Authentication success (45). A legend for MITRE ATT&CK techniques is provided.



Telefónica

Globant



Rappi

Mondelēz
International

Install wazuh.

Damos en Quickstart.

The screenshot shows the Wazuh Quickstart page. It features three main sections: "W.indexer", "W.server", and "W.dashboard".
W.indexer: Described as a highly scalable, full-text search and analytics engine that indexes and stores alerts generated by the Wazuh server.
W.server: Described as a component that analyzes data received from agents and processes it using threat intelligence, capable of managing thousands of agents and scaling as a cluster.
W.dashboard: Described as the web user interface for data visualization, analysis, and management, including dashboards for compliance, vulnerabilities, file integrity, configuration assessment, and cloud infrastructure events.
At the bottom right of the page are two buttons: "Quickstart" and "Installation guide".

Tras ello, nos llevará a la siguiente página.



SOCwAttack Lab © 2025 by R. Rubén is licensed under CC BY-NC-ND 4.0

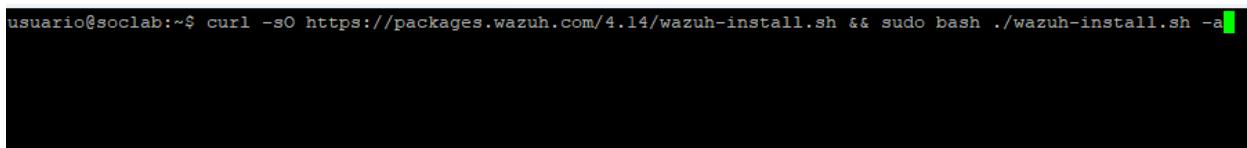
The screenshot shows the Wazuh Quickstart page. On the left, there's a sidebar with a search bar and a navigation menu including 'Getting started', 'Quickstart' (which is highlighted), 'Installation guide', 'Installation alternatives', 'User manual', 'Cloud security', 'Regulatory compliance', 'Proof of Concept guide', 'Upgrade guide', 'Integrations guide', 'Backup guide', 'Wazuh Cloud service', 'Development', and 'Release notes'. The main content area has a breadcrumb trail 'Home / Quickstart'. It lists supported operating systems: 'CentOS Stream 10', 'Red Hat Enterprise Linux 7, 8, 9, 10', and 'Ubuntu 16.04, 18.04, 20.04, 22.04, 24.04'. Below this, a section titled 'Installing Wazuh' contains instructions: '1. Download and run the Wazuh installation assistant.' followed by a command line example: '\$ curl -sO https://packages.wazuh.com/4.14/wazuh-install.sh && sudo bash ./wazuh-install.sh -a'. A note says 'Once the assistant finishes the installation, the output shows the access credentials and a message that confirms that the installation was successful.' A code block shows the terminal output of the installation process, including 'INFO: --- Summary ---', 'INFO: You can access the web interface https://<WAZUH_DASHBOARD_IP_ADDRESS>', 'User: admin', 'Password: <ADMIN_PASSWORD>', and 'INFO: Installation finished.'. At the bottom, a message says 'You now have installed and configured Wazuh.'

Cogeremos el comando:

```
curl -sO https://packages.wazuh.com/4.14/wazuh-install.sh && sudo bash  
./wazuh-install.sh -a
```

*Importante: hay que actualizar el sistema y repositorios antes de ejecutar el comando anterior.

Y lo ejecutamos en nuestro servidor.



```
usuario@soclab:~$ sudo apt update -y && sudo apt upgrade -y  
Obj:1 http://es.archive.ubuntu.com/ubuntu noble InRelease  
Obj:2 http://security.ubuntu.com/ubuntu noble-security InRelease  
Obj:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease  
Obj:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease  
Des:5 http://es.archive.ubuntu.com/ubuntu noble/main Translation-es [325 kB]  
Des:6 http://es.archive.ubuntu.com/ubuntu noble/restricted Translation-es [816 B]  
Des:7 http://es.archive.ubuntu.com/ubuntu noble/universe Translation-es [1.371 kB]  
Des:8 http://es.archive.ubuntu.com/ubuntu noble/multiverse Translation-es [63,1 kB]  
100% [7 Translation-es store 0 B]
```

Ejecutamos el comando.

```
usuario@soclab:~$ curl -s0 https://packages.wazuh.com/4.14/wazuh-install.sh && sudo bash ./wazuh-install.sh -a
09/11/2025 16:16:59 INFO: Starting Wazuh installation assistant. Wazuh version: 4.14.0
09/11/2025 16:16:59 INFO: Verbose logging redirected to /var/log/wazuh-install.log
```

Esperamos....

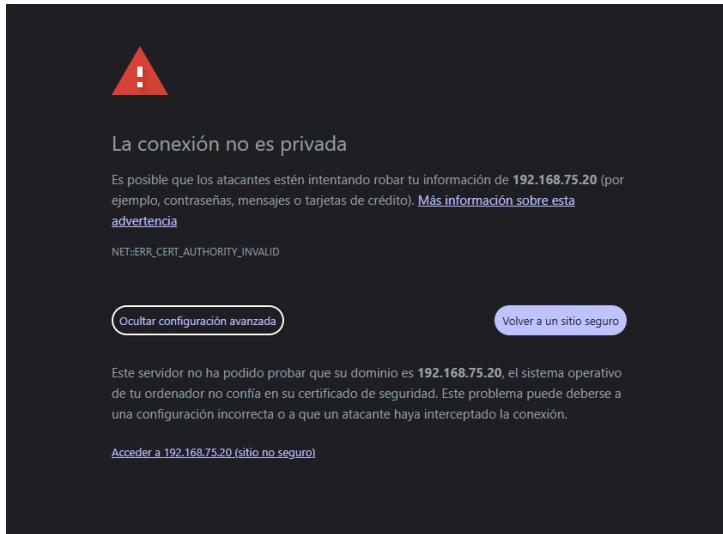
```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
usuario@soclab:~$ curl -s0 https://packages.wazuh.com/4.14/wazuh-install.sh && sudo bash ./wazuh-install.sh -a
09/11/2025 16:16:59 INFO: Starting Wazuh installation assistant. Wazuh version: 4.14.0
09/11/2025 16:16:59 INFO: Verbose logging redirected to /var/log/wazuh-install.log
09/11/2025 16:17:04 INFO: Verifying that your system meets the recommended minimum hardware requirements.
09/11/2025 16:17:04 INFO: Wazuh web interface port will be 443.
09/11/2025 16:17:09 INFO: --- Dependencies ---
09/11/2025 16:17:09 INFO: Installing apt-transport-https.
09/11/2025 16:17:12 INFO: Installing debhelper.
```

Y ya estaría terminado.

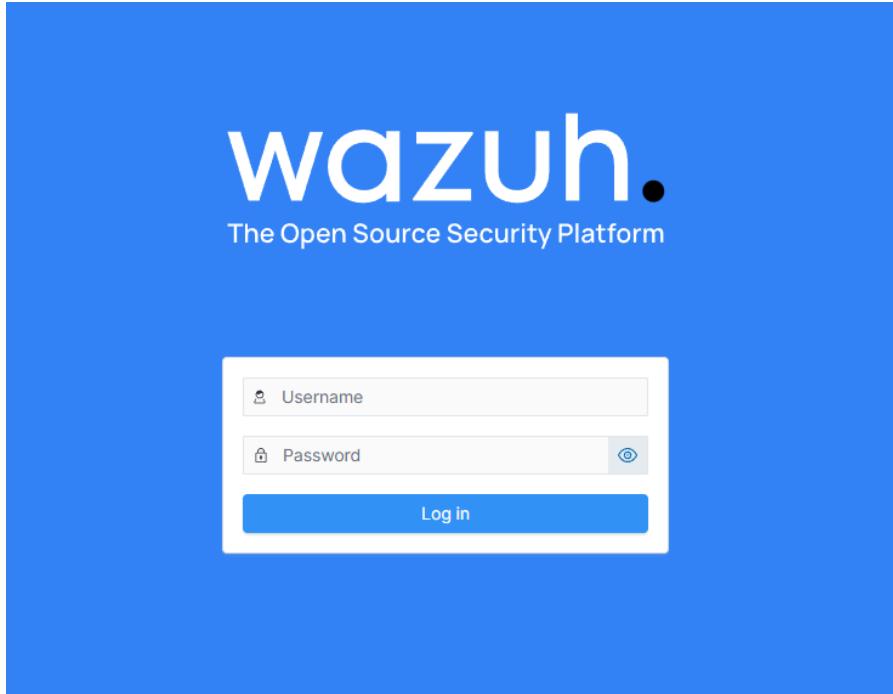
```
Filebeat
09/11/2025 16:21:05 INFO: The filebeat.yml file has been updated to use the Filebeat Keystore username and password.
09/11/2025 16:21:36 INFO: Initializing Wazuh dashboard web application.
09/11/2025 16:21:36 INFO: Wazuh dashboard web application initialized.
09/11/2025 16:21:36 INFO: --- Summary ---
09/11/2025 16:21:36 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
    User: admin
    Password: [REDACTED]
09/11/2025 16:21:36 INFO: Installation finished.
usuario@soclab:~$
```

Si accedemos por el navegador por ejemplo de nuestro host....

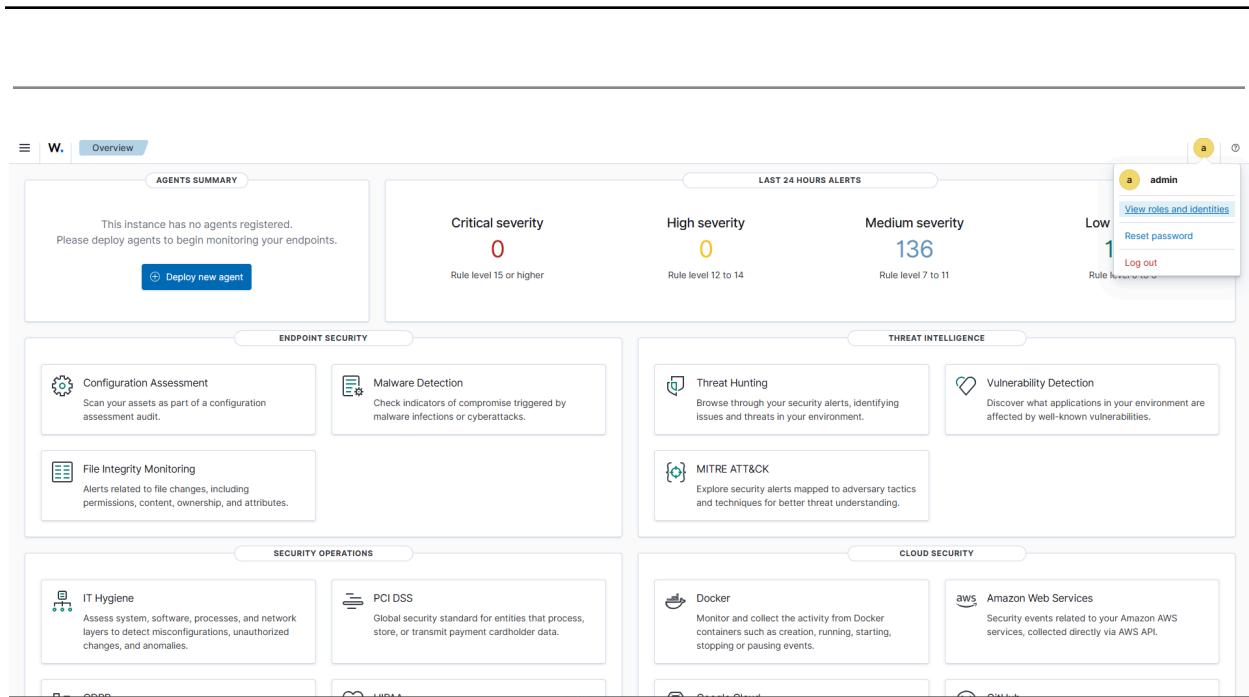
 No es seguro <https://192.168.75.20>



Continuamosdamos en acceder, esto es por el certificado autofirmado que tiene wazuh.



Ponemos las credenciales que se nos arrojó en la terminal.



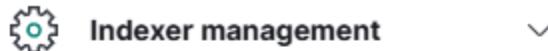
Y aquí lo tendríamos.

Paso 2.1.2 Creación de usuario.

Cómo buenos aspirantes a ser buenos en el control de seguridad en una aplicación, vamos a crearnos un usuario nuestro.

Para ello, hay que seguir unos pasos.

Nos vamos a Wazuh.



Index Management

Snapshot Management

Security

Sample Data

Dev Tools

Nos vamos a Security.

The OpenSearch security plugin lets you define the API calls that users can make and the data they can access. The most basic configuration consists of these steps.

1 Add backends
Add authentication (authc) and authorization (authz) information to `config/opensearch-security/config.yml`. The authc section contains the backends to check user credentials against. The authz section contains any backends to fetch backend roles from. The most common example of a backend role is an LDAP group. [Learn more](#)

[Config.yml documentation](#) [Review authentication and authorization](#)

2 Create roles
Roles are reusable collections of permissions. The default roles are a great starting point, but you might need to create custom roles that meet your exact needs. [Learn more](#)

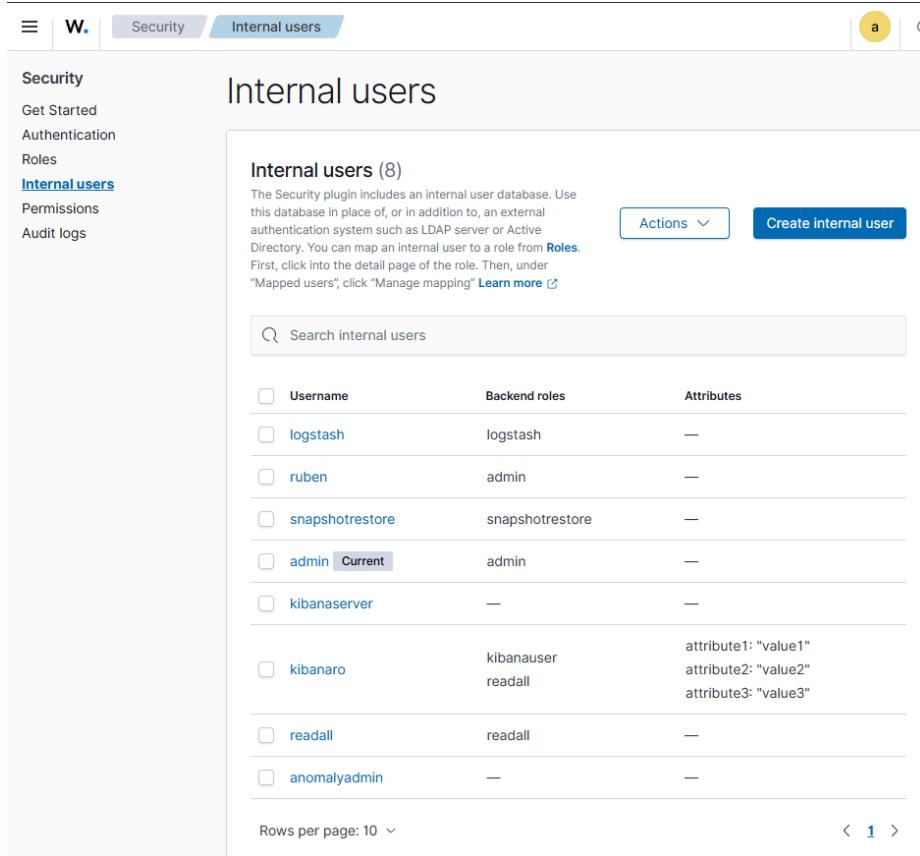
[Explore existing roles](#) [Create new role](#)

3 Map users
Map internal users and map backend roles

Damos en internal users.

SOCwAttack Lab © 2025 by R. Rubén is licensed under CC BY-NC-ND 4.0





The screenshot shows the Kibana Security plugin interface under the 'Internal users' tab. On the left, there's a sidebar with links like 'Security', 'Get Started', 'Authentication', 'Roles', 'Internal users' (which is highlighted in blue), 'Permissions', and 'Audit logs'. The main area is titled 'Internal users' and shows a table of users with 8 entries. The columns are 'Username', 'Backend roles', and 'Attributes'. The users listed are: logstash (backend role logstash), ruben (backend role admin), snapshotrestore (backend role snapshotrestore), admin (backend role admin, current user), kibanaserver (backend role none), kibanauser (backend role readall, attributes attribute1: "value1", attribute2: "value2", attribute3: "value3"), readall (backend role readall), and anomalyadmin (backend role none). There are also 'Actions' and 'Create internal user' buttons at the top right of the table area.

Yo me creé un nuevo usuario, y con ello, vamos a tomar acción para hacer otro usuario.

Damos en Create internal user.

The screenshot shows a 'Create internal user' form. At the top, there are tabs for 'Security', 'Internal users', and 'Create internal user'. A user icon with the letter 'a' is visible. The main area has three sections:

- Credentials**: Fields for 'Username' (with placeholder 'Specify a descriptive and unique user name. You cannot edit the name once the user is created.'), 'Password' (with placeholder 'The user name must contain from 2 to 50 characters. Valid characters are A-Z, a-z, 0-9, underscore, hyphen and unicode characters.'), and 'Re-enter password'.
- Backend roles - optional**: A section for mapping users from external authentication systems. It includes a 'Backend role' input field ('Type in backend role') and a 'Remove' button. Below it is a 'Add another backend role' button.
- Attributes - optional**: A section for further describing the user. It includes a note: 'Attributes can be used to further describe the user, and more importantly they can be used as variables in the Document Level Security query in the index permission of a role. This makes it possible to write dynamic DLS queries.'

Rellenamos.

Create internal user

The security plugin includes an internal user database. Use this database in place of, or in addition to, an external authentication system such as LDAP or Active Directory. [Learn more](#)

Credentials

Username
Specify a descriptive and unique user name. You cannot edit the name once the user is created.

The user name must contain from 2 to 50 characters. Valid characters are A-Z, a-z, 0-9, (_)underscore, (-) hyphen and unicode characters.

Password
 •••••••• eye icon
Password should be at least 8 characters long and contain at least one uppercase letter, one lowercase letter, one digit, and one special character.

>Password strength Very strong

Re-enter password
 •••••••• eye icon
The password must be identical to what you entered above.

Backend roles - optional

Backend roles are used to map users from external authentication systems, such as LDAP or SAML to OpenSearch security roles. [Learn more](#)

Backend role
 Remove

[Add another backend role](#)

Attributes - optional

Tenemos que poner en backend roles, admin, aunque hay miles, ya es granular el grado de permiso al usuario.

Ya lo tendríamos.

The screenshot shows the "Internal users" section of the Grafana Security plugin. On the left, there's a sidebar with links: Security, Get Started, Authentication, Roles, Internal users (which is underlined in blue), Permissions, and Audit logs. The main area has a title "Internal users" and a sub-section "Internal users (9)". It includes a brief description about the internal user database and how it can be used in addition to external authentication systems like LDAP or Active Directory. There are "Actions" and "Create internal user" buttons. Below this is a search bar labeled "Search internal users". A table lists two users: "logstash" with "Backend roles" set to "logstash" and "Attributes" set to "-".

Con esto, ya estará en el indexer (El acceso al backend).

Para poder acceder al dashboard, tendremos que irnos al Server management.

The screenshot shows the "Server management" section of Grafana. At the top, there's a "Logs" tab which is highlighted with a grey background. Other tabs include Rules, Decoders, CDB Lists, Status, Cluster, Statistics, Settings, Dev Tools, Ruleset Test, and Security. To the right of the tabs, there's a dropdown menu icon.

A security nuevamente.

The screenshot shows the Wazuh Security interface with the 'Roles mapping' tab selected. A yellow banner at the top right provides instructions: '⚠️ For the role mapping to take effect, enable run_as in the API host configuration, restart the dashboard service and clear your browser cache and cookies.' Below this, the 'Roles mapping' table lists three entries:

ID ↑	Name	Roles	Status	Ac...
1	wui_elastic_admin	administrator	Reserved	wazuh-wui
2	wui_opensearch_admin	administrator	Reserved	wazuh-wui
100	ruadmin	administrator		

At the bottom, there is a search bar, a 'Create Role mapping' button, and pagination controls.

Creamos un nuevo role mapping.

The screenshot shows the 'Security' section of a web application. On the left, there's a sidebar with 'Users', 'Roles', 'Policies', and 'Roles mapping'. The 'Roles mapping' tab is selected. A message at the top says: '⚠️ For the role mapping to take effect, enable run_as in the browser cache and cookies.' Below this is a table titled 'Roles mapping' with columns 'ID', 'Name', and 'Roles'. It contains three rows: '1 wui_elastic_admin administrator', '2 wui_opensearch_admin administrator', and '100 ruadmin administrator'. A search bar is also present. On the right, a modal window titled 'Create new role mapping' is open. It has a 'Role mapping name' field containing 'prueba_mapping', a 'Roles' dropdown menu listing various roles like 'administrator', 'agents_admin', etc., with 'administrator' selected, and sections for 'Map internal users' and 'Custom rules'.

Podemos elegir, según la importancia que queremos dar a dicho usuario.

Para hacerlo administrador, elegimos administrador.

Create new role mapping

Role mapping name
prueba_mapping

Roles
administrator

Mapping rules

Map internal users
Internal users
prueba

Custom rules
Any are true

Save role mapping

ID	Name	Roles
1	wui_elastic_admin	administrator
2	wui_opensearch_admin	administrator
100	ruadmin	administrator

Para finalizar, asignamos el mapeo al usuario creado anteriormente.

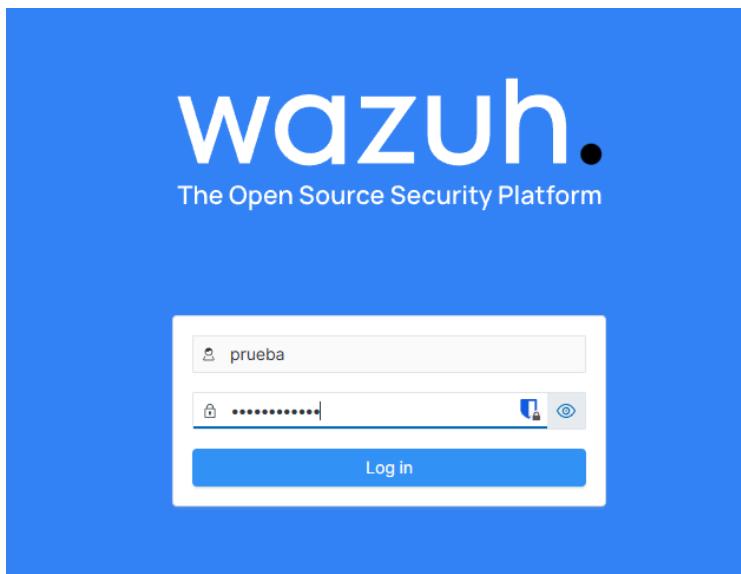
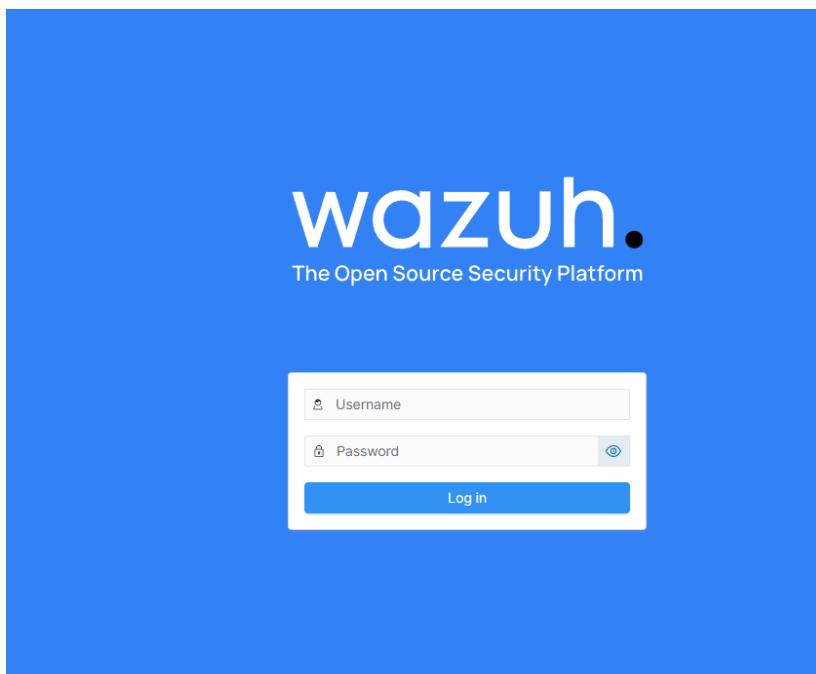
Y save.

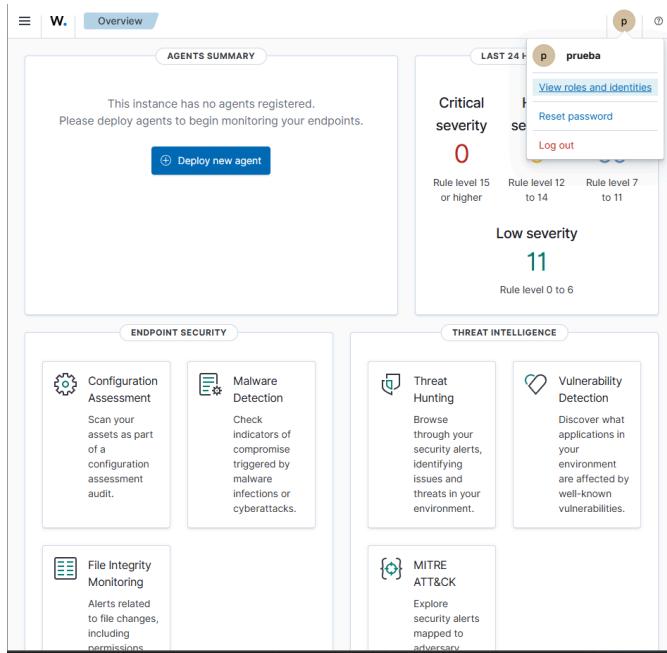
Roles mapping

Create Role mapping

ID	Name	Roles	Status	Ac...
1	wui_elastic_admin	administrator	Reserved	wazuh-wui
2	wui_opensearch_admin	administrator	Reserved	wazuh-wui
100	ruadmin	administrator		
101	prueba_mapping	administrator		

Vamos a probar a acceder.





Aquí lo tenemos.

Para eliminar, haríamos los mismos pasos, pero eliminando el rol y el usuario.

Para cambiar la contraseña de admin, hay que un script de wazuh oficial:

[GUIA_Cambio contraseña](#)

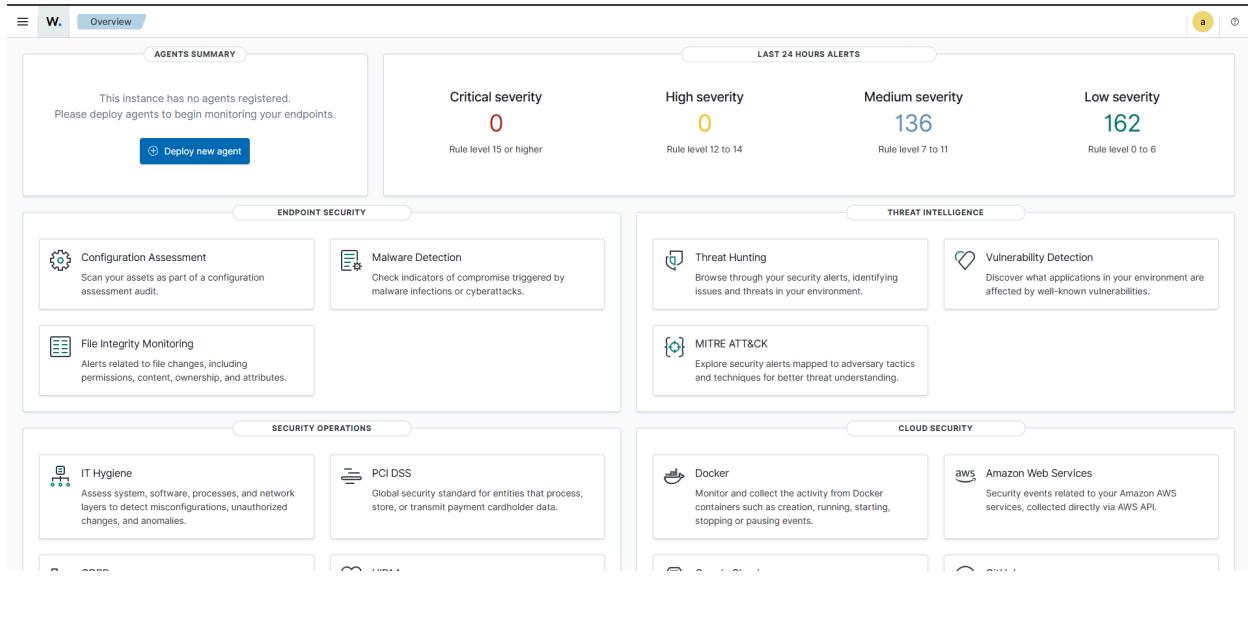
Punto 2.2 Configuración de los agentes.

En este punto vamos a ver cómo podemos meter la máquina virtual víctima, de Windows 10, en nuestro SIEM de Wazuh para luego ver las alertas que haremos con el Kali Linux.

Para ello, nos vamos a las tres barras.

Incidentes de ciberseguridad

Rubén



Agents management

Summary

Groups

Summary.

This section allows users to manage groups of agents. It displays a list of existing groups, including their names, agent counts, configuration checksums, and actions (Edit, Delete).

Name ↑	Agents	Configuration checksum	Actions
default	0	ab73af41699f13fdd81903b5f2 3d8d00	

Lo recomendable, es ser expertos, y hacer primero de todo un grupo.

Damos en add new group.

SOCwAttack Lab © 2025 by R. Rubén is licensed under CC BY-NC-ND 4.0



Name ↑	Agents	Configuration checksum	Actions
default	0	ab73af41699f13fdd81903b5f2 3d8d00	
IT	0	ab73af41699f13fdd81903b5f2 3d8d00	

Por poner un ejemplo. Creé un grupo llamado IT.

Luego, ya si nos vamos a Summary.

(o)

No agents were added to the manager

Add agents to fleet to start monitoring

+ Deploy new agent

Display new agent.

The screenshot shows a user interface for deploying a new agent. Step 1: Select the package to download and install on your system. It offers three main categories: LINUX, WINDOWS, and macOS. Under LINUX, there are two options: RPM amd64 and DEB amd64. Under WINDOWS, there is one option: MSI 32/64 bits. Under macOS, there are two options: Intel and Apple silicon. Step 2: Server address. This is the address the agent uses to communicate with the server. Enter an IP address or a fully qualified domain name (FQDN). Step 3: 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. A note states: "The agent name must be unique. It can't be changed once the agent has been enrolled." A yellow bar at the bottom indicates that the agent name must be unique.

Aquí tenemos un configurador para el agente, dependiendo de la situación deberemos de crear uno u otro perfil.

Vamos a crear uno de Windows.

The screenshot shows the 'Deploy new agent' interface for Windows. Step 1: Select the package to download and install on your system. The Windows section is selected, showing the MSI 32/64 bits option. Step 2: Server address. The field contains the IP address 192.168.75.20. Step 3: Optional settings. The agent name is set to 'Victim.lab'. A yellow bar at the bottom indicates that the agent name must be unique.



Lo podemos meter en el grupo anteriormente creado.

Select one or more existing groups: [?](#)



Se nos creará un comando para PS.

Run the following commands to download and install the agent:

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent...
```

Nos vamos a la vm de Victim.lab (Recordar, que estén en red, y que tenga otra interfaz con NAT, para descargar paquetes, y que no haya ningún problema.)

Abrimos una terminal en modo administrador de PS.

Y esperamos a que se termine de ejecutar.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Escribiendo solicitud web
Escribiendo secuencia de solicitud... (Número de bytes escritos: 2407884)

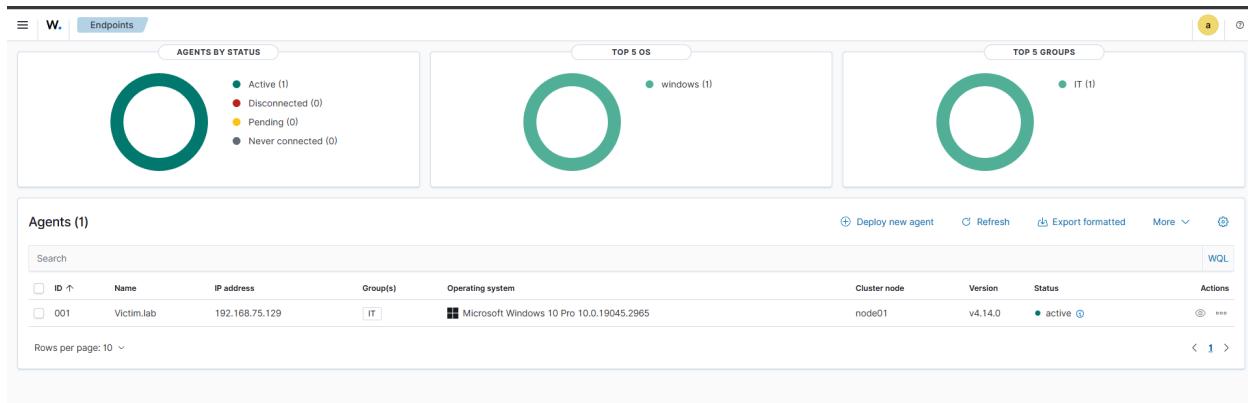
$env:tmp\wazuh-agent; msieexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='192.168.75.20' WAZUH_AGENT_NAME='Victim.lab'
```

E iniciamos el servicio de WAZUH

```
PS C:\Windows\system32> net start wazuh
El servicio de Wazuh está iniciándose.
El servicio de Wazuh se ha iniciado correctamente.

PS C:\Windows\system32> ■
```

En principio ha ido bien todo, vamos a nuestro server, y vemos si es así...



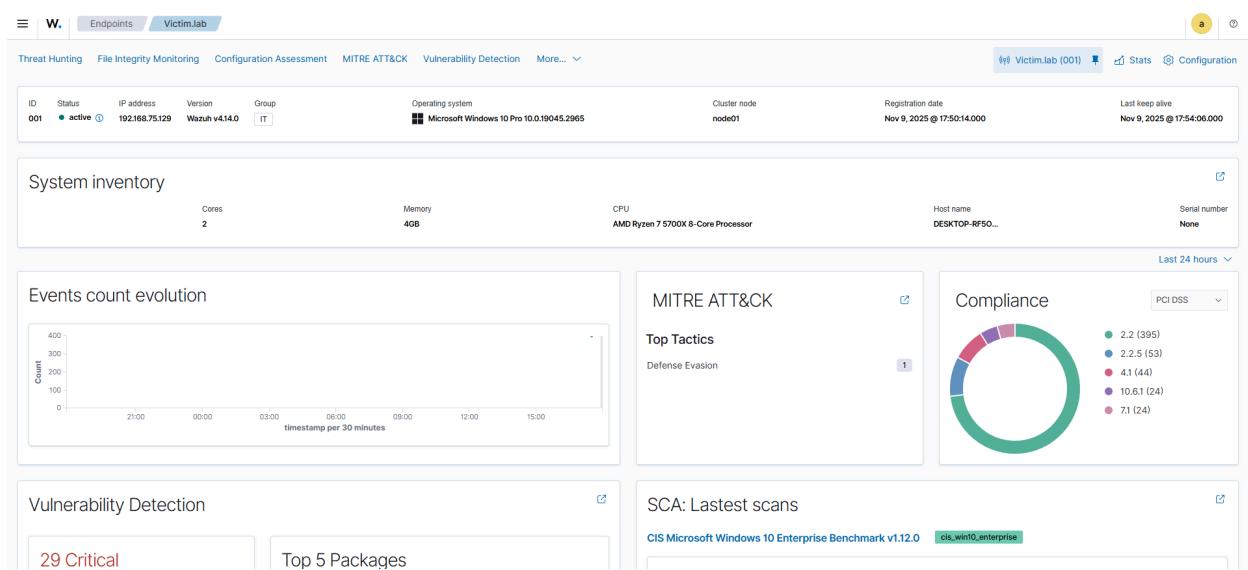
Dicho y hecho, ya tenemos aquí el dispositivo.

Punto 2.3 Información de los agentes.

Cómo bien podemos saber, estamos ante una herramienta muy buena, y tal es así que podemos saber de todo sobre los equipos enrollados en nuestro sistema SIEM.

Incidentes de ciberseguridad

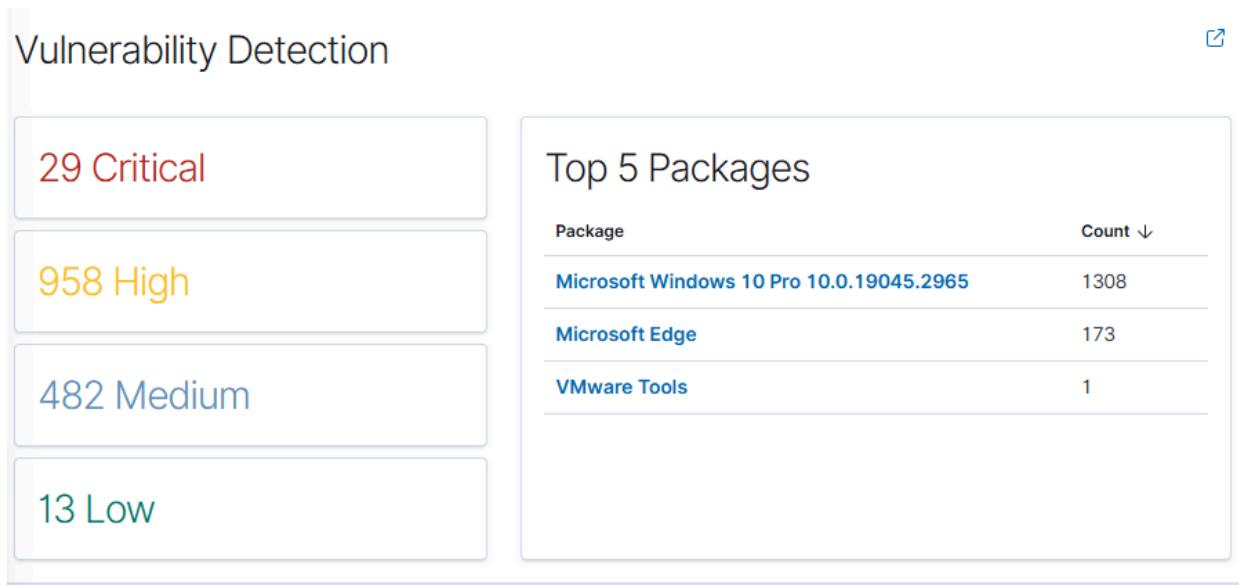
Rubén



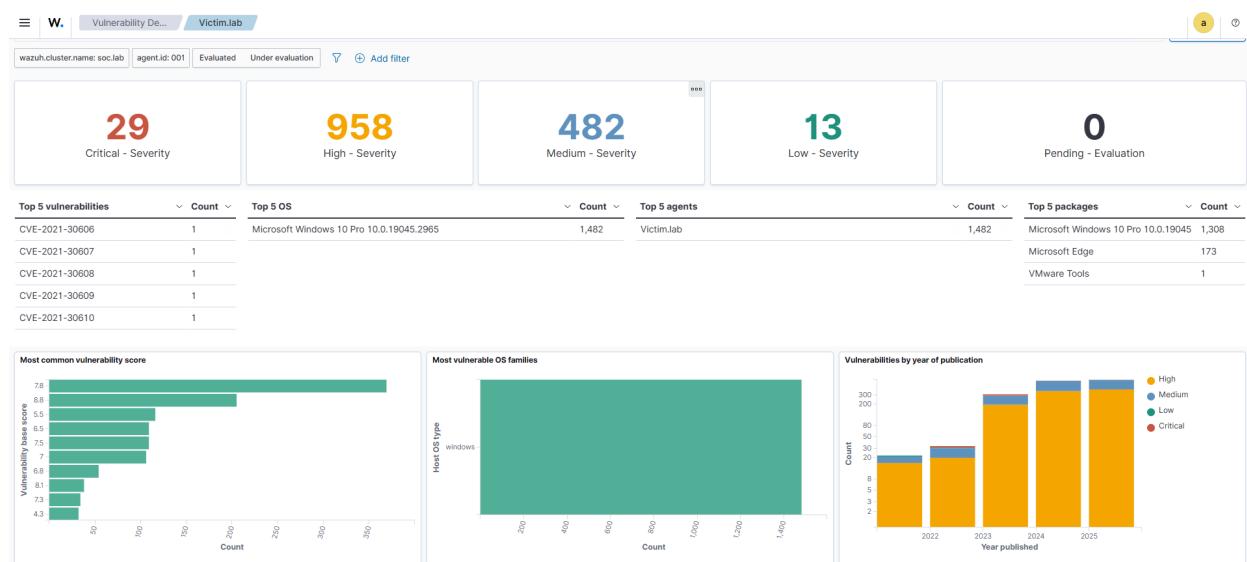
Esto por ejemplo es el dashboard de victim.lab.

Vemos mucha información, cómo la información del sistema, eventos...métricas.

Hast tal punto....



Qué hasta las vulnerabilidades de la máquina.



Más información detallada.

Podemos ver los CVE identificativos de las vulnerabilidades, para poder solventar y así no tener amenazas por ejemplo.

Punto 3º Pruebas de monitorización y detección de incidentes.

En este apartado se evaluará cómo responde el SOC ante un evento peligroso simulado. Para ello se realizarán pruebas controladas, como intentos de intrusión desde la máquina **Attacker.lab** hacia la máquina **Victim.lab**, ambas alojadas en un entorno virtual aislado (VM Host-Only). El objetivo es verificar que el sistema es capaz de generar, enviar y procesar eventos de seguridad ante comportamientos anómalos o maliciosos.

Declaración de responsabilidad, alcance y limitaciones

Todas las pruebas descritas en este trabajo se han realizado **únicamente con fines formativos**, dentro de un **entorno virtualizado, cerrado y completamente aislado de Internet**. Se certifica que:

1. Autorización:

El autor dispone de autorización para ejecutar las pruebas exclusivamente en las máquinas del laboratorio descritas.

2. Ámbito controlado:

Ninguna acción se ha realizado fuera del entorno virtual **ni ha afectado** a sistemas de terceros.

3. Uso ético:

No se ha distribuido, facilitado ni documentado código ni comandos que puedan emplearse con fines maliciosos.

Objetivo de este apartado

Con esta sección se busca demostrar que el SOC es capaz de:

- Detectar actividades sospechosas (por ejemplo, fuerza bruta).
- Registrar incidentes en tiempo real.

- Enviar alertas al panel de monitorización.

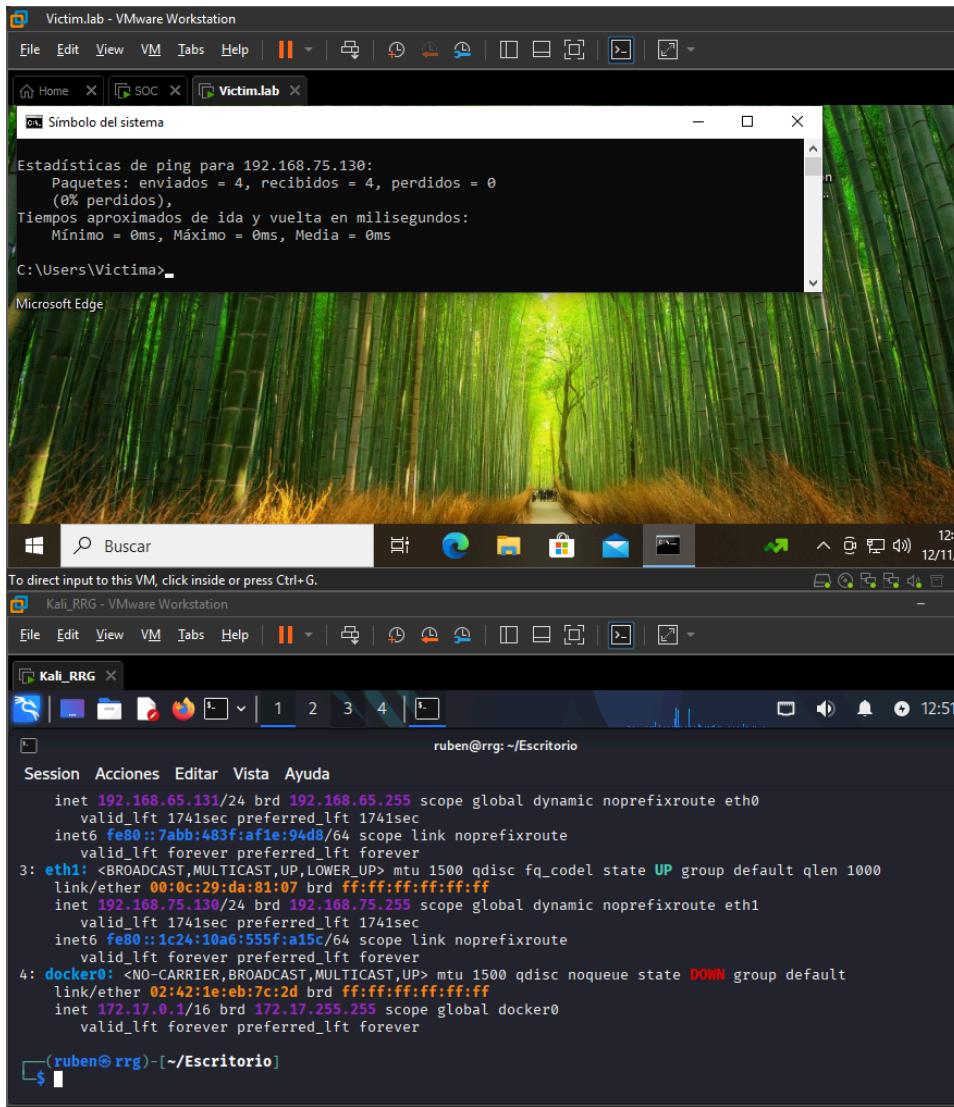
Tras este nuevo aviso, comenzamos.

En este apartado, vamos a usar Kali linux, como vm de nuestro laboratorio, dentro de la red Host-Only de vmware, para simular y poner a prueba el SOC de Wazuh, y ver claramente que el cliente manda los logs al soc, y este responde con alertas.

Para ello usaremos Hydra en la vm de Kali Linux, intentaremos que el SOC detecte esta actividad sospechosa.

Primero haremos si hay conexión entre víctima y atacante.





Luego, tras ver qué sivamos a hacer la simulación.

Tengamos en cuenta, que la vm de Kali linux, es la 192.168.75.135.

Porque luego nos hará falta para ver que realmente está funcionando.

Lo siguiente es verificar los eventos de seguridad de nuestro windows víctima.

Ejecutamos una terminal PS en modo administrador.

```
PS C:\Windows\system32> Get-WinEvent -LogName Security -MaxEvents 10 | Format-List -Property TimeCreated,Id,Message

TimeCreated : 14/11/2025 19:29:30
Id          : 5379
Message     : Las credenciales del Administrador de credenciales se leyeron.

Asunto:
  Id. de seguridad:      S-1-5-18
  Nombre de la cuenta:   DESKTOP-IB84R64$ 
  Dominio de la cuenta:  WORKGROUP
  Id. de inicio de sesión: 0x3E7
  Operación de lectura:  Enumerar las credenciales

Este evento se produce cuando un usuario realiza una operación de lectura en las credenciales almacenadas en el Administrador de credenciales.

TimeCreated : 14/11/2025 19:29:30
Id          : 5379
Message     : Las credenciales del Administrador de credenciales se leyeron.

Asunto:
  Id. de seguridad:      S-1-5-18
  Nombre de la cuenta:   DESKTOP-IB84R64$ 
  Dominio de la cuenta:  WORKGROUP
  Id. de inicio de sesión: 0x3E7
  Operación de lectura:  Enumerar las credenciales

Este evento se produce cuando un usuario realiza una operación de lectura en las credenciales almacenadas en el Administrador de credenciales.

TimeCreated : 14/11/2025 19:29:30
Id          : 5379
Message     : Las credenciales del Administrador de credenciales se leyeron.

Asunto:
  Id. de seguridad:      S-1-5-18
  Nombre de la cuenta:   DESKTOP-IB84R64$ 
  Dominio de la cuenta:  WORKGROUP
  Id. de inicio de sesión: 0x3E7
  Operación de lectura:  Enumerar las credenciales

Este evento se produce cuando un usuario realiza una operación de lectura en las credenciales almacenadas en el Administrador de credenciales.

TimeCreated : 14/11/2025 19:29:30
Id          : 5379
Message     : Las credenciales del Administrador de credenciales se leyeron.

Asunto:
  Id. de seguridad:      S-1-5-18
  Nombre de la cuenta:   DESKTOP-IB84R64$ 
  Dominio de la cuenta:  WORKGROUP
  Id. de inicio de sesión: 0x3E7
  Operación de lectura:  Enumerar las credenciales
```

Básicamente son los Logs de seguridad que hemos generado.

Paso 3.1 Asegurar que Victim.lab genera logs de error.

En este paso vamos a ver si nuestro Windows víctima genera logs de error de forma correcta.

Para ello, vamos a ejecutar un nuevo comando.

```
auditpol /set /subcategory:"Inicio de sesión" /success:enable /failure:enable
```

```
PS C:\Windows\system32> auditpol /set /subcategory:"Inicio de sesión" /success:enable /failure:enable
El comando se ejecutó correctamente.
```

Y luego verificamos...

```
PS C:\Windows\system32> auditpol /get /subcategory:"Inicio de sesión"
Directiva de auditoría del sistema
Categoría o subcategoría          Configuración
Inicio/cierre de sesión           Aciertos y errores
  Inicio de sesión
PS C:\Windows\system32>
```

Tras ver que está configurado, vamos a configurar la vm para que sea algo vulnerable.

Paso 3.1.1 RDP.

En este paso vamos a tocar la configuración como anteriormente dije para hacer que la máquina sea vulnerable.

Ejecutamos el comando...

```
PS C:\Windows\system32> Set-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name "fDenyTSConnections" -Value 0
PS C:\Windows\system32>
```

```
Set-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Terminal Server"
-Name "fDenyTSConnections" -Value 0
```

Luego de esto, que añade básicamente un ítem de propiedad al registro, vamos a habilitar una regla en el firewall de windows.



```
PS C:\Windows\system32> Enable-NetFirewallRule -DisplayGroup "Escritorio remoto"
PS C:\Windows\system32>
```

Cuando se escanee con alguna herramienta, nos aparecerá que tiene habilitado el puerto.

Paso 3.1.2 Ataque y resultados.

En este paso vamos a tomar acción en cómo reacciona la víctima y el SOC.

Primero vamos a crear un directorio y dentro dos archivos.

```
└─(ruben@rrg)─[~]
  $ mkdir brute
  └─(ruben@rrg)─[~]
    $ cd brute
    └─(ruben@rrg)─[~/brute]
      $ touch user.txt
      $ touch pass.txt
      └─(ruben@rrg)─[~/brute]
        $ █
```

Para evitar confusiones, los ficheros `user.txt` y `pass.txt` empleados en la prueba de fuerza bruta fueron generados exclusivamente para este laboratorio y contienen únicamente **usuarios y contraseñas de ejemplo extremadamente simples**, diseñados con fines pedagógicos.

Estos ficheros **no se mostrarán en la documentación**, con el objetivo de evitar cualquier malinterpretación o uso indebido. Las credenciales utilizadas no corresponden a sistemas reales, no poseen valor práctico y no deben emplearse fuera de este entorno controlado.

Tras esto, antes de realizar la acción, debemos de meter una regla en el fichero de `local_rules.xml` de nuestro server SOC.

```
GNU nano 7.2                               /var/ossec/etc/rules/local_rules.xml *

<!-- Local rules -->

<!-- Modify it at your will. -->
<!-- Copyright (C) 2015, Wazuh Inc. -->

<!-- Example -->
<group name="local,syslog,sshd,>

  <!--
  Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
  -->
  <rule id="100001" level="5">
    <if_sid>5716</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>sshd: authentication failed from IP 1.1.1.1.</description>
    <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
  </rule>

</group>
<group name="windows,bruteforce">
  <!-- Correlation rule: triggers when rule 60112 has matched 5 times within 60 seconds -->
  <rule id="100100" level="10" frequency="5" timeframe="60">
    <if_matched_sid>60112</if_matched_sid>
    <description>Custom: 5 Windows failed logons within 60 seconds - Possible brute-force attack</description>
  </rule>
</group>
```

Guardamos...y reiniciamos el servicio.

```
[root@soc ~]# sudo systemctl restart wazuh-manager
usuario@soc:~$ sudo systemctl status wazuh-manager -l
sudo tail -n 200 /var/ossec/logs/ossec.log
● wazuh-manager.service - Wazuh manager
   Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-11-14 20:05:24 CET; 24ms ago
     Process: 4086 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
       Tasks: 208 (limit: 4548)
      Memory: 275.9M (peak: 281.9M)
        CPU: 12.38ls
      CGroup: /system.slice/wazuh-manager.service
              └─4148 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
                ├─4149 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
                ├─4150 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
                ├─4153 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
                ├─4156 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
                ├─4198 /var/ossec/bin/wazuh-authd
                └─4212 /var/ossec/bin/wazuh-db
```

```
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-processes-soc.
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-hotfixes-soc.
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-ports-soc.
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-hardware-soc.
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-protocols-soc.
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-interfaces-soc.
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-networks-soc.
2025/11/14 20:05:22 logger-helper: INFO: InventoryHarvesterFacade module started.
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-users-soc.
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-groups-soc.
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-browser-extensions-soc.
2025/11/14 20:05:22 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-inventory-services-soc.
2025/11/14 20:05:23 wazuh-modulesd:vulnerability-scanner: ERROR: Error opening the database: Error getting CNA Mapping content from rocksdb..
2025/11/14 20:05:23 wazuh-modulesd:vulnerability-scanner: INFO: Vulnerability scanner module started.
2025/11/14 20:05:23 wazuh-modulesd:syscollector: INFO: Evaluation finished.
2025/11/14 20:05:28 sca: INFO: Evaluation finished for policy '/var/ossec/ruleset/sca/cis_ubuntu24-04.yml'
2025/11/14 20:05:28 sca: INFO: Security Configuration Assessment scan finished. Duration: 7 seconds.
usuario@soc:~$
```

Ahora sivamos al Kali y a realizar la acción de ataque.

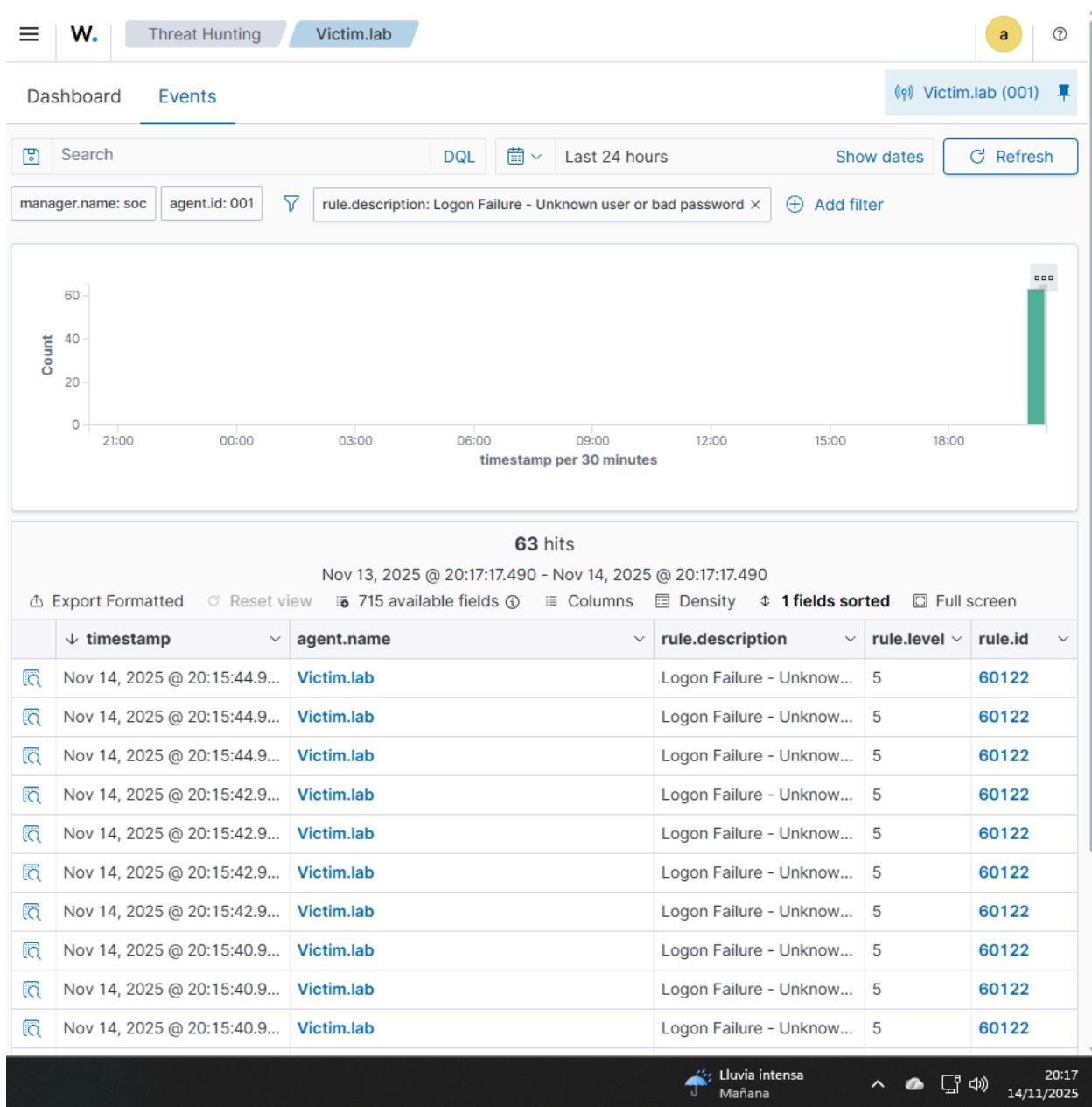


```
(ruben@rrg) [~/brute]
$ sudo hydra
```

*Por razones legales, no muestro el comando.

Y páramosporque no encontrará la contraseña.

Vamos a Threat Hunting, para ver los eventos de Windowsy....



Aquí lo tenemos.

Los intentos fallidos se muestran cómo Logon Failure, y así de fácil podemos ver el potencial de esta herramienta.

Aunque como alumno con interés por aprender....esto no se quedará así.

Implementaremos en el próximo paso algo como Virustotal, para que si se detecta un virus en algún dispositivo enrolado ... se detecte y salte un aviso.

Paso 4º Implementación de Virustotal en WAZUH

En este apartado se realiza la integración del SOC con el servicio VirusTotal únicamente con fines formativos.

Para garantizar la seguridad y evitar cualquier posibilidad de malinterpretación:

- **No se incluirá en esta memoria ningún enlace, ruta, muestra, hash, nombre de archivo ni referencia directa a los ficheros analizados.**
- Los archivos empleados en la prueba fueron usados **exclusivamente dentro del entorno cerrado del laboratorio**, sin ejecutarse y sin ser distribuidos.
- El objetivo de esta prueba es demostrar el funcionamiento de la integración de Wazuh con VirusTotal, **no manipular ni difundir malware real**.

El autor certifica que todas las actividades se han realizado en un entorno seguro, aislado y autorizado, sin representar riesgo para terceros.

Dicho esto, comenzamos la integración.

Primero, tenemos que ir a Virustotal para poder obtener una API key.

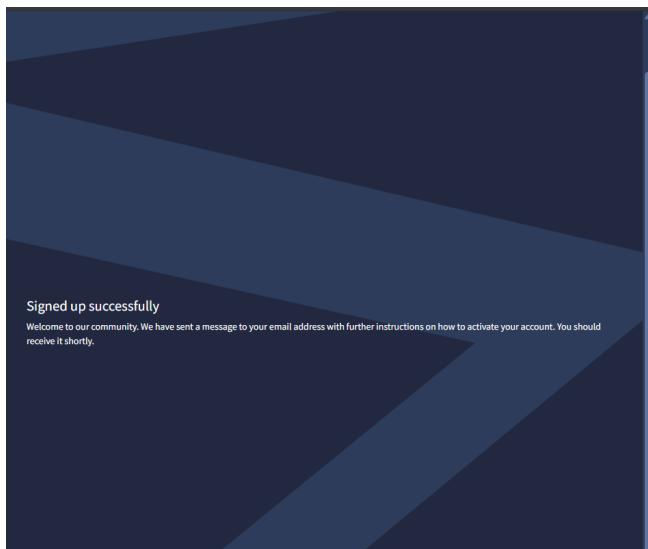




Damos en sign up, para registrarnos...

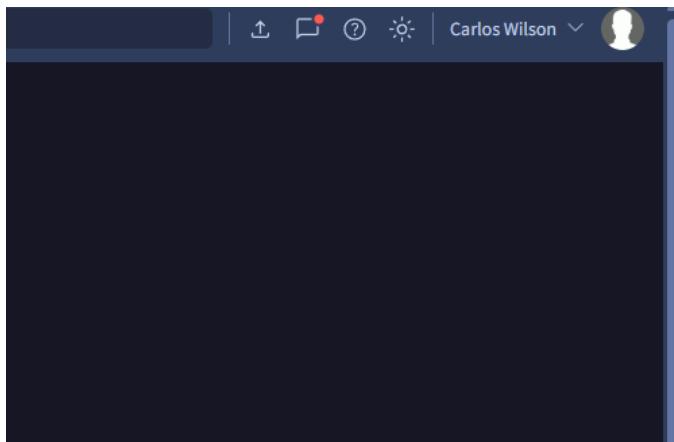
A screenshot of the VirusTotal sign-up form. The form is titled 'Join our community'. It contains fields for 'First name', 'Last name', 'Email', 'Username', 'Password', and 'Repeat password'. Each field has a placeholder text and a note below it. There is also a checkbox for accepting the 'Terms of Service' and a note about reading the 'Privacy Notice'. At the bottom, there is a 'Join us' button and an option to 'Or continue with' Google.

Metemos los datos...



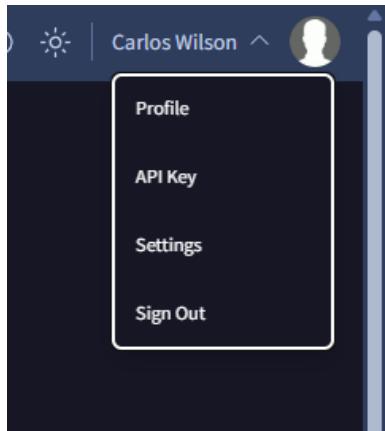
Tras ello, ya estaríamos dados de alta.

Importante, llegará un email de confirmacióndais en activate...y ya estaría listo.



Ya estoy registrado.

Para obtener la api key, damos click en el perfil...y nos saldrá esto...



This is your personal key. Do not disclose it to anyone that you do not trust, do not embed it in scripts or software from which it can be easily retrieved if you care about its confidentiality. By submitting data using your API key, you are agreeing to our [Terms of Service](#) and [Privacy Notice](#), and to the **sharing of your Sample submissions with the security community**. Please do not submit any personal information; we are not responsible for the contents of your submissions. [Learn more](#)

API QUOTA ALLOWANCES FOR YOUR USER

You own a standard free end-user account. It is not tied to any corporate group and so it does not have access to Premium services. You are subjected to the following limitations:

Access level	Limited , standard free public API	Upgrade to premium
Usage	Must not be used in business workflows, commercial products or services.	
Request rate	4 lookups / min	
Daily quota	500 lookups / day	
Monthly quota	15.5 K lookups / month	

Want to learn more about how our intelligence can supercharge your security operations? check our 360 overview brief.

Want to **upgrade your access?** Please do not hesitate to contact us, we'll go the extra mile to make you successful.

[Go premium](#) [Use in browser](#) [Discover feeds](#) [Other services](#)

API reference Python client Golang library Command-line interface

Tenemos un límite, por ser cuenta gratuita, pero nos sobra.

La api key está oculta, ya que es privada, cada uno tendrá una.

Ahora, nos vamos por ssh, a nuestro server SOC.

```
usuario@soc ~
GNU nano 7.2
/var/ossec/etc/ossec.conf

!--
Wazuh - Manager - Default configuration for ubuntu 24.04
More info at: https://documentation.wazuh.com
Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>15mc</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>

  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

  <!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->
  <logging>
    <log_format>plain</log_format>
  </logging>

  <remote>
    <connection>secure</connection>
    <port>1514</port>
    <protocol>tcp</protocol>
    <queue_size>131072</queue_size>
  </remote>

  <!-- Policy monitoring -->
  <rootcheck>
    <disabled>no</disabled>
    <check_files>yes</check_files>
    <check_trojans>yes</check_trojans>
    <check_dev>yes</check_dev>
    <check_sys>yes</check_sys>
    <check_pids>yes</check_pids>
    <check_ports>yes</check_ports>
    <check_if>yes</check_if>
  </rootcheck>

  <!-- Frequency that rootcheck is executed - every 12 hours -->
  <frequency>43200</frequency>

  <rootkit_files>etc/rootcheck/rootkit_files.txt</rootkit_files>
  <rootkit_trojans>etc/rootcheck/rootkit_trojans.txt</rootkit_trojans>
  <skip_nfs>yes</skip_nfs>
```

Y tenemos que modificar el fichero de políticas de WAZUH.

Tenemos que añadir un nuevo bloque...

```
GNU nano 7.2                               /var/ossec/etc/ossec.conf *
<!-- Frequency that rootcheck is executed - every 12 hours -->
<frequency>43200</frequency>

<rootkit_files>etc/rootcheck/rootkit_files.txt</rootkit_files>
<rootkit_trojans>etc/rootcheck/rootkit_trojans.txt</rootkit_trojans>

<skip_nfs>yes</skip_nfs>

<ignore>/var/lib/containerd</ignore>
<ignore>/var/lib/docker/overlay2</ignore>
</rootcheck>

<wodle name="cis-cat">
  <disabled>yes</disabled>
  <timeout>1800</timeout>
  <interval>1d</interval>
  <scan-on-start>yes</scan-on-start>

  <java_path>wodles/java</java_path>
  <ciscat_path>wodles/ciscat</ciscat_path>
</wodle>

<!-- Osquery integration -->
<wodle name="osquery">
  <disabled>yes</disabled>
  <run_daemon>yes</run_daemon>
  <log_path>/var/log/osquery/osqueryd.results.log</log_path>
  <config_path>/etc/osquery/osquery.conf</config_path>
  <add_labels>yes</add_labels>
</wodle>

/integration>
<name>virustotal</name>
<api_key>API KEY</api_key>
<group>syscheck</group>
<alert_format>json</alert_format>
/integration>

<!-- System inventory -->
<wodle name="syscollector">
  <disabled>no</disabled>
  <interval>lh</interval>
  <scan_on_start>yes</scan_on_start>
  <hardware>yes</hardware>
  <os>yes</os>
  <network>yes</network>
```

El bloque se pone antes de System inventory.

Tenemos que poner la api key de virustotal entre <api_key>

```
<api_key>API_KEY</api_key>
```

Guardamos, y reiniciamos servicio.

```
usuario@soc:~$ sudo nano /var/ossec/etc/ossec.conf
[sudo] password for usuario:
usuario@soc:~$ systemctl restart wazuh-manager
===== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units =====
Authentication is required to restart 'wazuh-manager.service'.
Authenticating as: ruben (usuario)
Password:
===== AUTHENTICATION COMPLETE =====
usuario@soc:~$
```

Tras modificar esto en el server, ahora hay que hacerlo en el agente.

Paso 4.1 Configurar Victim.lab.

En este vamos a configurar la Victim.lab para que pueda detectar ficheros maliciosos, lo suyo es no poner el disco duro entero, porque generaría muchas alertas, entonces vamos a tomar los directorios más comunes.

```
<!-- Monitoreo realista de carpetas de usuario -->

<directories check_all="yes" realtime="yes">C:\Users\*\Downloads</directories>
<directories check_all="yes" realtime="yes">C:\Users\*\Desktop</directories>
<directories check_all="yes"
realtime="yes">C:\Users\*\AppData\Local\Temp</directories>
```

Vamos a modificar el fichero de ossec.conf

```
C:\Windows\system32>notepad C:\Program Files (x86)\ossec-agent\ossec.conf
C:\Windows\system32>

```

Ponemos las líneas...

```
<!-- Monitoreo realista de carpetas de usuario -->
<directories check_all="yes" realtime="yes">C:\Users\*\Downloads</directories>
<directories check_all="yes" realtime="yes">C:\Users\*\Desktop</directories>
<directories check_all="yes" realtime="yes">C:\Users\*\AppData\Local\Temp</directories>
```

Reiniciamos servicio...

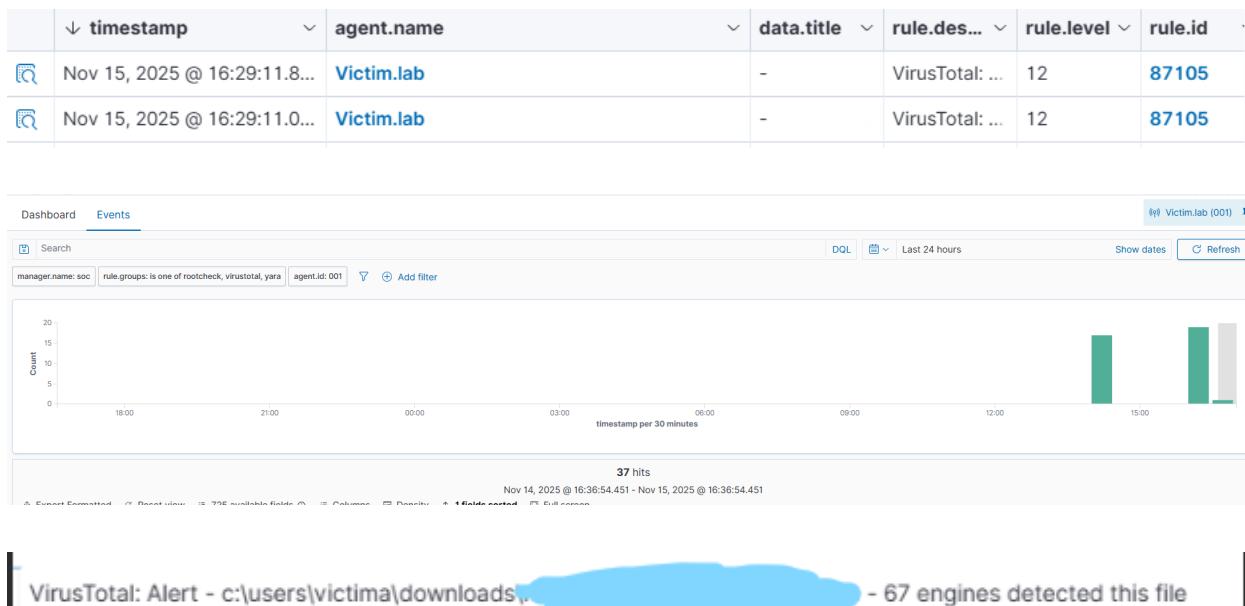
```
PS C:\Windows\system32>
PS C:\Windows\system32> # Iniciar el servicio
PS C:\Windows\system32> Start-Service -Name "WazuhSvc"
PS C:\Windows\system32>
PS C:\Windows\system32> # O reiniciarlo en un solo comando
PS C:\Windows\system32> Restart-Service -Name "WazuhSvc"
PS C:\Windows\system32>
```

Paso 4.1.2 Prueba de implantación.

En este paso, vamos a comprobar que funciona la implantación.

Para que funcione, deshabilitar el windows defender, sólo para esta prueba, nunca hay que desactivar las medidas de seguridad

Nos vamos a Malware Detection...



Nos salta la alerta de que se encontró un fichero malicioso en el sistema.

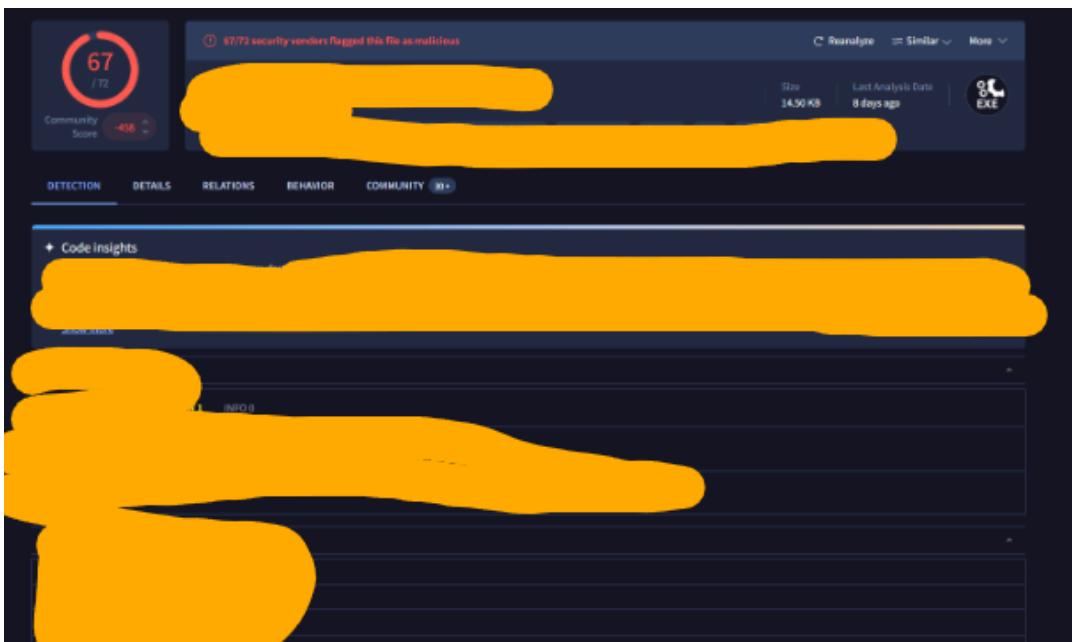


En este símbolo, le damos, y nos aparecerá esto, que es el reporte...

Document Details [View surrounding documents](#) [View single document](#)

[Table](#) [JSON](#)

✓ _index	wazuh-alerts-4.x-2025.11.15
✓ agent.id	001
✓ agent.ip	192.168.75.137
✓ agent.name	Victim.lab
✓ data.integration	virustotal
✓ data.virustotal.found	1
✓ data.virustotal.malicious	1
✓ data.virustotal_permalink	[REDACTED]



*Censuro la información, pero se puede apreciar.

Aquí está la prueba, lo han detectado como malicioso 67 antivirus de la base de datos de Virustotal.

CONCLUSIÓN:

Este laboratorio ha demostrado la importancia de contar con un sistema de monitorización y detección de incidentes en un entorno de producción. La implementación de Wazuh permite centralizar la recolección de eventos, analizar alertas de seguridad y mantener registros confiables de la actividad del sistema.

Se ha comprobado que una infraestructura de seguridad bien configurada facilita la **detección temprana de incidentes**, la **respuesta eficaz ante ataques** y contribuye al **cumplimiento de políticas y normativas de seguridad**.

En definitiva, contar con un SOC operativo y herramientas como Wazuh aumenta la **resiliencia de la infraestructura** y proporciona un entorno más seguro y controlado frente a amenazas externas e internas.



