TRABAJO FIN DE TÍTULO: DISEÑO Y DESPLIEGUE DE UNA INFRAESTRUCTURA DE CLÚSTER QUE OFRECE UNA HERRAMIENTA DE MONITORIZACIÓN DE EVENTOS DE SEGURIDAD EN ALTA DISPONIBILIDAD EN CENTOS 7 Y CON BALANCEO DE CARGA EN NGINX

Daniel Herrero Averchenko

Tutor: Francisco Alexis Quesada Arencibia

JUNIO DE 2024
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Grado Ingeniería Informática

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que han hecho posible este Trabajo de Fin de Grado.

A mi tutor, Francisco Alexis Quesada Arencibia, por su constante apoyo y guía. A los profesores de la Escuela de Ingeniería Informática de la Universidad de Las Palmas de Gran Canaria por su dedicación y enseñanzas.

A mis compañeros y amigos por su apoyo moral y académico.

Y, sobre todo, a mi familia. A mis padres, por su amor incondicional, comprensión y apoyo constante. Gracias por ser mi inspiración y motivación diaria. Su aliento y paciencia han sido fundamentales para alcanzar este logro.

A todos ustedes, muchas gracias.

Resumen

En este TFT se desplegará la herramienta Wazuh para monitorizar la seguridad de los agentes conectados al clúster de Wazuh a través del balanceador de carga Nginx, ofreciendo alta disponibilidad.

La instalación y configuración de las máquinas virtuales se realizará en VirtualBox, creando primero una Red NAT virtual. Cuatro máquinas virtuales se configurarán como clúster de Wazuh: Indexer (con Wazuh indexer y dashboard), Nodo1 (con Wazuh server como nodo master), Nodo2 y Nodo3 (con Wazuh server como nodos worker).

Nginx balanceará las conexiones de los agentes a los nodos worker, ofreciendo alta disponibilidad en caso de caída de uno de los nodos worker. Se instalarán cuatro máquinas virtuales para los agentes (Agente1, Agente2, Agente3 y Agente4) y una con Kali Linux 2024.1 para realizar ataques sobre los agentes y el clúster.

Finalmente, desde el Wazuh dashboard se monitorizarán todos los eventos generados en los agentes y nodos.

Abstract

In this TFT, the Wazuh tool will be deployed to monitor the security of agents connected to the Wazuh cluster through the Nginx load balancer, providing high availability.

The installation and configuration of the virtual machines will be done in VirtualBox, first creating a virtual NAT network. Four virtual machines will be configured as a Wazuh cluster: Indexer (with Wazuh indexer and dashboard), Node1 (with Wazuh server as master node), Node2, and Node3 (with Wazuh server as worker nodes).

Nginx will balance the connections of the agents to the worker nodes, providing high availability in case one of the worker nodes fails. Four virtual machines will be installed for the agents (Agent1, Agent2, Agent3, and Agent4) and one with Kali Linux 2024.1 to carry out attacks on the agents and the cluster.

Finally, all events generated on the agents and nodes will be monitored from the Wazuh dashboard.

Índice de contenidos

Índice de imágenes

Índice de Tablas

1. Introducción y objetivos

1.1 Introducción

Hoy en día, las organizaciones se enfrentan cada vez más a un mayor número ataques cibernéticos que representan una constante amenaza para la seguridad y privacidad de sus datos corporativos. Los Sistemas de Gestión de Información y Eventos de Seguridad (SIEM) [1] emergen como herramientas vitales proporcionando capacidades necesarias para detectar, analizar y responder a incidentes de seguridad de manera eficaz y en tiempo real, garantizando así la seguridad no solo de la información, sino también de las personas que dependen de ella.

Estos sistemas no solo cuidan de las redes y los datos críticos, sino que también protegen las tareas diarias de quienes dependen de estas tecnologías para su trabajo. Al monitorear continuamente y analizar la información de la red, los SIEM proporcionan una visión clara y actualizada de la seguridad, lo que es crucial para mantener un entorno de trabajo seguro y confiable.

La implementación de SIEM no solo refuerza la seguridad contra los ciberataques, sino que también fortalece la confianza de los empleados en sus herramientas tecnológicas, promoviendo un ambiente donde la seguridad y la eficiencia van de la mano. Esta capacidad de mantener seguras las operaciones es fundamental para el éxito empresarial en nuestro mundo interconectado y competitivo.

En este trabajo se va a desplegar un SIEM utilizando la herramienta *Wazuh*. *Wazuh* es una plataforma de seguridad gratuita y de código abierto [2]. Esta herramienta ayuda a organizaciones e individuos a proteger sus activos contra amenazas de seguridad. Proporciona análisis de datos de registro, detección de intrusiones y malware, monitoreo de la integridad de archivos, evaluación de la configuración, detección de vulnerabilidades y soporte para el cumplimiento de la normativa [3].

Los componentes centrales de *Wazuh* son [3]:

- *Wazuh indexer*: es un motor de análisis y búsqueda de texto completo altamente escalable. Indexa y almacena alertas generadas por el *Wazuh server*.
- *Wazuh server*: analiza los datos recibidos de los agentes. Los procesa a través de decodificadores y reglas, utilizando inteligencia de amenazas para buscar indicadores de compromiso (IOC) bien conocidos. Un solo servidor puede analizar los datos de cientos de agentes, y escalar horizontalmente cuando se configura como un clúster. Este componente central también se utiliza para gestionar los agentes, configurándolos y actualizándolos a distancia cuando es necesario.
- *Wazuh dashboard*: es la interfaz de web del usuario para la visualización y el análisis de datos. Incluye paneles listos para usar para eventos de seguridad, aplicaciones vulnerables detectadas, datos de monitoreo de integridad de archivos, resultados de evaluación de configuración, eventos de monitoreo de

infraestructura en la nube y otros. También se utiliza para gestionar la configuración de *Wazuh* y supervisar su estado.

- *Wazuh* agent: los agentes se instalan en *endpoints* como portátiles, ordenadores de sobremesa, servidores, instancias en la nube o máquinas virtuales. Proporcionan funciones de prevención, detección y respuesta ante amenazas. Funcionan en sistemas operativos como *Linux*, Windows, macOS, Solaris, AIX y HP-UX.

1.2 Comparación de *Wazuh* con otras herramientas

A continuación, se pasará a comparar la herramienta *Wazuh* con otras herramientas de monitorización de eventos de seguridad para justificar finalmente su elección parta este trabajo.

- *Wazuh* vs *OpenVAS: OpenVAS* [4] es una herramienta de evaluación de vulnerabilidades y gracias a su escáner se obtienen pruebas para detectar vulnerabilidades. No incluye funcionalidades de monitorización continua o respuesta a incidentes, mientras que *Wazuh* incluye detección de intrusiones, monitorización en tiempo real, respuesta a incidentes, y cumplimiento de normativas.
- *Wazuh* vs *AlienVault OSSIM*: *AlienVault OSSIM* [5] múltiples funciones de seguridad y gestión de vulnerabilidades. Ofrece una solución todo en uno para la seguridad de la información. *Wazuh* en cambio es más modular y flexible, permitiendo una personalización más profunda.
- *Wazuh* vs *Splunk*: *Splunk* [6] es una potente herramienta para la gestión y análisis de grandes volúmenes de datos, incluyendo datos de seguridad, mientras que *Wazuh* ofrece además capacidades como la detección de amenazas y respuestas a incidentes.
- *Wazuh* vs *Cisco SecureX*: *Cisco SecureX* [7] proporciona una plataforma de seguridad integrada con capacidades amplias en protección de *endpoints*, análisis de amenazas, y automatización. En cambio, *Wazuh* al ser de código abierto es una solución más accesible y flexible, adecuada para organizaciones de cualquier tamaño y con un presupuesto más limitado.

Una vez realizadas las comparaciones, se justifica la elección de *Wazuh* considerando los siguientes puntos clave:

- Flexibilidad y código abierto: *Wazuh* es altamente personalizable y se adapta bien a una variedad de entornos y necesidades específicas gracias a su naturaleza de código abierto.
- Integración y Ampliación: La capacidad de *Wazuh* para integrarse con otras herramientas, como *Elasticsearch* para análisis de datos o *OpenVAS* para evaluación de vulnerabilidades, lo hace una solución versátil que puede crecer y adaptarse a las necesidades cambiantes de tu organización.
- Completo en Seguridad Operativa: A diferencia de herramientas como *OpenVAS* o *Splunk*, *Wazuh* ofrece un enfoque integral que no solo identifica

problemas, sino que también ayuda a gestionar y responder a incidentes de seguridad.

Wazuh proporciona un balance óptimo entre profundidad de funcionalidades de seguridad, flexibilidad y costo, lo que lo convierte en una elección excelente para mejorar la seguridad sin comprometer la versatilidad y el presupuesto.

1.4 Esquema clúster de Wazuh, balanceador de carga Nginx y los agentes

Se ha realizado un esquema tanto de las máquinas que contienen a los principales componentes y que conforman el clúster de *Wazuh*, como de los agentes de *Wazuh* conectados al clúster través del balanceador de carga, para visualizar y entender un poco mejor como será el clúster y la conexión entre los diferentes componentes que conformarán el servidor [Imagen 1].

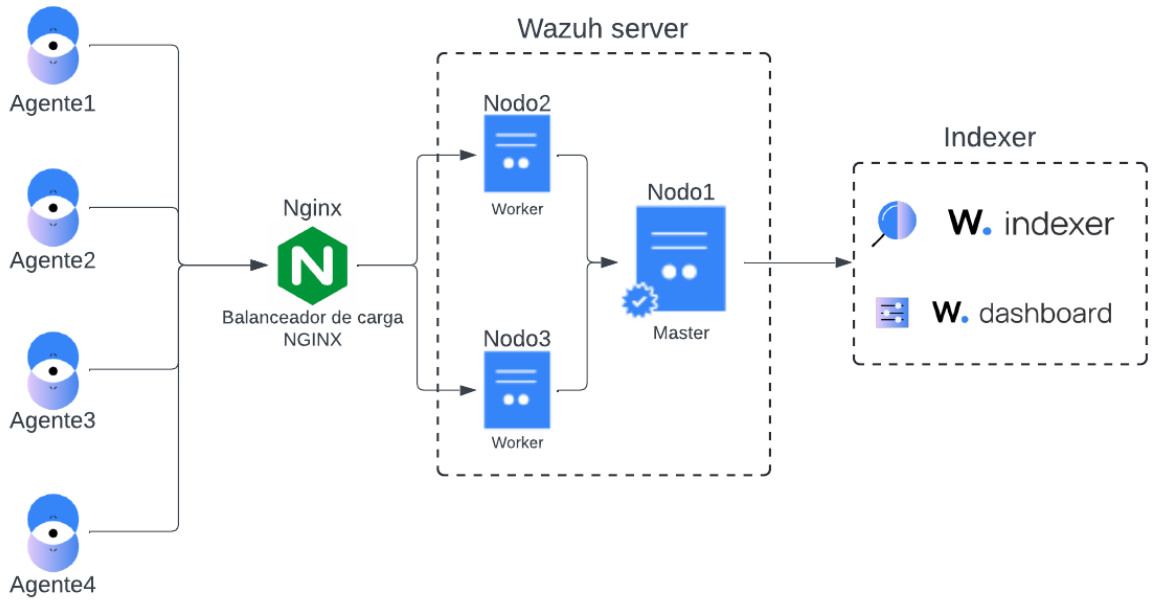


Imagen 1. *Esquema cluster Wazuh*

1.5 Objetivos

El objetivo principal de este trabajo es el despliegue de un SIEM utilizando la herramienta *Wazuh* para monitorizar eventos de seguridad en diferentes agentes con balanceo de carga y ofreciendo un servicio en alta disponibilidad.

Para ello, se desplegarán una serie de máquinas virtuales donde se instalarán la herramienta de *Wazuh*, el balanceador de carga *Nginx* y los diferentes agentes. El balanceador de carga se encargará de distribuir las conexiones de los agentes entre los nodos *worker* del servidor y ofrecer un servicio en alta disponibilidad en caso de caída de uno de los nodos.

Finalmente, realizar pruebas de ataques a los diferentes agentes para posteriormente observar y analizar los resultados obtenidos con la monitorización de la herramienta *Wazuh*.

2. Competencias específicas

Las competencias específicas cubiertas en este TFT se dividen en dos categorías, las competencias específicas Común a la Ingeniería Informática (CII) y las competencias específicas de las Tecnologías de la Información (TI).

2.1 Común a la Ingeniería Informática (CII)

CII010. "Conocimiento de las características, funcionalidades y estructura de los Sistemas Operativos y diseñar e Implementar aplicaciones basadas en sus servicios."

Esta competencia ha sido cubierta con la instalación y configuración de los sistemas operativos usados a lo largo del proyecto, así como de la instalación y configuración de la herramienta Wazuh atendiendo a las características del sistema operativo.

CII11. "Conocimiento y aplicación de las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores e Internet y diseñar e implementar aplicaciones basadas en ellas."

Esta competencia ha sido cubierta con la creación y configuración de una red NAT, así como de la configuración de red de las máquinas para su posible interconexión dentro de esa red.

2.2 Tecnologías de la Información (TI)

TI01. "Capacidad para comprender el entorno de una organización y sus necesidades en el ámbito de las tecnologías de la información y las comunicaciones."

Esta competencia ha sido cubierta con la redacción de la importancia de la implantación de un SIEM en las organizaciones.

TI02. "Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados."

Esta competencia ha sido cubierta con el diseño y despliegue de una infraestructura en clúster que utiliza la herramienta Wazuh. Se demuestra la capacidad de gestionar un sistema que ofrece un servicio de monitorización de eventos de seguridad de calidad sin costos adicionales, ya que Wazuh es una plataforma gratuita y de código abierto.

TI04. “Capacidad para seleccionar, diseñar, desplegar, integrar y gestionar redes e infraestructuras de comunicaciones en una organización.”

Esta competencia ha sido cubierta con la creación y configuración de una red NAT, así como de las conexiones entre los equipos que se han configurado en esa red a través de la configuración y gestión de las redes para asegurar la comunicación entre los nodos del clúster y agentes.

TI05. “Capacidad para seleccionar, desplegar, integrar y gestionar sistemas de información que satisfagan las necesidades de la organización, con los criterios de coste y calidad identificados.”

Esta competencia ha sido cubierta con la herramienta Wazuh que ofrece una monitorización de eventos de calidad y coste gratuito.

TI06. “Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.”

Esta competencia ha sido cubierta con la integración de la herramienta Wazuh que proporciona una interfaz web intuitiva para visualizar y analizar datos. La monitorización en tiempo real muestra el uso de tecnologías de red avanzadas.

TI07. “Capacidad para comprender, aplicar y gestionar la garantía y seguridad de los sistemas informáticos.”

Esta competencia ha sido cubierta con el despliegue de la herramienta Wazuh, que analiza datos, detecta intrusiones, supervisa la integridad de archivos y encuentra vulnerabilidades, mostrando la habilidad de implementar y gestionar medidas de seguridad efectivas para proteger los sistemas informáticos.

3. Creación y configuración de una MV *CentOS 7* en *VirtualBox*

Se creará una máquina virtual a partir de una imagen *ISO* de *CentOS 7*. El uso de una versión obsoleta es debido a que este proyecto se empezó hace tiempo y en aquel entonces no se trataba de una versión obsoleta de *CentOS*.

3.1 Descarga de una imagen *ISO* de CentOS 7

Para la descarga de una imagen de disco *ISO* del sistema operativo *CentOS 7*, se accederá al navegador de internet y se introducirá la siguiente URL [8] "*https://www.centos.org/download/*" donde se mostrará la siguiente página web [Imagen 2].

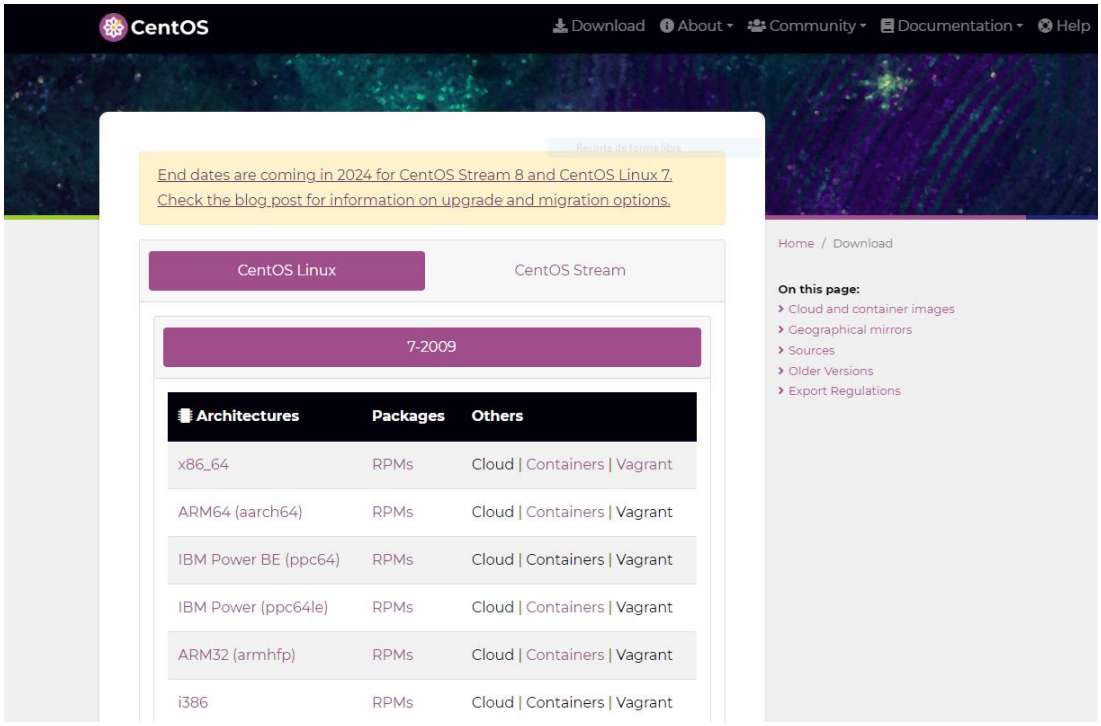


Imagen 2. *Página de descargas de CentOS*

A Continuación, se seleccionará la arquitectura x86_64 que llevará a la siguiente página [Imagen 3].

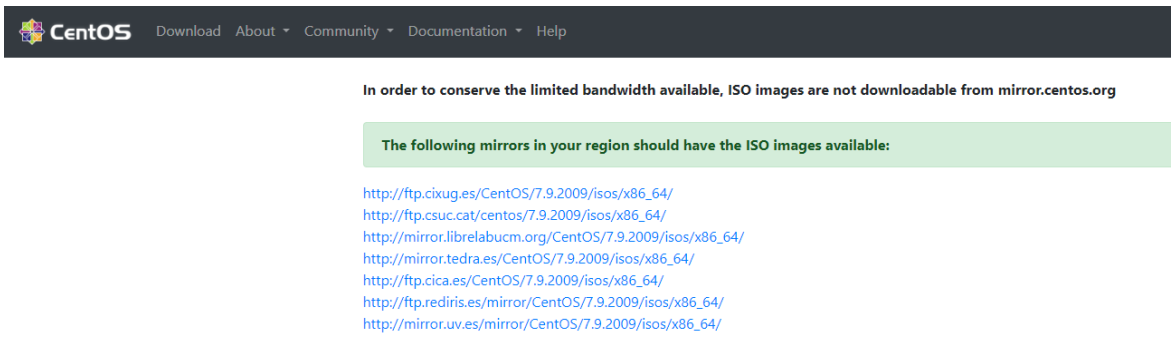


Imagen 3. *Página de descargas de imágenes de CentOS de arquitectura x86_64*

Donde se seleccionará la siguiente URL "*http://ftp.cica.es/CentOS/7.9.2009/isos/x86_64/*" , y se seleccionará el tipo de imagen *ISO* [Imagen 4] que se quiera descargar. En este caso *CentOS-7-x86_64-DVD-2009ISO* y se iniciará la descarga de la imagen de disco del sistema operativo *CentOS 7* a instalar en una máquina virtual posteriormente.

Index of /CentOS/7.9.2009/isos/x86_64/

../		
0_README.txt	04-Aug-2022 18:03	2740
CentOS-7-x86_64-DVD-2009.iso	04-Nov-2020 11:37	4G
CentOS-7-x86_64-DVD-2009.torrent	06-Nov-2020 14:44	176K
CentOS-7-x86_64-DVD-2207-02.iso	26-Jul-2022 15:10	4G
CentOS-7-x86_64-Everything-2009.iso	02-Nov-2020 15:18	10G
CentOS-7-x86_64-Everything-2009.torrent	06-Nov-2020 14:44	381K
CentOS-7-x86_64-Everything-2207-02.iso	26-Jul-2022 18:09	10G
CentOS-7-x86_64-Minimal-2009.iso	03-Nov-2020 14:55	973M
CentOS-7-x86_64-Minimal-2009.torrent	06-Nov-2020 14:44	39K
CentOS-7-x86_64-Minimal-2207-02.iso	26-Jul-2022 15:10	988M
CentOS-7-x86_64-NetInstall-2009.iso	26-Oct-2020 16:26	575M
CentOS-7-x86_64-NetInstall-2009.torrent	06-Nov-2020 14:44	23K
sha256sum.txt	04-Aug-2022 17:56	703
sha256sum.txt.asc	04-Aug-2022 17:58	1563

Imagen 4. *Página que muestra los tipos de ISO a descargar*

3.2 Creación de la MV

Para la creación y administración de las máquinas virtuales en este trabajo se utilizará programa de virtualización *VirtualBox*. Una vez ejecutado *VirtualBox* se seleccionará la opción *"Nueva"* [Imagen 5].

Imagen 5. *Inicio VirtualBox*

A continuación, se abrirá una ventana llamada "*Crear máquina virtual*" donde se introducirán los datos de Nombre y sistema operativo de la máquina virtual [Imagen 6], también se seleccionará la opción "*Omitir instalación desatendida*". Una vez introducidos todos los campos se hará *clic* en el botón "*Siguiente*".

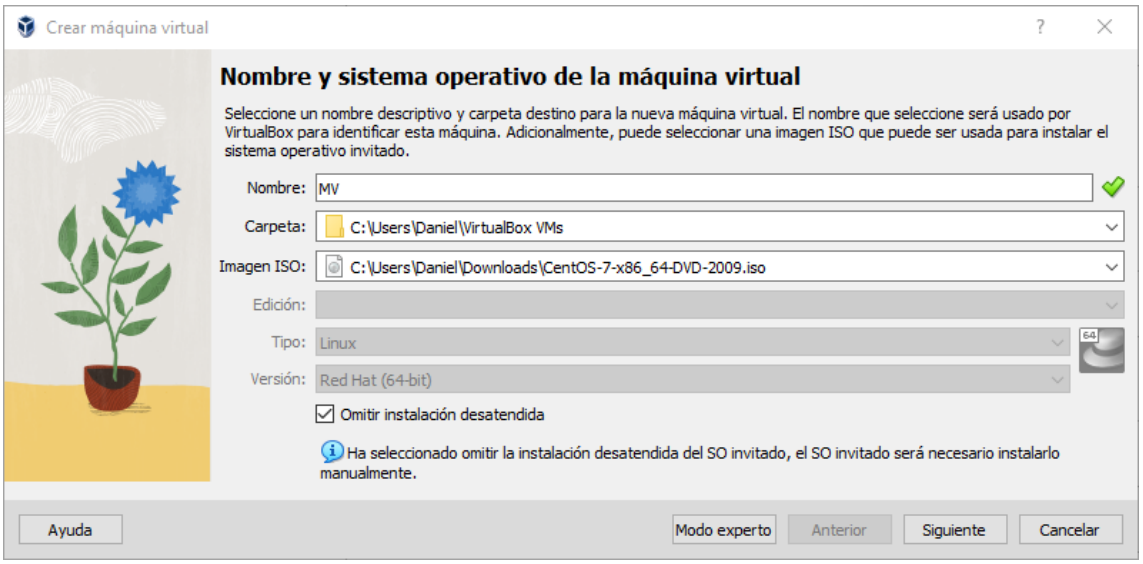


Imagen 6. *Nombre y SO de la MV*

Se pasará a modificar el hardware de la máquina virtual cambiando la cantidad de memoria RAM y número de CPU virtuales [Imagen 7]. Para la memoria base RAM se añadirán 2048 MB y para el número de procesadores se añadirán 2 CPU. Una vez editados estos parámetros se hará clic en el botón "*Siguiente*".

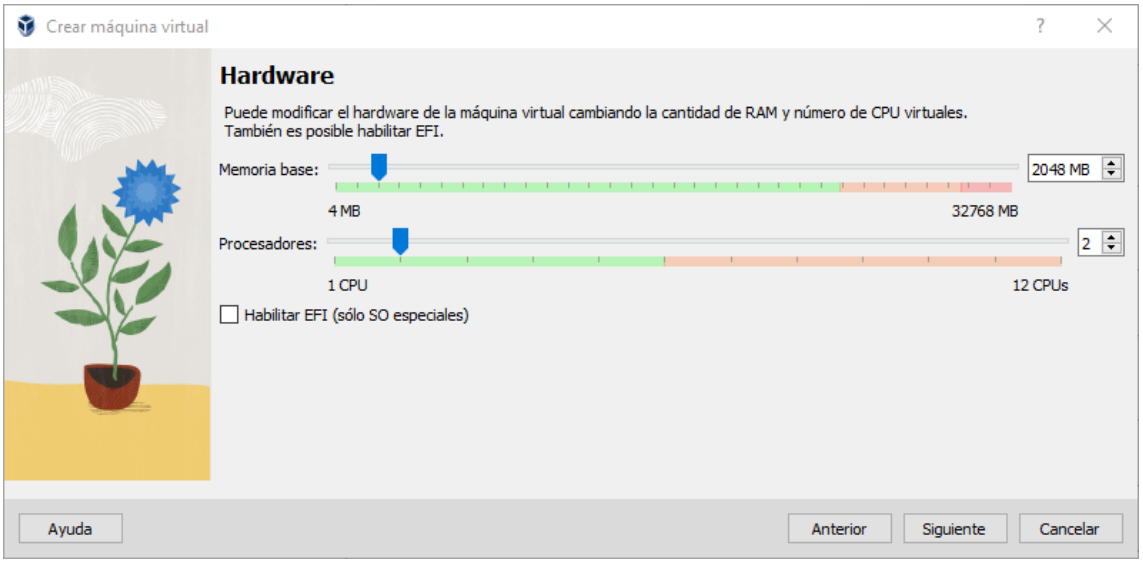


Imagen 7. *Hardware de la MV*

Para el disco duro virtual se seleccionará la opción "Crear un disco duro virtual ahora" y se añadirán 50 GB para el tamaño del disco [Imagen 8]. Una vez introducidos estos parámetros se hará *clic* en la opción "*Siguiente*".

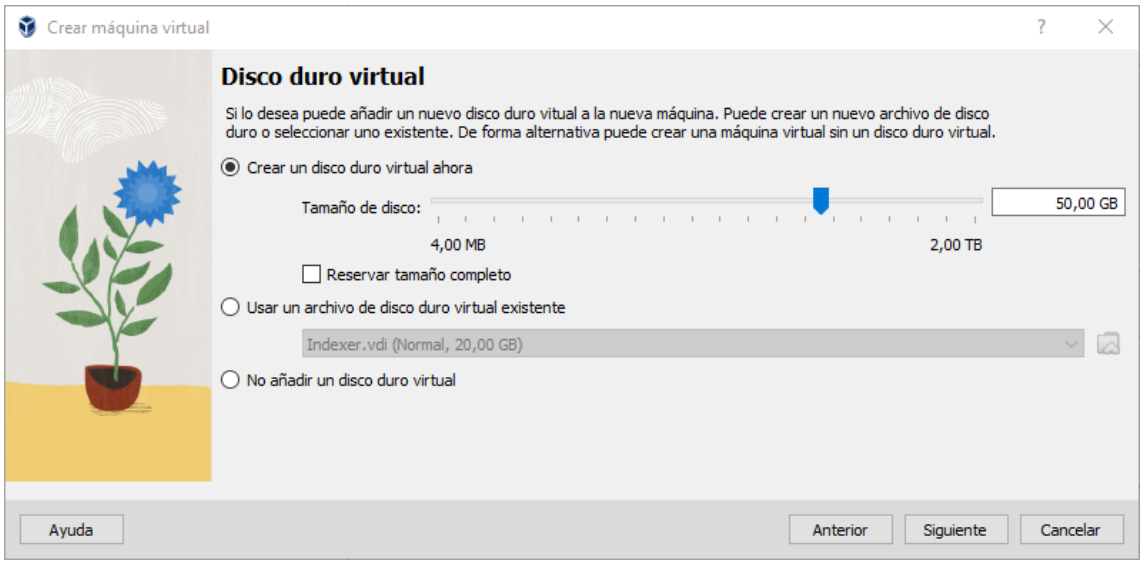


Imagen 8. *Disco duro virtual de la MV*

Una vez completados todos los pasos anteriores aparecerá un resumen de los parámetros introducidos para la creación de nuestra nueva máquina virtual [Imagen 9] que se revisarán para ver si los datos introducidos son los correctos y una vez verificado esto se hará *clic* en el botón "*Terminar*".

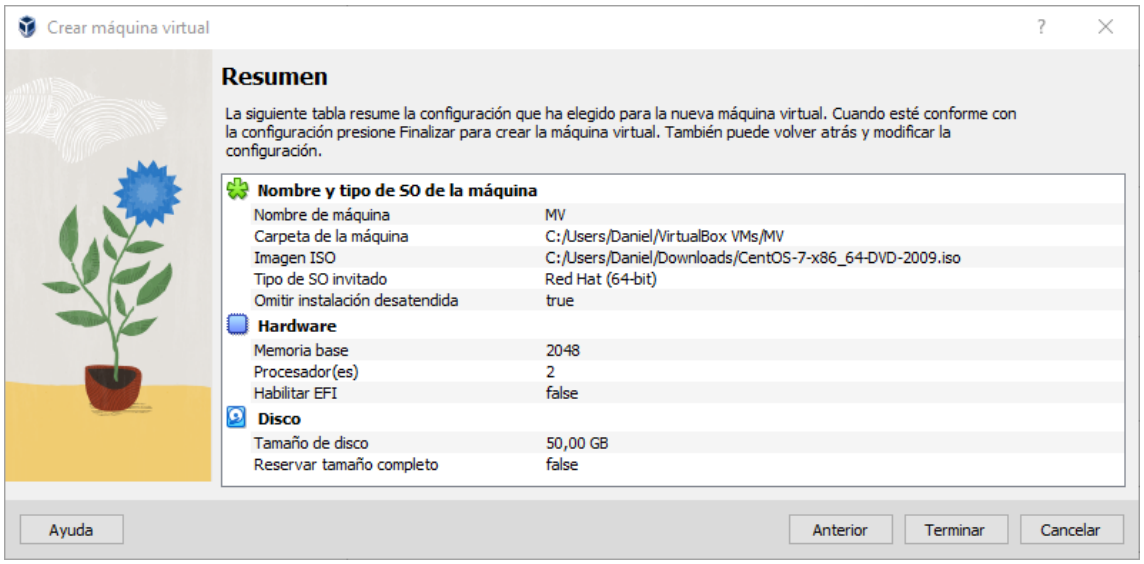


Imagen 9. *Resumen de características de la MV*

Se dispondrá de una nueva máquina virtual con el nombre que se le haya elegido para su creación en el menú de inicio de *VirtualBox* [Imagen 10]. Para iniciar la nueva máquina virtual se hará doble *clic* sobre ella. Una vez arrancada la máquina virtual se ejecutará el disco *ISO* añadido previamente con la instalación del *SO* (*SO – sistema operativo*).

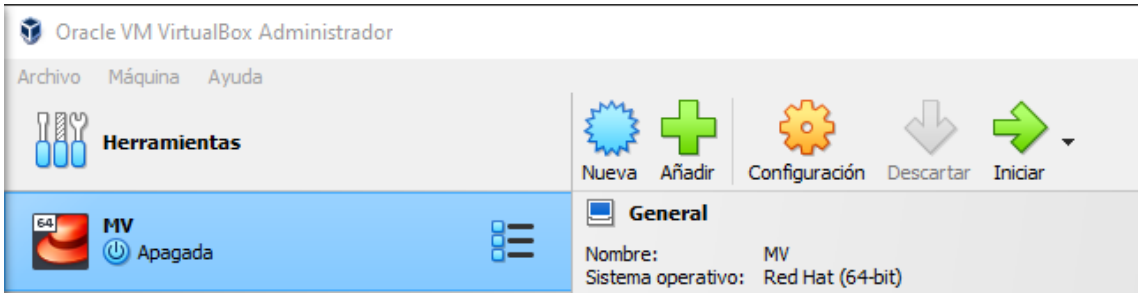


Imagen 10. *MV en el inicio de VirtualBox*

Una vez iniciada la máquina virtual se ejecutará la instalación del *SO* elegido previamente con la descarga de la imagen *ISO* [Imagen 11]. Se seleccionará la opción "*Test this media & install CentOS 7*".

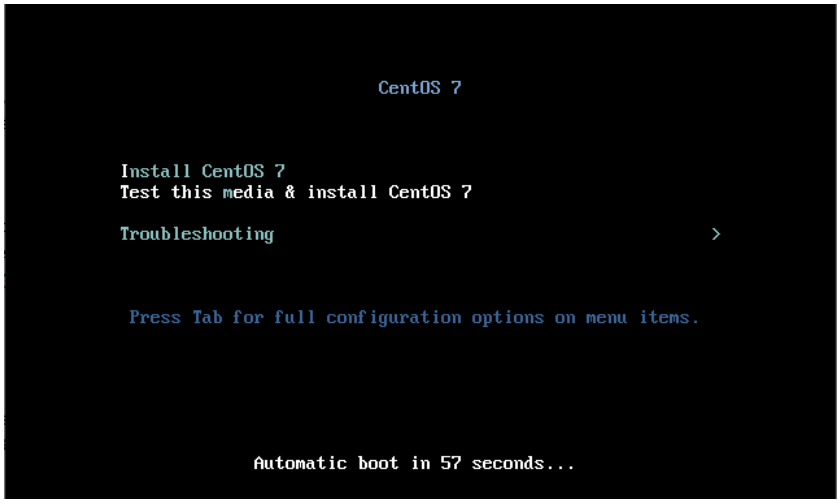


Imagen 11. *Instalación CentOS 7*

Se seleccionará el idioma del *SO* [Imagen 12] y a continuación se seleccionarán una serie de parámetros antes de empezar la instalación.

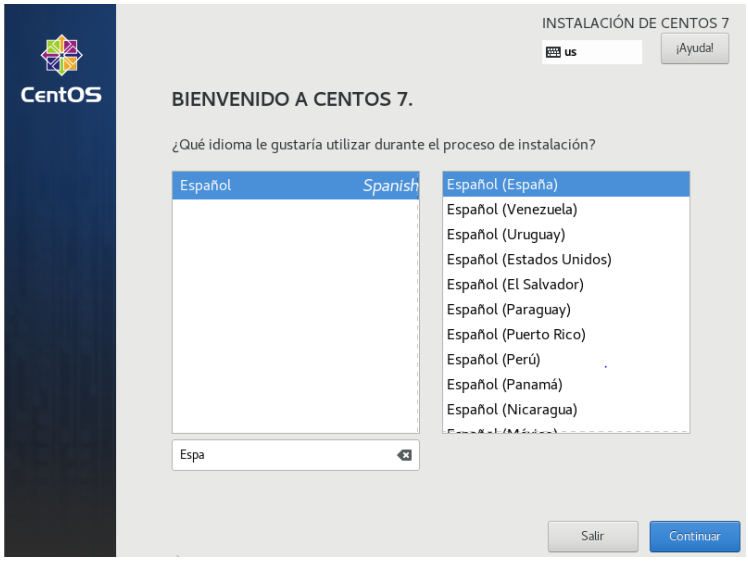


Imagen 12. *Selección de idioma*

En "*SELECCIÓN DE SOFTWARE*" se seleccionará la opción "*Escritorio Gnome*" [Imagen 13] el cual instala el sistema operativo con un entorno gráfico fácil de manejar.

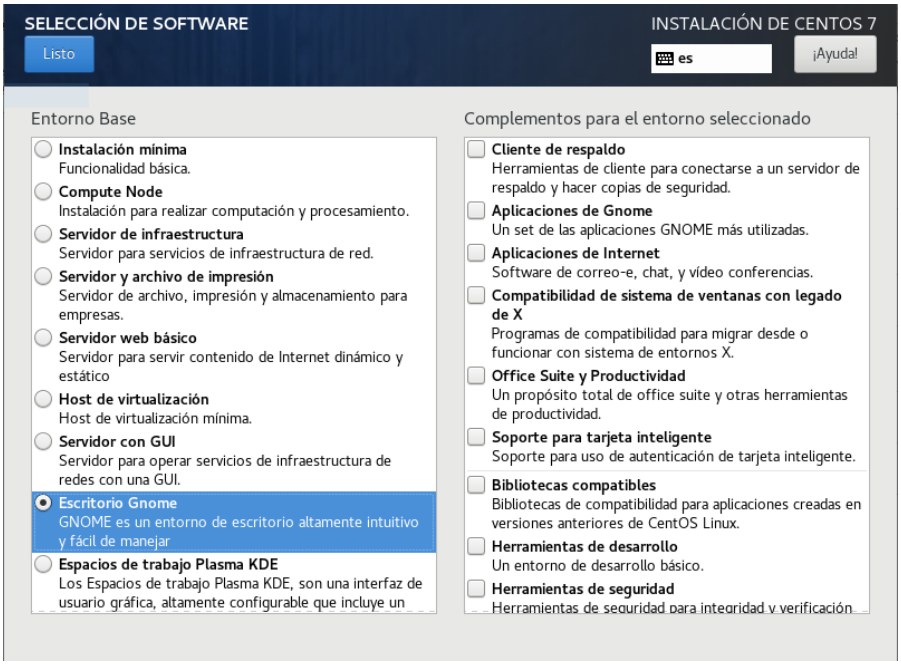


Imagen 13. *Selección de software*

En "DESTINO DE LA INSTALACIÓN" [Imagen 14] se mantendrán los parámetros por defecto manteniendo la opción de particionado automático y se hará *clic* en el botón "Listo".

Imagen 14. *Destino de la instalación*

En “RED & NOMBRE DE EQUIPO” [Imagen 15] se activará la tarjeta de red “*Ethernet (enp0s3)*” y se hará *clic* en el botón “*Listo*”.

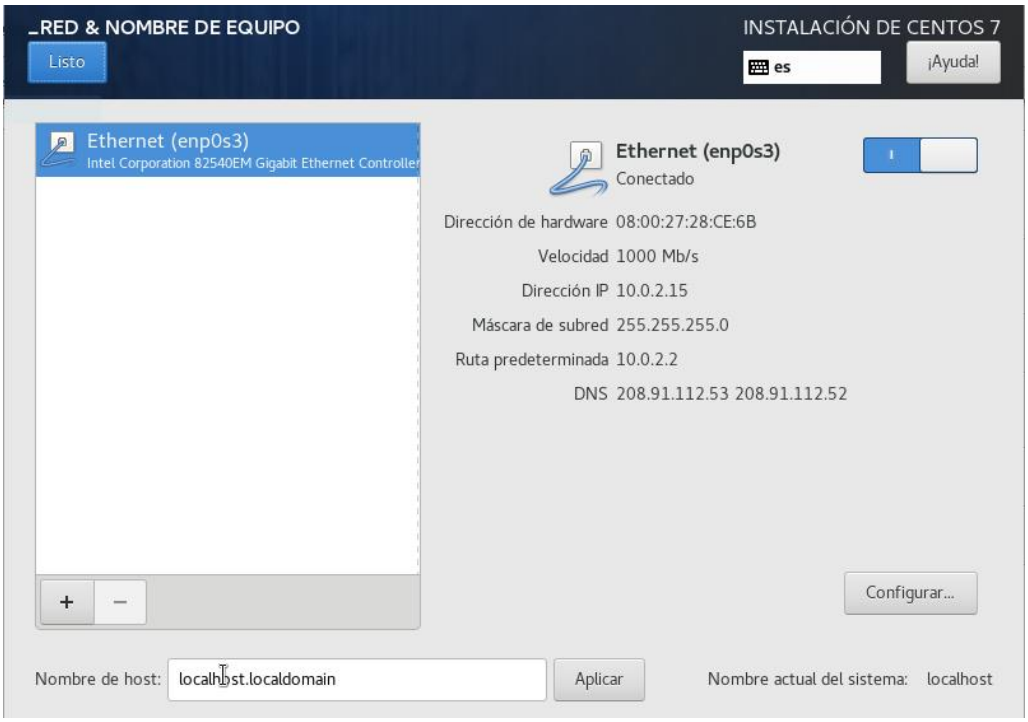


Imagen 15. *Red y nombre del equipo*

Editadas todas las opciones se hará *clic* en “Empezar instalación” [Imagen 16] y comenzará la instalación de nuestro *SO*.

Imagen 16. *Resumen de la instalación*

Mientras se instala el *SO* se creará la contraseña de *root* y se creará un usuario con su respectiva contraseña [Imagen 17], una vez completada la instalación se hará *clic* en el botón "*Reiniciar*".

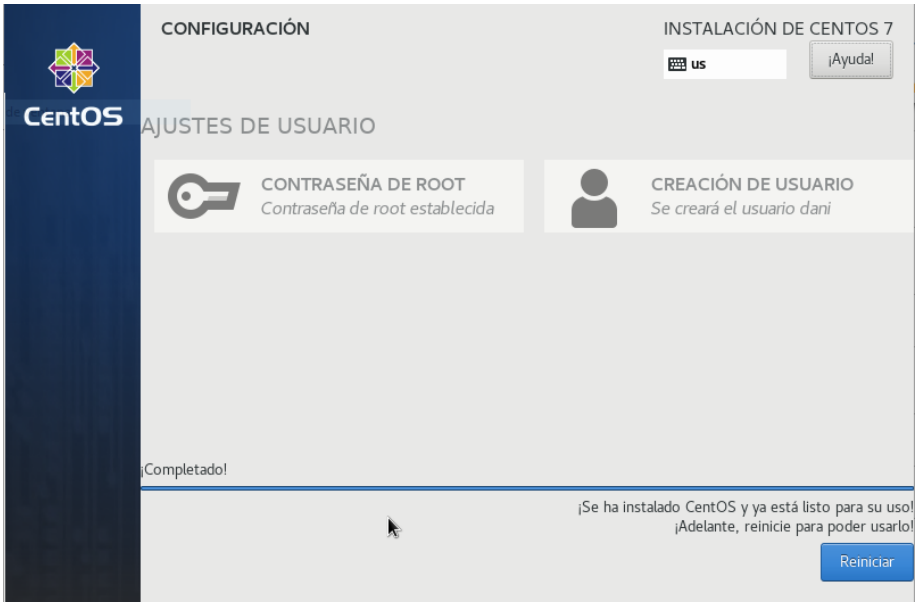


Imagen 17. *Configuración de usuario y contraseñas*

Reiniciada la máquina virtual se accederá al apartado *"LICENSE INFORMATION"*, se aceptará el acuerdo de licencia marcando la casilla *"Acepto el acuerdo de licencia"* [Imagen 18] y se hará *clic* en el botón *"Listo"*.

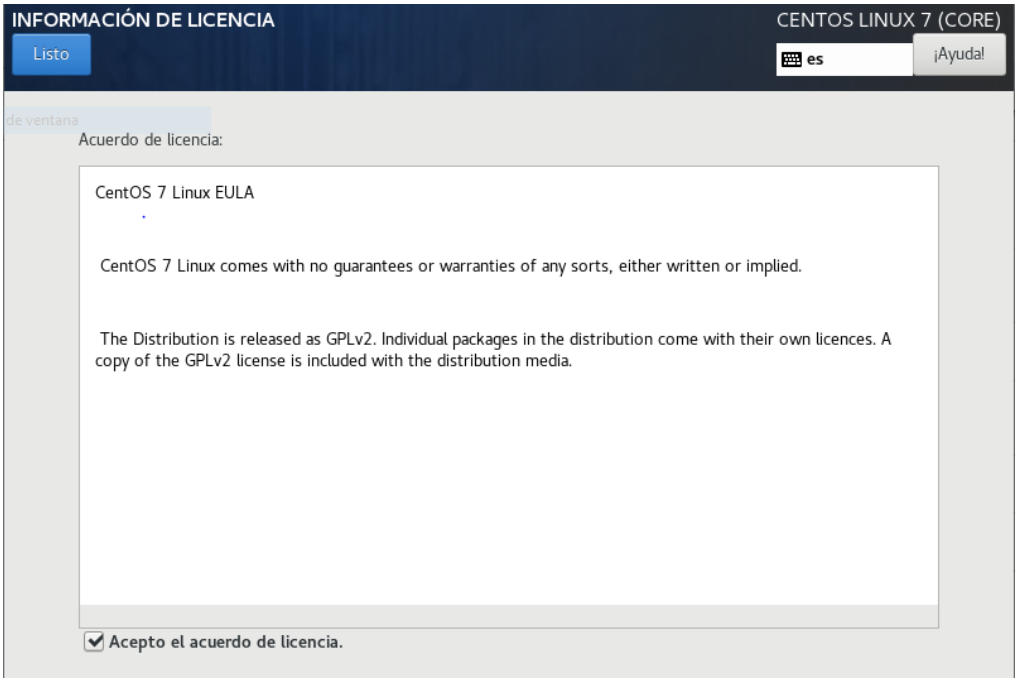


Imagen 18. *Información de licencia*

Para finalizar, se hará *clic* en el botón "*FINALIZAR CONFIGURACIÓN*" [Imagen 18]. Hecho esto se accederá con usuario y contraseña, se realizará paso por paso una

configuración inicial con nuestras preferencias personales y ya se dispondrá de una máquina virtual con un *SO* listo para usarse.

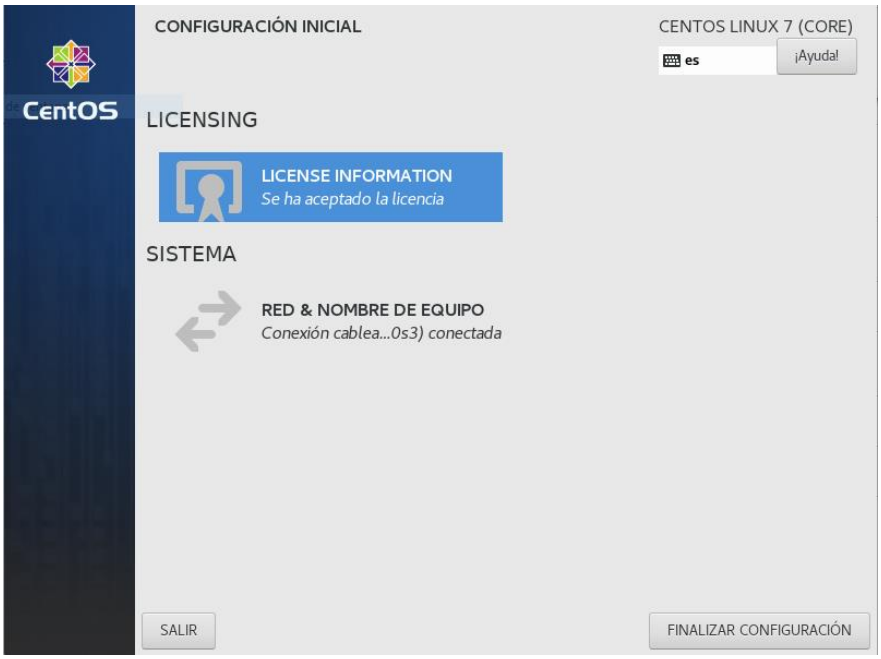


Imagen 19. *Configuración inicial*

3.3 Actualización de *CentOS 7* e instalación de *Guest Additions*.

3.3.1 Actualización del sistema CentOS 7

El primer paso que debe realizar en un sistema CentOS [9] recién instalado es asegurarse de que el sistema esté actualizado con los últimos parches de seguridad del *kernel* y del sistema, repositorios de software y paquetes.

Para actualizar completamente un sistema CentOS 7, se emitirán los siguientes comandos:

- `# yum check-update`
- `# yum upgrade`

3.3.2 Instalación de Guest Additions

VirtualBox Guest Additions es un paquete especial de software que consiste en una serie de controladores y aplicaciones para un mejor rendimiento y usabilidad de la máquina virtual [10].

Antes de instalar *Guest Additions*, primero se habilitará el repositorio *EPEL* con el siguiente comando:

- `# sudo rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm`

A continuación, se ejecutará el siguiente comando para instalar los prerrequisitos que son una serie de paquetes necesarios para la posterior instalación de las *Guest Additions*.

- `# sudo yum install -y bzip2 gcc make perl kernel-devel kernel-headers`

En la Ventana de *VirtualBox* de la máquina virtual se irá a "*Dispositivos*" y posteriormente se hará *clic* en "*Insertar imagen de CD de los complementos del invitado...*" [Imagen 20].

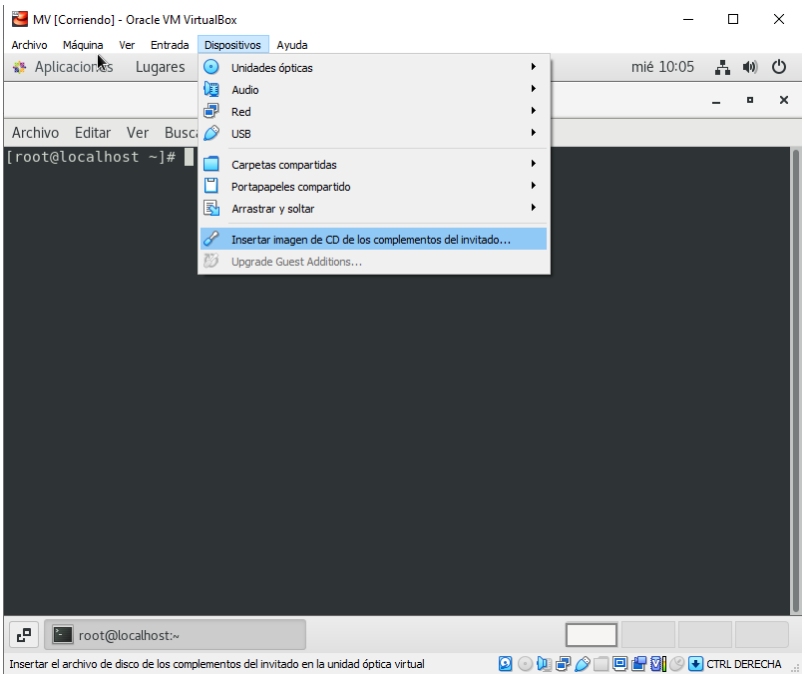


Imagen 20. *Insertar Imagen de CD Guest Additions*

Para terminar, se abrirá una pequeña ventana para ejecutar el software de *Guest Additions* y se hará *clic* en "*Ejecutar*" [Imagen 21].

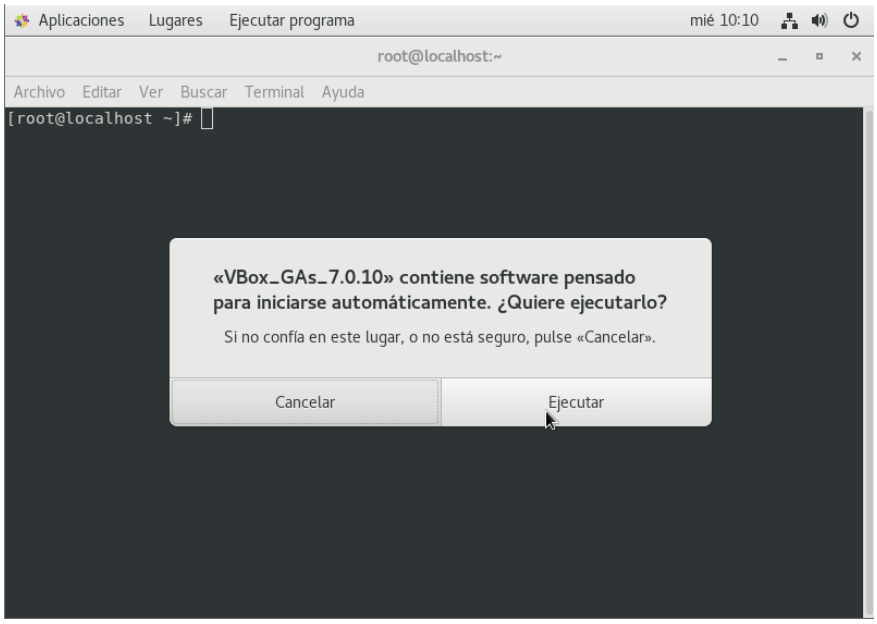


Imagen 21. *Ejecutar Guest Additions*

Una vez finalizada la ejecución, se reiniciará la máquina virtual y se dispondrá de una máquina virtual con una mejor usabilidad para facilitar nuestro trabajo.

3.4 Creación de una Red *NAT*

Para la posterior configuración de las máquinas virtuales y la conexión entre ellas se creará una red virtual para poder establecer direcciones IP a las diferentes configuraciones de red de las máquinas que conformarán el servidor de *Wazuh* y los agentes conectados a éstas.

Para ello se irá a "*Herramientas*" en el menú de inicio de *VirtualBox* y se hará *clic* en el icono que pone "*Crear*" [Imagen 22].

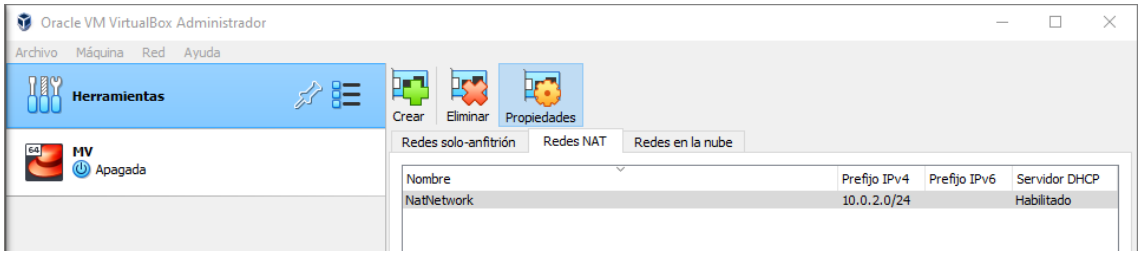


Imagen 22. *Crear una nueva Red NAT*

A continuación, en "*Opciones generales*" se editará el nombre de nuestra red y se le llamará "*Red Nat*". Se editará también el prefijo IPv4 y se establecerá *192.168.22.0/24* como la dirección IP de la nueva red y se marcará la opción "*Habilitar DHCP*". Una vez configurados estos parámetros se hará *clic* en "*Aplicar*" [Imagen 23].

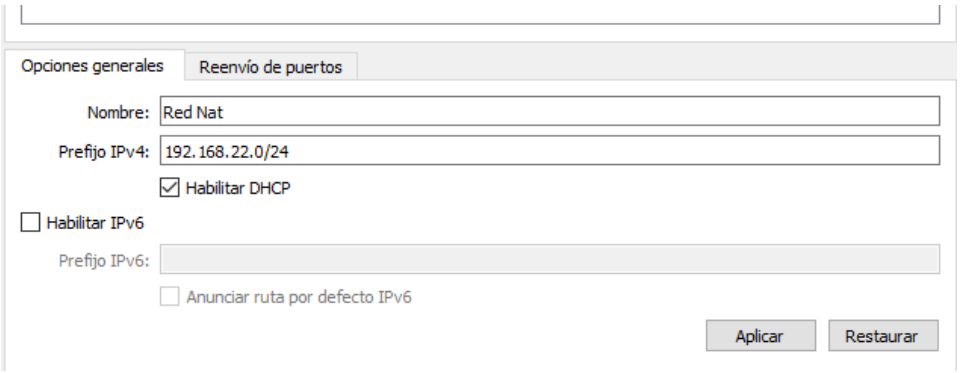


Imagen 23. *Opciones generales Red Nat*

Creada la red, ya se dispondrá de una red *NAT* para la posterior configuración de las máquinas virtuales a lo largo del trabajo.

3.5 Configuración de red de la máquina virtual

Una vez creada la red *Nat* y actualizada la máquina virtual creada anteriormente, se dispondrá a realizar la configuración de red asignando una dirección IP a la máquina de modo que se encuentre dentro de la red creada en el paso anterior.

Primero se añadirá la máquina virtual a la red *Nat* creada. Para ello, en el inicio de *VirtualBox* se hará *clic* en nuestra máquina creada y de nuevo se hará *clic* en "*Configuración*" [Imagen 24].

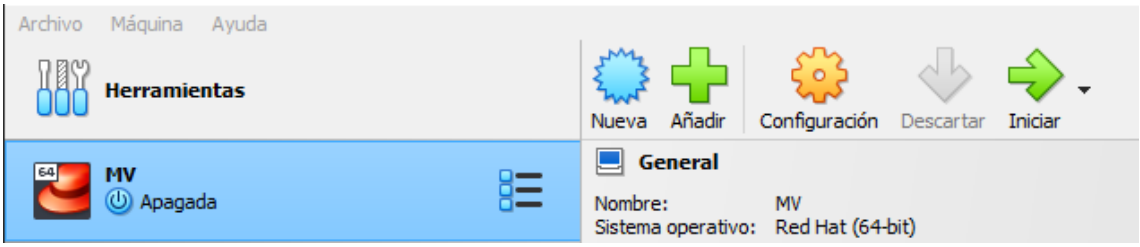


Imagen 24. *Configuración de la MV*

A continuación, se abrirá una ventana de configuración de la máquina virtual. En ella se irá al apartado de "*Red*" y en el "*Adaptador 1*" se editará la configuración cambiando el tipo de red seleccionando "*Red NAT*" y se elegirá la red "*Red Nat*" creada anteriormente [Imagen 25].

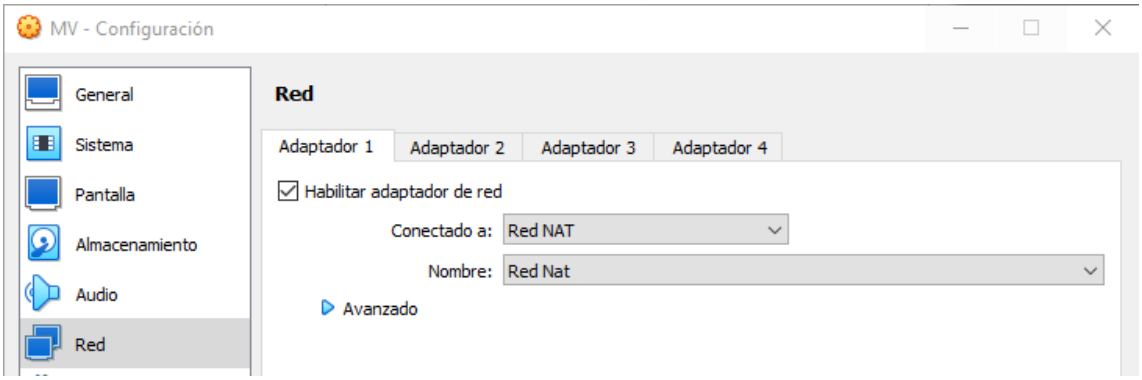


Imagen 25. *Configuración de red de la MV*

Una vez cambiados los parámetros, se hará *clic* en el botón "*Aceptar*" para guardar los cambios y se iniciará la máquina virtual. Una vez iniciada la máquina se escribirá en la consola el siguiente comando para ver la configuración del adaptador de red [Imagen 26].

- `# ifconfig`

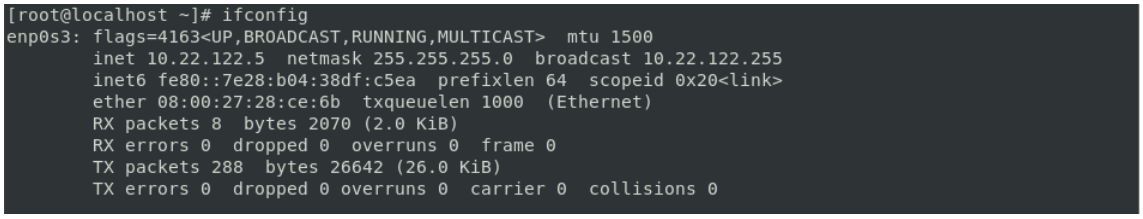


Imagen 26. *Resultado del comando ifconfig*

Como se puede observar, la dirección IP del adaptador de red es 10.22.122.5, por lo que en nuestro caso se cambiará a la 192.168.22.5. Para ello se editará el fichero */etc/sysconfig/network-scripts/ifcfg-enp0s3* añadiendo o modificando los siguientes campos:

- `BOOTPROTO="static"`
- `NETWORK="192.168.22.0"`
- `IPADDR="192.168.22.5"`
- `GATEWAY="192.168.22.1"`
- `NETMASK="255.255.255.0"`
- `DNS1="8.8.8.8"`

Configurado el fichero, se guardarán los cambios y se escribirá en la consola el siguiente comando para reiniciar el adaptador de red:

- `# systemctl restart network.service`

Para verificar que todo esté configurado correctamente y que se dispone de conexión hacia internet se ejecutará el siguiente comando:

- `# ping google.es`

4. Instalación y configuración del clúster de *Wazuh*

4.1 Introducción

En este apartado se pasará a configurar las diferentes máquinas virtuales que contendrán los componentes de *Wazuh* que se instalarán en ellas. A partir de la máquina virtual creada anteriormente, se clonará y se harán sus respectivas modificaciones de configuración para la creación de las máquinas virtuales que se usarán posteriormente. Estas máquinas virtuales conformarán un clúster que será controlado y configurado a través de la herramienta *Wazuh*.

Primero se pasará a la instalación y configuración de los diferentes componentes de *Wazuh*. Para la configuración en clúster se tendrá en cuenta que las máquinas virtuales donde se instalarán los componentes deberán estar en la misma red, creada anteriormente la red *NAT* con este último propósito.

4.2 Requisitos de configuración de las máquinas virtuales

Antes de empezar a instalar los componentes centrales de *Wazuh* en las distintas máquinas virtuales hay que tener en cuentas los requisitos y recomendaciones de hardware que ofrece el manual de *Wazuh* y que deberán tener las mismas. Para ello se han generado unas tablas con la configuración que hay que aplicar para cada una de las máquinas virtuales clonadas. En este trabajo se realizará configuración mínima para el funcionamiento de *Wazuh*.

4.2.1 Requisitos *Indexer*

En esta máquina virtual se instalarán *Wazuh Indexer* y el *Wazuh Dashboard*. A continuación, se presenta la tabla con las configuraciones en función de los requisitos [11] [12] que debe tener la máquina virtual [Tabla 1].

Tabla 1. *Requisitos máquina virtual Indexer*

Indexer	
Sistema Operativo	CentOS 7
RAM	8GB
CPUs	4
Almacenamiento	50GB
Dirección IP	192.168.22.10

4.2.2 Requisitos Nodo1

En esta máquina virtual se instalará *Wazuh Server* y se configurará como nodo maestro. A continuación, se presenta la tabla con las configuraciones en función de los requisitos [13] que debe tener la máquina virtual [Tabla 2].

Tabla 2. *Requisitos máquina virtual Nodo1*

Nodo1	
Sistema Operativo	CentOS 7
RAM	2GB
CPUs	2
Almacenamiento	50GB
Dirección IP	192.168.22.11

4.2.3 Requisitos Nodo2

En esta máquina virtual se instalará *Wazuh Server* y se configurará como nodo *worker*. A continuación, se presenta la tabla con las configuraciones en función de los requisitos [13] que debe tener la máquina virtual [Tabla 3].

Tabla 3. *Requisitos máquina virtual nodo2*

Nodo2	
Sistema Operativo	CentOS 7
RAM	2GB
CPUs	2
Almacenamiento	50GB
Dirección IP	192.168.22.12

4.2.4 Requisitos Nodo3

En esta máquina virtual se instalará *Wazuh Server* y se configurará como nodo *worker*. A continuación, se presenta la tabla con las configuraciones en función de los requisitos [13] que debe tener la máquina virtual [Tabla 4].

Tabla 4. *Requisitos máquina virtual Nodo3*

Nodo3	
Sistema Operativo	CentOS 7
RAM	2GB
CPUs	2
Almacenamiento	50GB
Dirección IP	192.168.22.13

4.2.5 Puertos requeridos

Se listarán en la siguiente tabla una serie de puertos [14] a tener en cuenta y que están implicados en la conexión de los diferentes componentes de *Wazuh* o en su correcto funcionamiento [Tabla 5].

Tabla 5. *Puertos requeridos*

Componente	Puerto	Protocolo	Función
Wazuh server	1514	TCP	Servicio de conexión con agente
	1514	UDP	Servicio de conexión con agente
	1515	TCP	Servicio de registro del agente
	1516	TCP	Demonio del *Wazuh* clúster
	514	UDP	Recopilador de *Wazuh* Syslog
	514	TCP	Recopilador de *Wazuh* Syslog
	55000	TCP	API RESTful de *Wazuh server*
Wazuh indexer	9200	TCP	API RESTful de *Wazuh indexer*
Wazuh dashboard	443	TCP	Interfaz de usuario web de *Wazuh*

4.3 Clonación de MV

Una vez conocidas los requisitos de cada máquina virtual, se empezarán a crear las distintas máquinas a partir de la clonación de la máquina virtual creada anteriormente en el apartado *Creación de una MV.*

Para clonar una máquina virtual, se hará *clic* derecho sobre la máquina virtual que se quiera clonar y se seleccionará la opción *"Clonar..."* [Imagen 27]

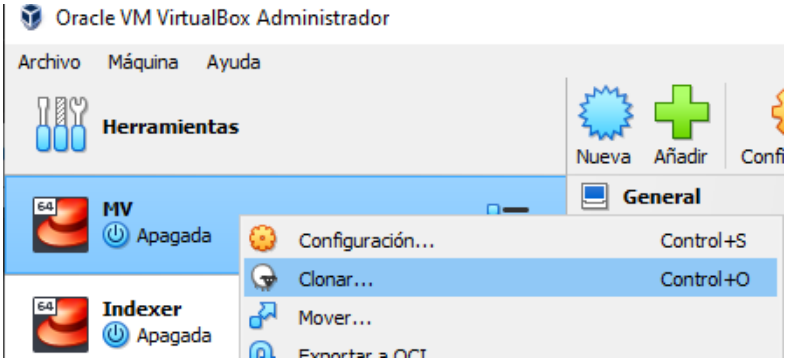


Imagen 27. *Clonar máquina virtual*

A continuación, se abrirá una ventana llamada "*Clonar máquina virtual"* donde se escribe el nombre de la nueva máquina que se creará a partir de la clonada, la ruta donde se guardará en nuestro equipo y en la opción "*Política de dirección MAC"* se seleccionará la opción "*Generar nuevas direcciones MAC para todos los adaptadores de red"*. Una vez editadas las opciones se hará *clic* en "*Siguiente"* [Imagen 28].

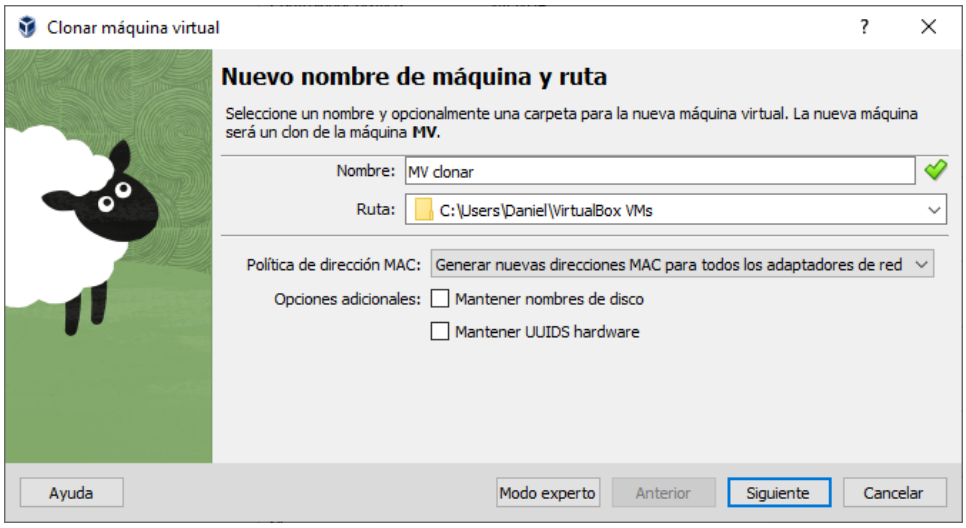


Imagen 28. *Nuevo nombre de máquina y ruta de la clonación*

Para finalizar, se seleccionará el tipo de clonación que en este caso será "*Clonación Completa"*, seleccionada esta opción se hará *clic* en "*Terminar"* [Imagen 29] y se ejecutará la clonación creándose una nueva máquina virtual que es una copia de la máquina virtual original.

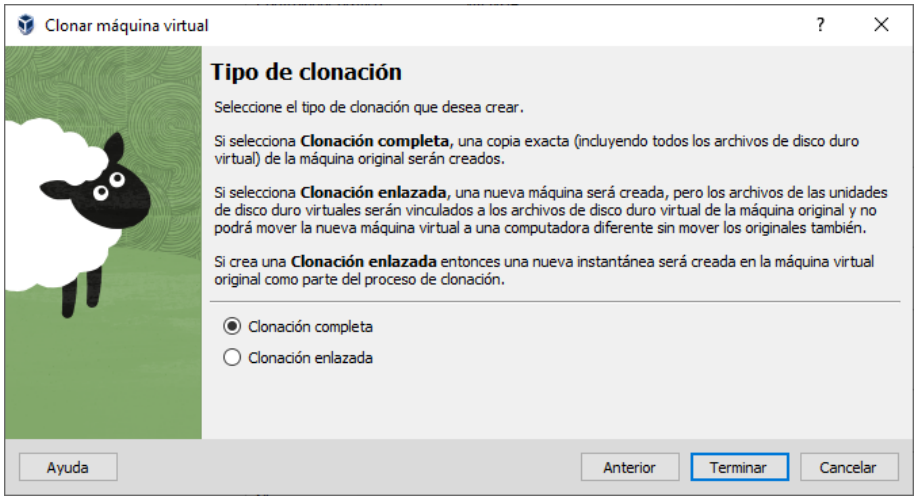


Imagen 29. *Tipo de clonación*

Este proceso de clonación será igual para las demás máquinas virtuales a utilizar a lo largo del proyecto.

4.4 Configuración *Indexer*

Una vez se disponga de la máquina virtual *Indexer* se realizarán una serie de configuraciones.

4.4.1 Hardware

Primero se pasará a modificar el hardware fijándose en los requisitos que hay de hardware para esta máquina virtual reflejados en puntos anteriores. Para ello se

hará *clic* derecho en la máquina virtual *Indexer* y se procederá a hacer clic en el apartado "*Configuración...*" [Imagen 30].

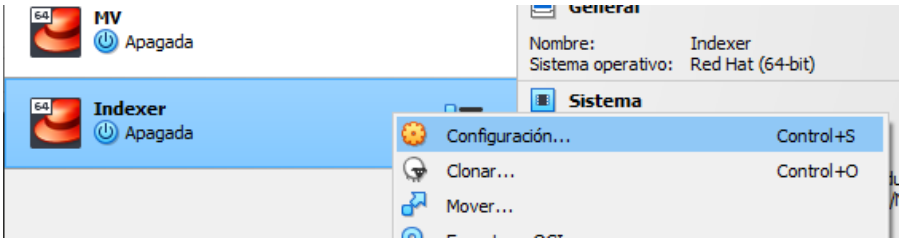


Imagen 30. *Configuración de Indexer*

A continuación, se abrirá una ventana de configuración de la máquina virtual, donde se procederá a hacer *clic* en "*Sistema*" donde se mostrará una serie de configuraciones de hardware del sistema de la máquina virtual [Imagen 31]. Dentro de "*Sistema*" en el apartado de "*Placa base*" en "*Memoria base*" se seleccionarán 8192 MB de memoria RAM para la máquina virtual.

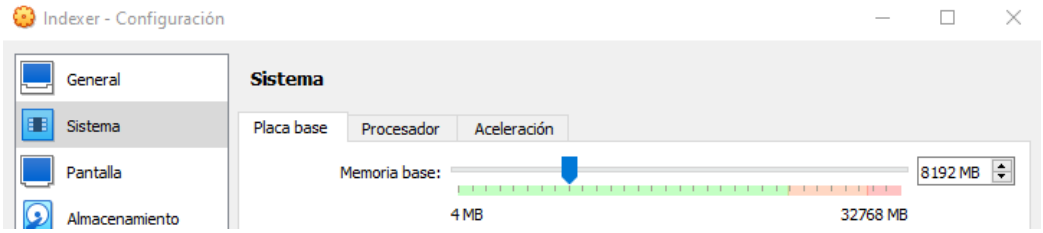


Imagen 31. *Configuración Indexer memoria base*

Para terminar, en "Sistema" en el apartado "Procesador" en "Procesadores" se seleccionarán 4 *CPUs* [Imagen 32], que se corresponden con el número de núcleos del procesador. Una vez modificadas las configuraciones se hará *clic* en el botón de "Aceptar" para guardar la configuración.

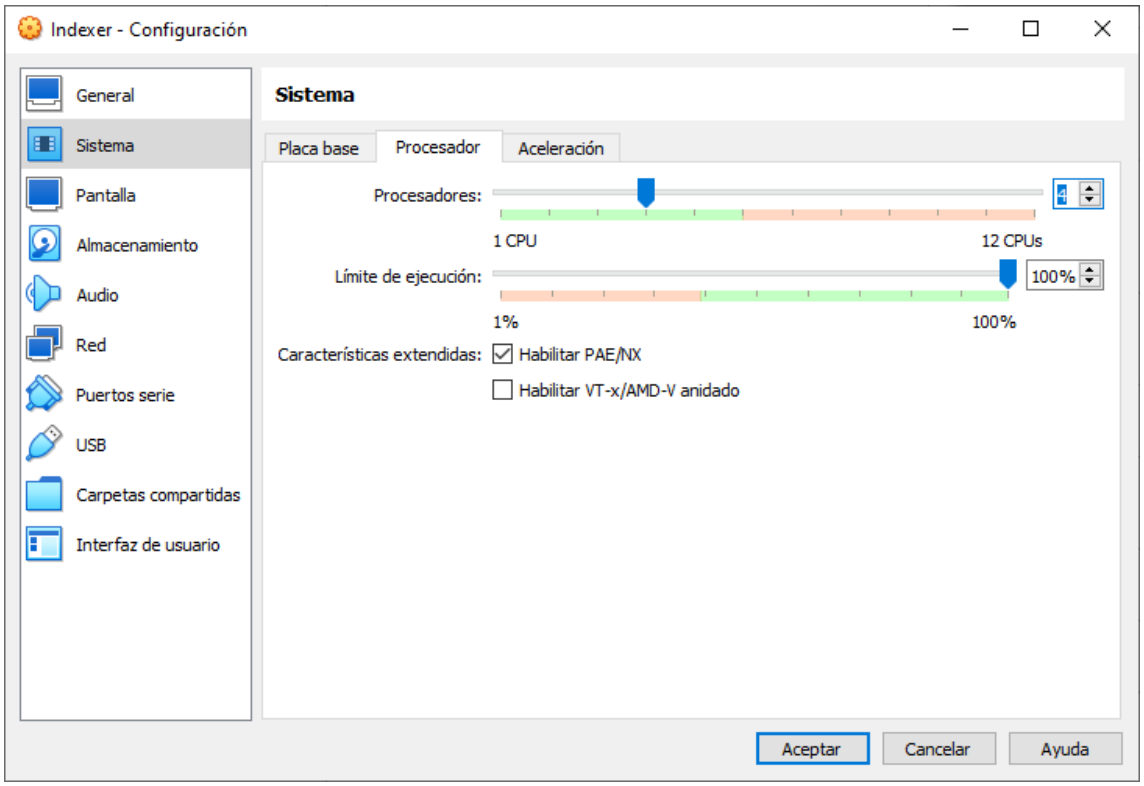


Imagen 32. *Configuración Indexer procesador*

En el menú de inicio de *VirtualBox* si se hace *clic* en la máquina virtual *Indexer* se podrán observar justo a la derecha las características de la máquina virtual y comprobar que las configuraciones se han modificado correctamente [Imagen 33].

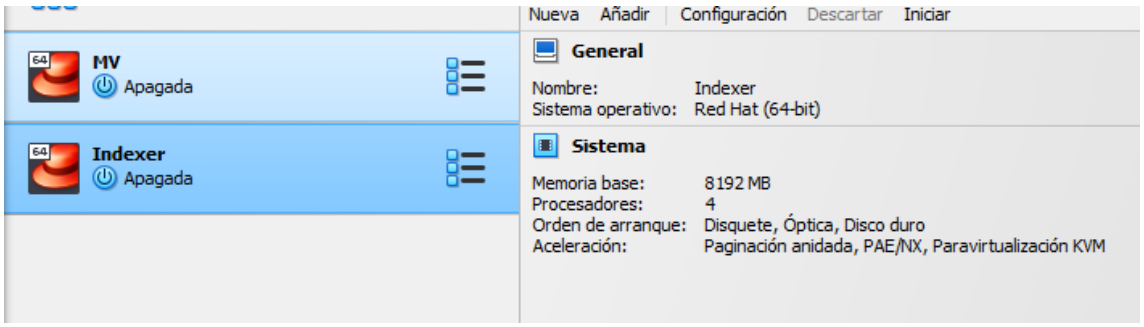


Imagen 33. *Características hardware Indexer*

4.4.2 Red

Una vez concluida la configuración de hardware se pasará a la configuración de red de la máquina virtual *Indexer*. Para ello se modificará la IP dentro de la red *NAT* creada anteriormente y el UUID de la tarjeta de red de la máquina virtual. Una vez realizados los cambios se comprobará que la máquina sigue teniendo conexión a internet con el exterior.

Se pasará a iniciar la máquina virtual *Indexer* y se abrirá la consola de comandos y se introducirá el siguiente comando:

- `# ifconfig`

Se mostrarán las diferentes puertas de enlace de red de la máquina virtual [Imagen 34] y como se podrá observar la dirección IP actual del adaptador de red "enp0s3" es 192.168.22.5 que se corresponde con la dirección IP de la máquina virtual MV que es la que se clonó anteriormente para la creación de la máquina virtual *Indexer*.

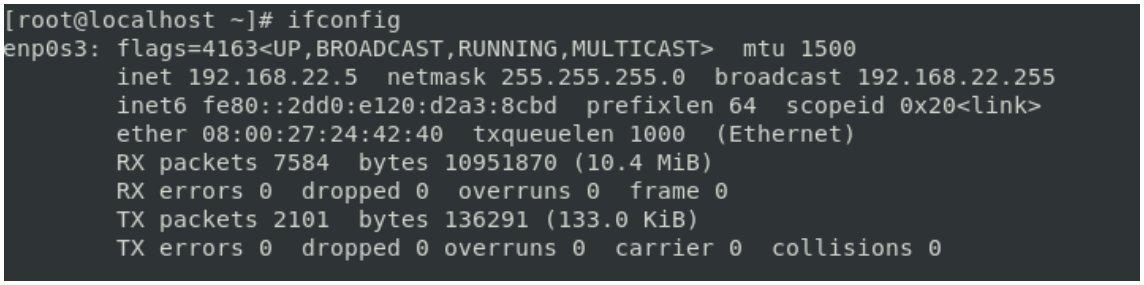


Imagen 34. *Resultado comando ifconfig Indexer*

La dirección IP del adaptador de red se cambiará a la *192.168.22.10*. Para ello se editará el fichero de configuración del adaptador de red */etc/sysconfig/network-scripts/ifcfg-enp0s3* modificando los siguientes campos:

- `UUID="456fe53e-7e7a-446e-ac57-7d5d368a7181"`
- `IPADDR="192.168.22.10"`

El UUID nuevo se ha generado escribiendo el siguiente comando en consola:

- `# uuidgen`

Una vez modificados los campos y guardados los cambios se reiniciará el adaptador de red para que el sistema actualice los cambios mediante el comando:

- `# systemctl restart network`

Después se comprobará que se tiene conexión a internet usando el comando ping [Imagen 35]:

- `# ping google.es`

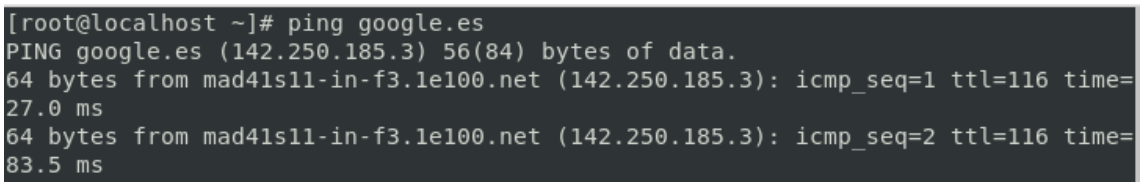


Imagen 35. *Resultado comando ping*

Se podrá observar que existe conexión a internet por lo que los cambios en la configuración de red se han realizado correctamente.

Finalmente, se editará el *hostname* de la máquina virtual.

- `# hostnamectl set-hostname indexer`

Se reiniciará la máquina virtual.

- `# reboot`

4.5 Configuración Nodo1

Una vez se disponga de la máquina virtual *Nodo1* se realizarán una serie de configuraciones.

4.5.1 Hardware

Primero se pasará a modificar el hardware fijándose en los requisitos que hay de hardware para esta máquina virtual reflejados en puntos anteriores. Para ello se hará *clic* derecho en la máquina virtual *Nodo1* y se procederá a hacer *clic* en el apartado "*Configuración...*" [Imagen 36].

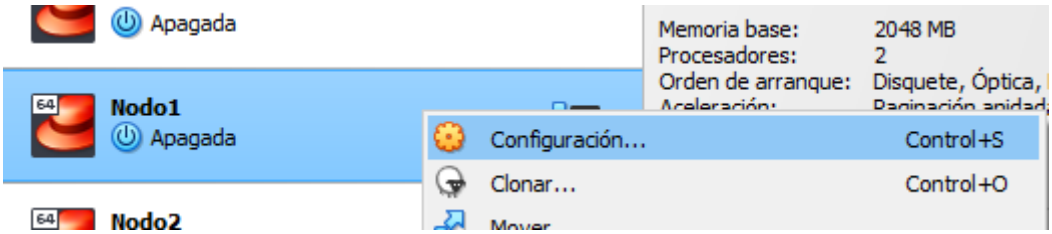


Imagen 36. *Configuración de Nodo1*

A continuación, se abrirá una ventana de configuración de la máquina virtual, donde se procederá a hacer *clic* en "*Sistema*" donde se mostrará una serie de configuraciones de hardware del sistema de la máquina virtual [Imagen 37]. Dentro de "*Sistema*" en el apartado de "*Placa base*" en "*Memoria base*" se seleccionarán 2048 MB de memoria RAM para la máquina virtual.

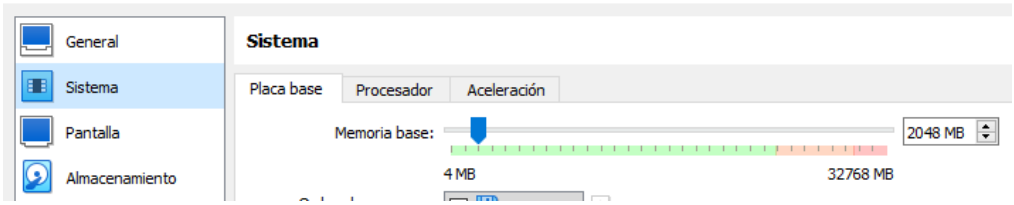


Imagen 37. *Configuración Nodo1 memoria base*

Para terminar, en "Sistema" en el apartado "Procesador" en "Procesadores" se seleccionarán 2 CPUs [Imagen 38], que se corresponden con el número de núcleos del procesador. Una vez modificadas las configuraciones se hará *clic* en el botón de "Aceptar" para guardar la configuración.

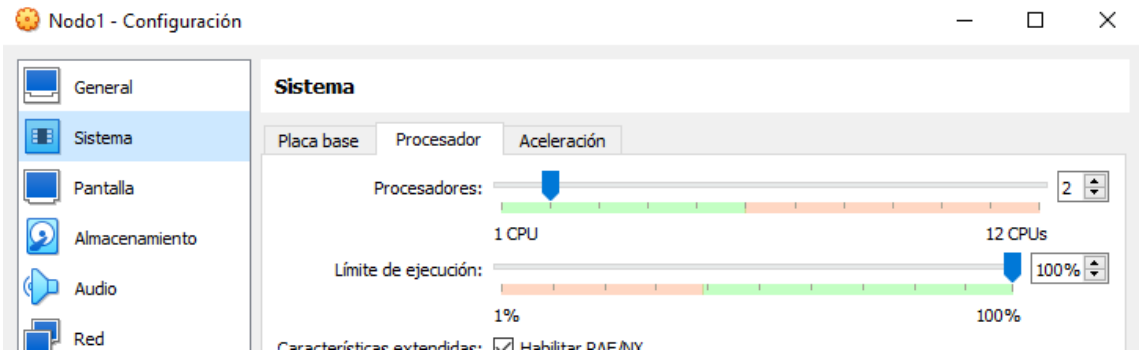


Imagen 38. *Configuración Nodo1 procesador*

En el menú de inicio de *VirtualBox* si se hace *clic* en la máquina virtual *Nodo1* se podrán observar justo a la derecha las características de la máquina virtual y comprobar que las configuraciones se han modificado correctamente [Imagen 39].

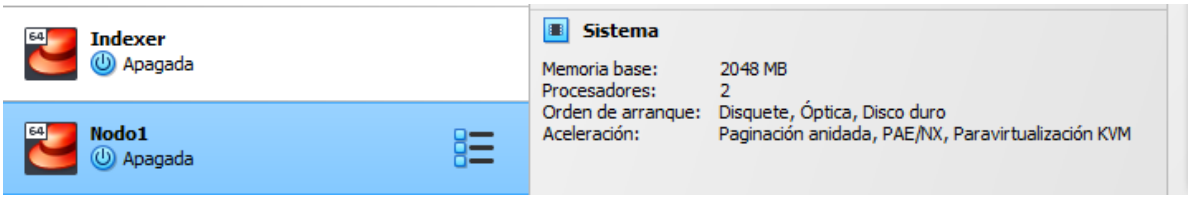


Imagen 39. *Características hardware Nodo1*

4.5.2 Red

Una vez concluida la configuración de hardware se pasará a la configuración de red de la máquina virtual *Nodo1*. Para ello se modificará la IP dentro de la red *NAT* creada anteriormente y el UUID de la tarjeta de red de la máquina virtual. Una vez realizados los cambios se comprobará que la máquina sigue teniendo conexión a internet con el exterior.

Se pasará a iniciar la máquina virtual *Nodo1* y se abrirá la consola de comandos y se introducirá el siguiente comando:

- `# ifconfig`

Se mostrarán las diferentes puertas de enlace de red de la máquina virtual [Imagen 40] y como se podrá observar la dirección IP actual del adaptador de red "enp0s3" es *192.168.22.5* que se corresponde con la dirección IP de la máquina virtual *MV* que es la que se clonó anteriormente para la creación de la máquina virtual *Nodo1*.

```
[root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.22.5  netmask 255.255.255.0  broadcast 192.168.22.255
        inet6 fe80::2dd0:e120:d2a3:8cbd  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:9f:14:6f  txqueuelen 1000  (Ethernet)
        RX packets 430  bytes 250096 (244.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 438  bytes 33905 (33.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Imagen 40. *Resultado comando ifconfig Nodo1*

La dirección IP del adaptador de red se cambiará a la *192.168.22.11*. Para ello se editará el fichero de configuración del adaptador de red */etc/sysconfig/network-scripts/ifcfg-enp0s3* modificando los siguientes campos:

- `UUID="7f660b8d-442c-4cc4-b26c-80c71609cb05"`
- `IPADDR="192.168.22.11"`

El UUID nuevo se ha generado escribiendo el siguiente comando en consola:

- `# uuidgen`

Una vez modificados los campos y guardados los cambios se reiniciará el adaptador de red para que el sistema actualice los cambios mediante el comando:

- `# systemctl restart netwok`

Después se comprobará que se tiene conexión a internet usando el comando ping:

- `# ping google.es`

Se podrá observar que existe conexión a internet por lo que los cambios en la configuración de red se han realizado correctamente. Para concluir, ya que se disponen de 2 máquinas virtuales configuradas en la misma red como son *Indexer* y *Nodo1*, se ejecutará el comando ping con la dirección IP de las máquinas para comprobar que se conectan entre ellas. Por ejemplo, desde la máquina virtual *Nodo1* ejecutar el comando ping de la siguiente manera:

- `# ping 192.168.22.10`

Una vez comprobado esto, desde la máquina virtual *Indexer* también se ejecutará el comando ping a la dirección IP de la máquina virtual *Nodo1*.

Finalmente, se editará el hostname de la máquina virtual.

- `# hostnamectl set-hostname nodo1`

Se reiniciará la máquina virtual.

- `# reboot`

4.6 Configuración Nodo2

Una vez se disponga de la máquina virtual *Nodo2* se realizarán una serie de configuraciones.

4.6.1 Hardware

Primero se pasará a modificar el hardware fijándose en los requisitos que hay de hardware para esta máquina virtual reflejados en puntos anteriores. Para ello se hará *clic* derecho en la máquina virtual *Nodo2* y se procederá a hacer *clic* en el apartado "*Configuración...*" [Imagen 41].

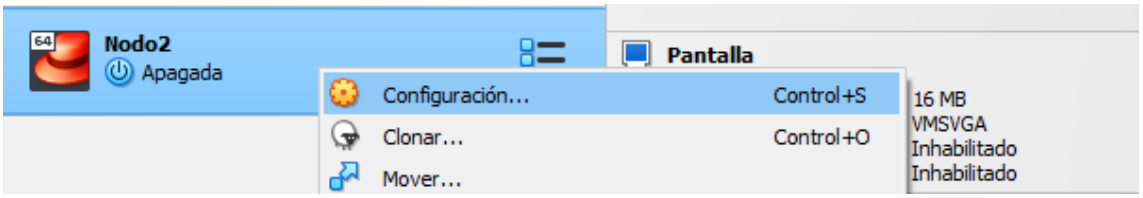


Imagen 41. *Configuración de Nodo2*

A continuación, se abrirá una ventana de configuración de la máquina virtual, donde se procederá a hacer *clic* en "*Sistema*" donde se mostrará una serie de configuraciones de hardware del sistema de la máquina virtual [Imagen 42]. Dentro

de “*Sistema*” en el apartado de “*Placa base*” en “*Memoria base*” se seleccionarán 2048 MB de memoria RAM para la máquina virtual.

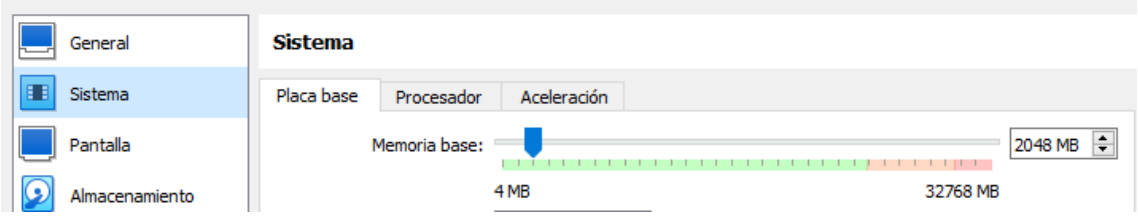


Imagen 42. *Configuración Nodo1 memoria base*

Para terminar, en “Sistema” en el apartado “Procesador” en “Procesadores” se seleccionarán 2 *CPUs* [Imagen 43], que se corresponden con el número de núcleos del procesador. Una vez modificadas las configuraciones se hará *clic* en el botón de “Aceptar” para guardar la configuración

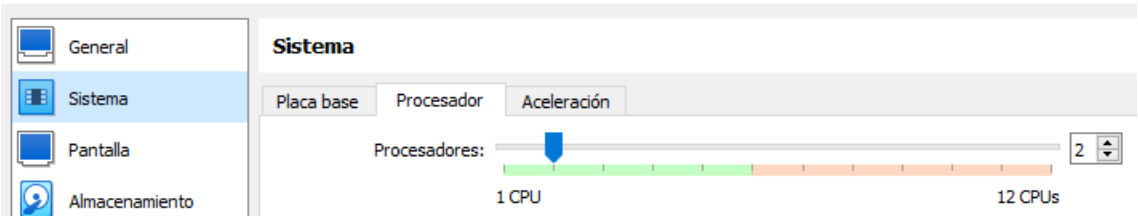


Imagen 43. *Configuración Nodo2 procesador*

En el menú de inicio de *VirtualBox* si se hace *clic* en la máquina virtual *Nodo2* se podrán observar justo a la derecha las características de la máquina virtual y comprobar que las configuraciones se han modificado correctamente [Imagen 44].

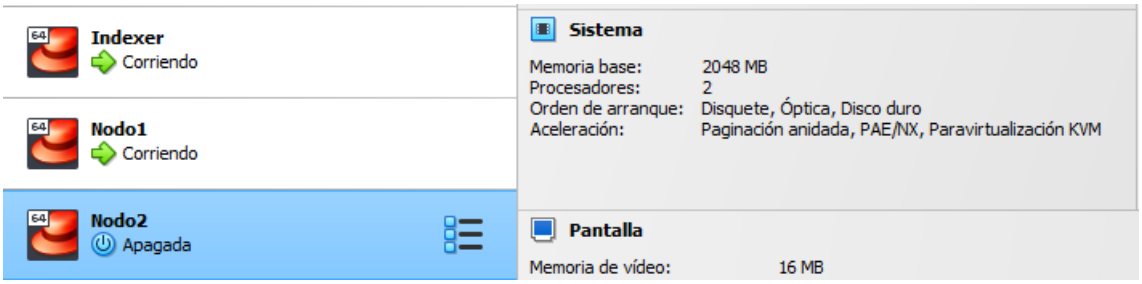


Imagen 44. *Características hardware Nodo2*

4.6.2 Red

Una vez concluida la configuración de hardware se pasará a la configuración de red de la máquina virtual *Nodo2*. Para ello se modificará la IP dentro de la red *NAT* creada anteriormente y el UUID de la tarjeta de red de la máquina virtual. Una vez realizados los cambios se comprobará que la máquina sigue teniendo conexión a internet con el exterior.

Se pasará a iniciar la máquina virtual *Nodo2* y se abrirá la consola de comandos y se introducirá el siguiente comando:

- `# ifconfig`

Se mostrarán las diferentes puertas de enlace de red de la máquina virtual [Imagen 45] y como se podrá observar la dirección IP actual del adaptador de red “enp0s3” es *192.168.22.5* que se corresponde con la dirección IP de la máquina virtual *MV* que es la que se clonó anteriormente para la creación de la máquina virtual *Nodo2*.

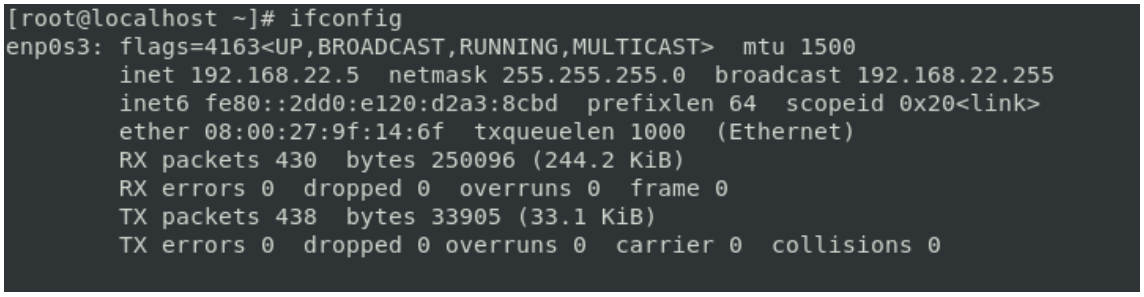


Imagen 45. *Resultado comando ifconfig Nodo2*

La dirección IP del adaptador de red se cambiará a la *192.168.22.12*. Para ello se editará el fichero de configuración del adaptador de red */etc/sysconfig/network-scripts/ifcfg-enp0s3* modificando los siguientes campos:

- `UUID=“8098fd23-a97c-43e8-afb7-f0d2a2e76d36”`
- `IPADDR=”192.168.22.12”`

El UUID nuevo se ha generado escribiendo el siguiente comando en consola:

- `# uuidgen`

Una vez modificados los campos y guardados los cambios se reiniciará el adaptador de red para que el sistema actualice los cambios mediante el comando:

- `# systemctl restart netwok`

Después se comprobará que se tiene conexión a internet usando el comando ping:

- `# ping google.es`

Se podrá observar que existe conexión a internet por lo que los cambios en la configuración de red se han realizado correctamente. Para concluir, ya que se disponen de 3 máquina virtuales configuradas en la misma red como son *Indexer*, *Nodo1* y *Nodo2*, se realizarán comprobaciones de conexión entre ellas mediante el comando ping para comprobar que dichas máquinas están conectadas entre ellas correctamente en la misma red, tal y como se realizó al final del apartado de configuración de red del *Nodo1*.

Finalmente, se editará el hostname de la máquina virtual.

- `# hostnamectl set-hostname nodo2`

Se reiniciará la máquina virtual.

- `# reboot`

4.7 Configuración Nodo3

Una vez se disponga de la máquina virtual *Nodo3* se realizarán una serie de configuraciones.

4.7.1 Hardware

Primero se pasará a modificar el hardware fijándose en los requisitos que hay de hardware para esta máquina virtual reflejados en puntos anteriores. Para ello se hará *clic* derecho en la máquina virtual *Nodo3 y* se procederá a hacer *clic* en el apartado "*Configuración…*" [Imagen 46].

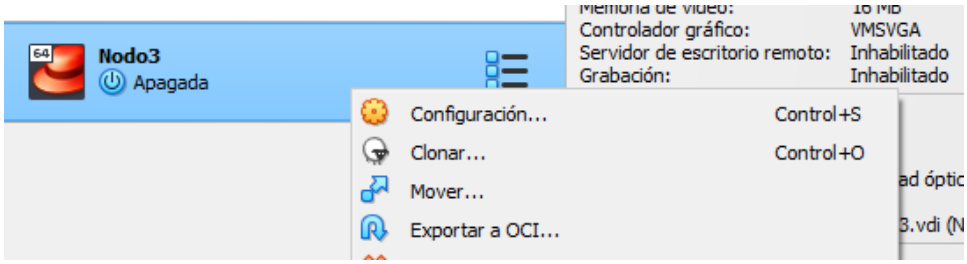


Imagen 46. *Configuración de Nodo3*

A continuación, se abrirá una ventana de configuración de la máquina virtual, donde se procederá a hacer *clic* en "*Sistema*" donde se mostrará una serie de configuraciones de hardware del sistema de la máquina virtual [Imagen 47]. Dentro de "*Sistema*" en el apartado de "*Placa base*" en "*Memoria base*" se seleccionarán 2048 MB de memoria RAM para la máquina virtual.

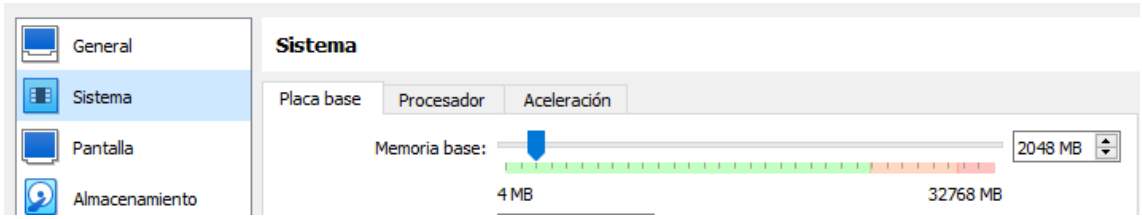


Imagen 47. *Configuración Nodo3 memoria base*

Para terminar, en "Sistema" en el apartado "Procesador" en "Procesadores" se seleccionarán 2 CPUs [Imagen 48], que se corresponden con el número de núcleos del procesador. Una vez modificadas las configuraciones se hará *clic* en el botón de "Aceptar" para guardar la configuración

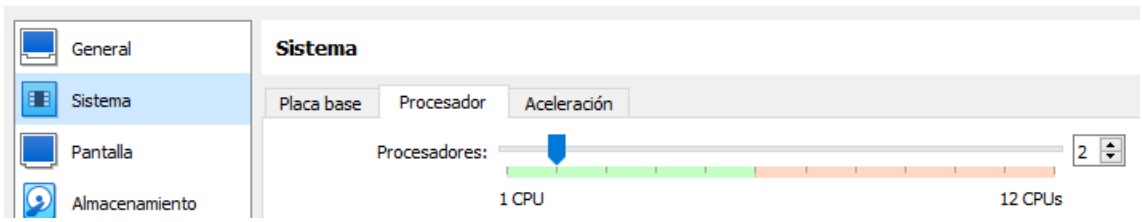


Imagen 48. *Configuración Nodo3 procesador*

En el menú de inicio de *VirtualBox* si se hace *clic* en la máquina virtual *Nodo3* se podrán observar justo a la derecha las características de la máquina virtual y comprobar que las configuraciones se han modificado correctamente [Imagen 49].

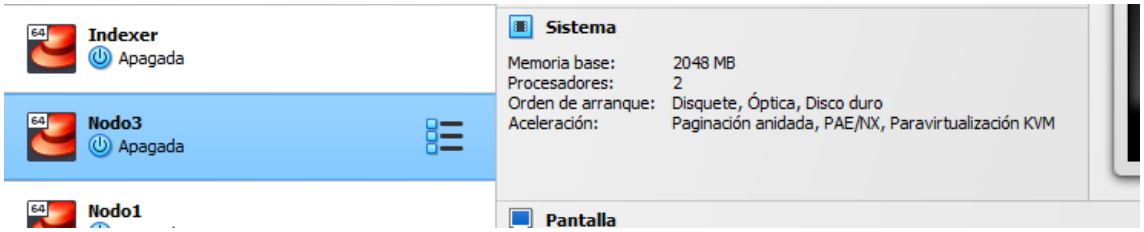


Imagen 49. *Características hardware Nodo3*

4.7.2 Red

Una vez concluida la configuración de hardware se pasará a la configuración de red de la máquina virtual *Nodo3*. Para ello se modificará la IP dentro de la red *NAT* creada anteriormente y el UUID de la tarjeta de red de la máquina virtual. Una vez realizados los cambios se comprobará que la máquina sigue teniendo conexión a internet con el exterior.

Se pasará a iniciar la máquina virtual *Nodo3* y se abrirá la consola de comandos y se introducirá el siguiente comando:

- `# ifconfig`

Se mostrarán las diferentes puertas de enlace de red de la máquina virtual [Imagen 50] y como se podrá observar la dirección IP actual del adaptador de red "enp0s3" es *192.168.22.5* que se corresponde con la dirección IP de la máquina virtual *MV* que es la que se clonó anteriormente para la creación de la máquina virtual *Nodo3*.

```
[root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.22.5  netmask 255.255.255.0  broadcast 192.168.22.255
        inet6 fe80::2dd0:e120:d2a3:8cbd  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:9f:14:6f  txqueuelen 1000  (Ethernet)
        RX packets 430  bytes 250096 (244.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 438  bytes 33905 (33.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Imagen 50. *Resultado comando ifconfig Nodo2*

La dirección IP del adaptador de red se cambiará a la *192.168.22.13*. Para ello se editará el fichero de configuración del adaptador de red */etc/sysconfig/network-scripts/ifcfg-enp0s3* modificando los siguientes campos:

- `UUID="5035de18-adc7-456a-ad77-bba22dce8086"`
- `IPADDR="192.168.22.13"`

El UUID nuevo se ha generado escribiendo el siguiente comando en consola:

- `# uuidgen`

Una vez modificados los campos y guardados los cambios se reiniciará el adaptador de red para que el sistema actualice los cambios mediante el comando:

- `# systemctl restart netwok`

Después se comprobará que se tiene conexión a internet usando el comando ping:

- `# ping google.es`

Se podrá observar que existe conexión a internet por lo que los cambios en la configuración de red se han realizado correctamente. Para concluir, ya que se disponen de 4 máquina virtuales configuradas en la misma red como son *Indexer*, *Nodo1*, Nodo2 y Nodo3 se realizarán comprobaciones de conexión entre ellas mediante el comando ping para comprobar que dichas máquinas están conectadas entre ellas correctamente en la misma red, tal y como se realizó al final del apartado de configuración de red del *Nodo1*.

Finalmente, se editará el hostname de la máquina virtual.

- `# hostnamectl set-hostname nodo3`

Se reiniciará la máquina virtual.

- `# reboot`

Se dispondrá de 4 máquinas correctamente configuradas para la posterior instalación de la herramienta *Wazuh* en cada una de ellas.

4.8 Instalación y configuración de *Wazuh indexer*

Para la generación de este punto se ha utilizado la guía de instalación de *Wazuh indexer* que se encuentra en la documentación de *Wazuh* [15]. La instalación y configuración del *Wazuh Indexer* se hará en la máquina virtual *Indexer*.

4.8.1 Creación de certificados

Se descargará el script *Wazuh*-certs-tool.sh y el archivo de configuración *config.yml* para generar los certificados SSL. Se crearán los certificados que cifran las comunicaciones entre los componentes centrales de *Wazuh*.

- `# curl -sO https://packages.Wazuh.com/4.5/Wazuh-certs-tool.sh`
- `# curl -sO https://packages.Wazuh.com/4.5/config.yml`

Se editará el fichero *./config.yml* y se reemplazarán los nombres de los nodos y los valores IP con los nombres y direcciones IP correspondientes. Se deberá hacer esto para todos los nodos del servidor *Wazuh*, el *indexer* y el *dashboard*. Se agregarán tantos campos de nodo como sean necesarios, que en este caso en el *Wazuh server* será de 3 nodos [Imagen 51].

- `# gedit ./config.yml &`

```
nodes:
  # Wazuh indexer nodes
  indexer:
    - name: indexer
      ip: "192.168.22.10"
    #- name: node-2
    #  ip: "<indexer-node-ip>"
    #- name: node-3
    #  ip: "<indexer-node-ip>"

  # Wazuh server nodes
  # If there is more than one Wazuh server
  # node, each one must have a node_type
  server:
    - name: nodo1
      ip: "192.168.22.11"
      node_type: master
    - name: nodo2
      ip: "192.168.22.12"
      node_type: worker
    - name: nodo3
      ip: "192.168.22.13"
      node_type: worker

  # Wazuh dashboard nodes
  dashboard:
    - name: dashboard
      ip: "192.168.22.10"
```

Imagen 51. *Fichero ./config.yml editado*

A continuación, se ejecutará el script *./Wazuh*-certs-tool.sh para crear los certificados [Imagen 52]. Estos certificados se deberán de implementar más adelante en todos los nodos del clúster de *Wazuh*.

- `# bash ./Wazuh-certs-tool.sh -A`

```
[root@indexer ~]# bash ./wazuh-certs-tool.sh -A
21/05/2024 15:41:49 INFO: Admin certificates created.
21/05/2024 15:41:49 INFO: Wazuh indexer certificates created.
21/05/2024 15:41:49 INFO: Wazuh server certificates created.
21/05/2024 15:41:49 INFO: Wazuh dashboard certificates created.
```

Imagen 52. *Creación de certificados*

Se comprimirán todos los ficheros necesarios.

- `# tar -cvf ./Wazuh-certificates.tar -C ./Wazuh-certificates/ .`
- `# rm -rf ./Wazuh-certificates`

Para terminar, se copiará el fichero *Wazuh*-certificates.tar a los nodos *Nodo1*, *Nodo2* y *Nodo3* que conformarán el *Wazuh server*. Para realizar esta acción se usará el comando *scp* [Imagen 53].

- `# scp -p root@192.168.22.10:./Wazuh-certificates.tar root@192.168.22.11:./`
- `# scp -p root@192.168.22.10:./Wazuh-certificates.tar root@192.168.22.12:./`
- `# scp -p root@192.168.22.10:./Wazuh-certificates.tar root@192.168.22.13:./`

De esta manera se copiará el fichero en el directorio raíz de las máquinas virtuales *Nodo1*, *Nodo2* y *Nodo3*.

```
[root@indexer ~]# scp -p root@192.168.22.10:./wazuh-certificates.tar root@192.168.22.11:./
The authenticity of host '192.168.22.10 (192.168.22.10)' can't be established.
ECDSA key fingerprint is SHA256:ft4QbAQXHL0LRw3QjnO68aXZ7m2+/X1O0+qH7mUukGo.
ECDSA key fingerprint is MD5:27:0b:d0:4f:ea:ed:be:b3:d7:27:b7:dc:4a:f4:a3:9b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.22.10' (ECDSA) to the list of known hosts.
root@192.168.22.10's password:
The authenticity of host '192.168.22.11 (192.168.22.11)' can't be established.
ECDSA key fingerprint is SHA256:ft4QbAQXHL0LRw3QjnO68aXZ7m2+/X1O0+qH7mUukGo.
ECDSA key fingerprint is MD5:27:0b:d0:4f:ea:ed:be:b3:d7:27:b7:dc:4a:f4:a3:9b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.22.11' (ECDSA) to the list of known hosts.
root@192.168.22.11's password:
wazuh-certificates.tar                                        100%   40KB  50.5MB/s   00:00
Connection to 192.168.22.10 closed.
```

Imagen 53. *Comando scp*

4.8.2 Instalación del nodo *indexer*

Antes de empezar se instalarán los siguientes paquetes.

- `# yum install coreutils`

A continuación, se añadirá el repositorio de *Wazuh*. Primero se importará la clave GPG.

- `# rpm --import https://packages.Wazuh.com/key/GPG-KEY-WAZUH`

Importada ya la clave se añadirá el repositorio [Imagen 54].

- `# echo -e '[Wazuh]\ngpgcheck=1\ngpgkey=https://packages.Wazuh.com/key/GPG-KEY-WAZUH\nenabled=1\nname=EL-$releasever - Wazuh\nbaseurl=https://packages.Wazuh.com/4.x/yum/\nprotect=1' | tee /etc/yum.repos.d/Wazuh.repo`

```
[root@indexer ~]# rpm --import https://packages.wazuh.com/key/GPG-KEY-WAZUH
[root@indexer ~]# echo -e '[wazuh]\ngpgcheck=1\ngpgkey=https://packages.wazuh.com/key/GPG-KEY-WAZUH\nena
bled=1\nname=EL-$releasever - Wazuh\nbaseurl=https://packages.wazuh.com/4.x/yum/\nprotect=1' | tee /etc/
yum.repos.d/wazuh.repo
[wazuh]
gpgcheck=1
gpgkey=https://packages.wazuh.com/key/GPG-KEY-WAZUH
enabled=1
name=EL-$releasever - Wazuh
baseurl=https://packages.wazuh.com/4.x/yum/
protect=1
```

Imagen 54. *Repositorio Wazuh*

Se pasará a instalar el paquete del *Wazuh indexer*

- `# yum -y install Wazuh-indexer`

Una vez instalado el paquete se pasará a configurar el *Wazuh indexer*. Para empezar, se editará el fichero de configuración */etc/Wazuh-indexer/opensearch.yml* editando los siguientes campos [Imagen 55]:

- `# gedit /etc/Wazuh-indexer/opensearch.yml`

a. *network.host*: establece la dirección de este nodo para el transporte y el tráfico HTTP. El nodo se vinculará a esta dirección y la usará como su dirección pública. Acepta tanto una dirección IP como un hostname. Se usará la misma dirección que se asignó en el fichero de configuración *config.yml* anteriormente para crear los certificados SSL. En este caso la dirección IP *192.168.22.10*.
b. node.name: el nombre del *Wazuh indexer* como se ha definido anteriormente en el fichero *config.yml*. En este caso se asignó el nombre *indexer*.
c. *cluster.initial_master_nodes*: Lista de los nombres de los nodos elegibles para el maestro. En este caso se dispondrá de un solo nodo para *Wazuh indexer*, que será *indexer,* tal y como está configurado en el fichero *config.yml*.
d. *plugins.security.nodes_dn*: Lista de los certificados del *Wazuh indexer*. Se editará la línea cambiando el nombre común (CN) por el de *indexer*, tal y como aparece en el fichero *config.yml*.

```
network.host: "192.168.22.10"
node.name: "indexer"
cluster.initial_master_nodes:
- "indexer"
#- "node-2"
#- "node-3"
cluster.name: "wazuh-cluster"
#discovery.seed_hosts:
#  - "node-1-ip"
#  - "node-2-ip"
#  - "node-3-ip"
node.max_local_storage_nodes: "3"
path.data: /var/lib/wazuh-indexer
path.logs: /var/log/wazuh-indexer

plugins.security.ssl.http.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.http.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.http.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.transport.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.transport.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.http.enabled: true
plugins.security.ssl.transport.enforce_hostname_verification: false
plugins.security.ssl.transport.resolve_hostname: false

plugins.security.authcz.admin_dn:
- "CN=admin,OU=Wazuh,O=Wazuh,L=California,C=US"
plugins.security.check_snapshot_restore_write_privileges: true
plugins.security.enable_snapshot_restore_privilege: true
plugins.security.nodes_dn:
- "CN=indexer,OU=Wazuh,O=Wazuh,L=California,C=US"
#- "CN=node-2,OU=Wazuh,O=Wazuh,L=California,C=US"
#- "CN=node-3,OU=Wazuh,O=Wazuh,L=California,C=US"
```

Imagen 55. *Fichero de configuración Wazuh indexer*

Se implementarán los certificados con el nombre del *Wazuh indexer* tal y como se encuentra definido en el fichero *config.yml* para cifrar las comunicaciones entre los componentes centrales de *Wazuh*.

- `# NODE_NAME=indexer`

- # mkdir /etc/Wazuh-indexer/certs
- # tar -xf ./Wazuh-certificates.tar -C /etc/Wazuh-indexer/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./admin.pem ./admin-key.pem ./root-ca.pem
- # mv -n /etc/Wazuh-indexer/certs/$NODE_NAME.pem /etc/Wazuh-indexer/certs/indexer.pem
- # mv -n /etc/Wazuh-indexer/certs/$NODE_NAME-key.pem /etc/Wazuh-indexer/certs/indexer-key.pem
- # chmod 500 /etc/Wazuh-indexer/certs
- # chmod 400 /etc/Wazuh-indexer/certs/*
- # chown -R Wazuh-indexer:Wazuh-indexer /etc/Wazuh-indexer/certs

Para terminar, se habilitará y se iniciará el servicio de *Wazuh indexer*.

- # systemctl daemon-reload
- # systemctl enable Wazuh-indexer
- # systemctl start Wazuh-indexer

4.8.3 Inicialización del clúster

Se ejecutará el script *indexer*-security-init.sh del *Wazuh indexer* para cargar la información de los nuevos certificados e iniciar el clúster [Imagen 56].

- # /usr/share/Wazuh-indexer/bin/indexer-security-init.sh

```
[root@indexer ~]# /usr/share/wazuh-indexer/bin/indexer-security-init.sh
************************************************************************
** This tool will be deprecated in the next major release of OpenSearch **
** https://github.com/opensearch-project/security/issues/1755           **
************************************************************************
Security Admin v7
Will connect to 192.168.22.10:9200 ... done
Connected as "CN=admin,OU=Wazuh,O=Wazuh,L=California,C=US"
OpenSearch Version: 2.8.0
Contacting opensearch cluster 'opensearch' and wait for YELLOW clusterstate ...
Clustername: wazuh-cluster
Clusterstate: GREEN
Number of nodes: 1
Number of data nodes: 1
.opendistro_security index does not exists, attempt to create it ... done (0-all replicas)
Populate config from /etc/wazuh-indexer/opensearch-security/
Will update '/config' with /etc/wazuh-indexer/opensearch-security/config.yml
   SUCC: Configuration for 'config' created or updated
Will update '/roles' with /etc/wazuh-indexer/opensearch-security/roles.yml
   SUCC: Configuration for 'roles' created or updated
Will update '/rolesmapping' with /etc/wazuh-indexer/opensearch-security/roles_mapping.yml
   SUCC: Configuration for 'rolesmapping' created or updated
Will update '/internalusers' with /etc/wazuh-indexer/opensearch-security/internal_users.yml
   SUCC: Configuration for 'internalusers' created or updated
Will update '/actiongroups' with /etc/wazuh-indexer/opensearch-security/action_groups.yml
   SUCC: Configuration for 'actiongroups' created or updated
Will update '/tenants' with /etc/wazuh-indexer/opensearch-security/tenants.yml
   SUCC: Configuration for 'tenants' created or updated
Will update '/nodesdn' with /etc/wazuh-indexer/opensearch-security/nodes_dn.yml
   SUCC: Configuration for 'nodesdn' created or updated
Will update '/whitelist' with /etc/wazuh-indexer/opensearch-security/whitelist.yml
   SUCC: Configuration for 'whitelist' created or updated
Will update '/audit' with /etc/wazuh-indexer/opensearch-security/audit.yml
   SUCC: Configuration for 'audit' created or updated
Will update '/allowlist' with /etc/wazuh-indexer/opensearch-security/allowlist.yml
   SUCC: Configuration for 'allowlist' created or updated
SUCC: Expected 10 config types for node {"updated_config_types":["allowlist","tenants","rolesmapping","n
odesdn","audit","roles","whitelist","internalusers","actiongroups","config"],"updated_config_size":10,"m
essage":null} is 10 (["allowlist","tenants","rolesmapping","nodesdn","audit","roles","whitelist","intern
alusers","actiongroups","config"]) due to: null
Done with success
```

Imagen 56. *Ejecución script indexer-security-init.sh*

Una vez cargada la información de los nuevos certificados correctamente, se comprobará que la instalación se ha realizado satisfactoriamente [Imagen 57].

- `# curl -k -u admin:admin https://192.168.22.10:9200`

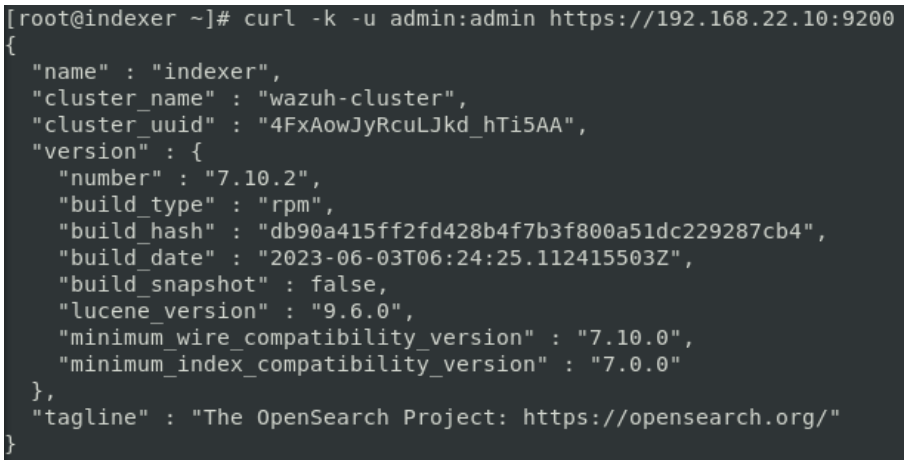


Imagen 57. *Comprobación de la correcta instalación de Wazuh indexer*

Para finalizar, se comprobará que el clúster funciona correctamente [Imagen 58].

- `# curl -k -u admin:admin https://192.168.22.10:9200/_cat/nodes?v`

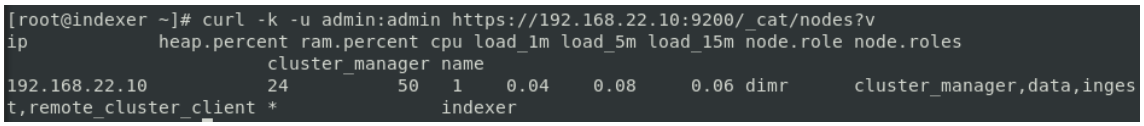


Imagen 58. *Comprobación del correcto funcionamiento del clúster*

Realizadas estas comprobaciones finales ya quedaría configurado el nodo *Indexer* que contendrá el *Wazuh indexer* instalado y listo para usarse en este clúster de *Wazuh*. A partir de aquí se procederá a la instalación de uno de los nodos del *Wazuh server* en los nodos *Nodo1, Nodo2* y *Nodo3*.

4.9 Instalación de *Wazuh server*

Para la generación de este punto se ha utilizado la guía de instalación de *Wazuh server* que se encuentra en la documentación de *Wazuh* [16]. El *Wazuh server* es un componente central de *Wazuh* e incluye al *Wazuh manager* y *Filebeat*. La instalación del *Wazuh server* se realizará de la misma manera y siguiendo los mismos pasos en las máquinas virtuales *Nodo1*, *Nodo2* y *Nodo3*. La configuración de *Wazuh server* se realizará por separado y de diferente manera en los nodos *Nodo1*, *Nodo2* y *Nodo3*, ya que uno será el nodo maestro y los otros dos los nodos esclavos o trabajadores.

4.9.1 Repositorio *Wazuh*

Se importará la clave GPG:

- `# rpm --import https://packages.Wazuh.com/key/GPG-KEY-WAZUH`

Luego se añadirá el repositorio [Imagen 59]:

- `# echo -e '[Wazuh]\ngpgcheck=1\ngpgkey=https://packages.Wazuh.com/key/GPG-KEY-WAZUH\nenabled=1\nname=EL-$releasever - Wazuh\nbaseurl=https://packages.Wazuh.com/4.x/yum/\nprotect=1' | tee /etc/yum.repos.d/Wazuh.repo`

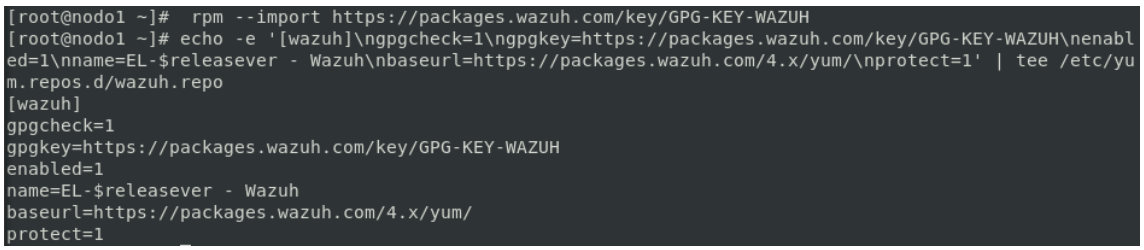


Imagen 59. *Repositorio Wazuh manager nodo1*

4.9.2 Instalación de *Wazuh manager*

Se procederá a instalar el paquete de *Wazuh manager*:

- `# yum -y install Wazuh-manager`

A continuación, se habilitará y se iniciará el servicio de *Wazuh manager*:

- `# systemctl daemon-reload`
- `# systemctl enable Wazuh-manager`
- `# systemctl start Wazuh-manager`

Para finalizar, se comprobará el estado para verificar el correcto funcionamiento de *Wazuh manager* [Imagen 60]:

- `# systemctl status Wazuh-manager`

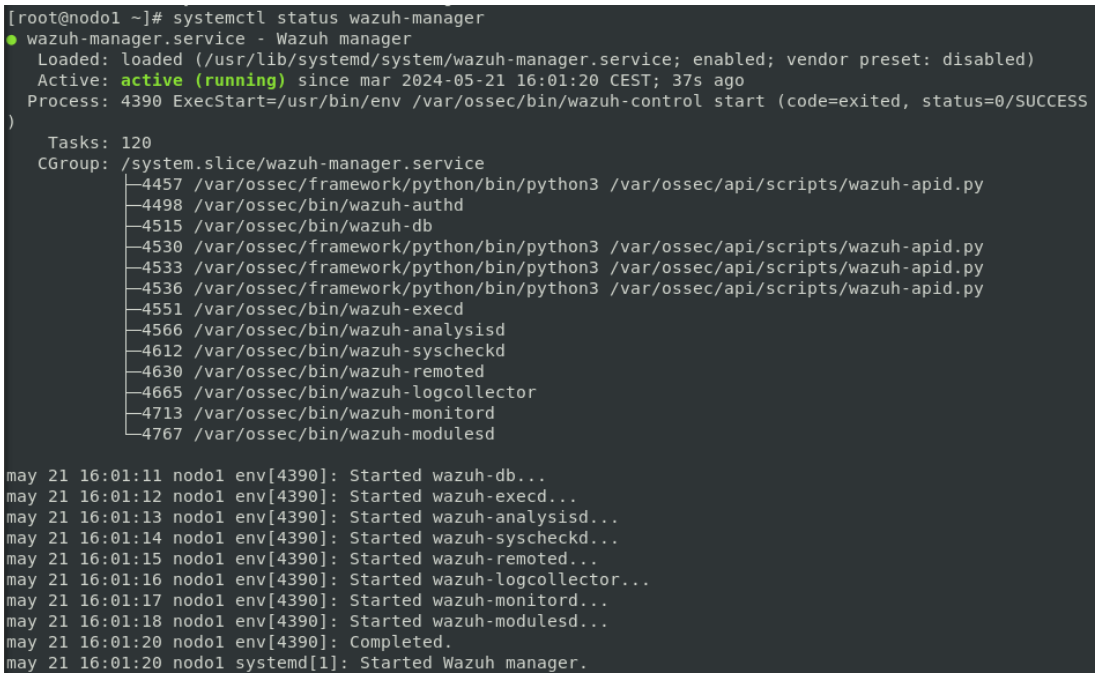


Imagen 60. *Estado Wazuh manager*

4.9.3 Instalación de Filebeat

Se instalará el paquete de Filebeat:

- `# yum -y install filebeat`

Una vez instalado se procederá a la configuración de Filebeat.

4.9.4 Configuración de Filebeat

Se descargará el fichero de Filebeat preconfigurado:

- `# curl -so /etc/filebeat/filebeat.yml https://packages.Wazuh.com/4.7/tpl/Wazuh/filebeat/filebeat.yml`

A continuación, se editará el fichero de configuración */etc/filebeat/filebeat.yml* modificando los siguientes parámetros:

a. Hosts: la lista de los nodos de *Wazuh indexer* a los que se conectará el nodo de *Wazuh server*. En este caso se dispone de un solo nodo de *Wazuh indexer* por lo que se pondrá 192.168.22.10:9200 [Imagen 61].

```
# Wazuh - Filebeat configuration file
output.elasticsearch:
  hosts: ["192.168.22.10:9200"]
  protocol: https
  username: ${username}
  password: ${password}
```

Imagen 61. *Configuración Filebeat*

Una vez modificado el fichero de configuración de Filebeat, se procederá a la creación de un almacén de claves de Filebeat para almacenar las credenciales de autenticación de forma segura:

- `# filebeat keystore create`

Se agregarán el nombre de usuario y la contraseña predeterminados *admin:admin* al almacén de claves:

- `# echo admin | filebeat keystore add username --stdin --force`
- `# echo admin | filebeat keystore add password --stdin --force`

Se descargará la plantilla de alertas para el *Wazuh indexer*:

- `# curl -so /etc/filebeat/Wazuh-template.json https://raw.githubusercontent.com/Wazuh/Wazuh/v4.7.3/extensions/elasticsearch/7.x/Wazuh-template.json`
- `# chmod go+r /etc/filebeat/Wazuh-template.json`

Finalmente se instalará el módulo de *Wazuh* para Filebeat [Imagen 62]:

- `# curl -s https://packages.Wazuh.com/4.x/filebeat/Wazuh-filebeat-0.3.tar.gz | tar -xvz -C /usr/share/filebeat/module`

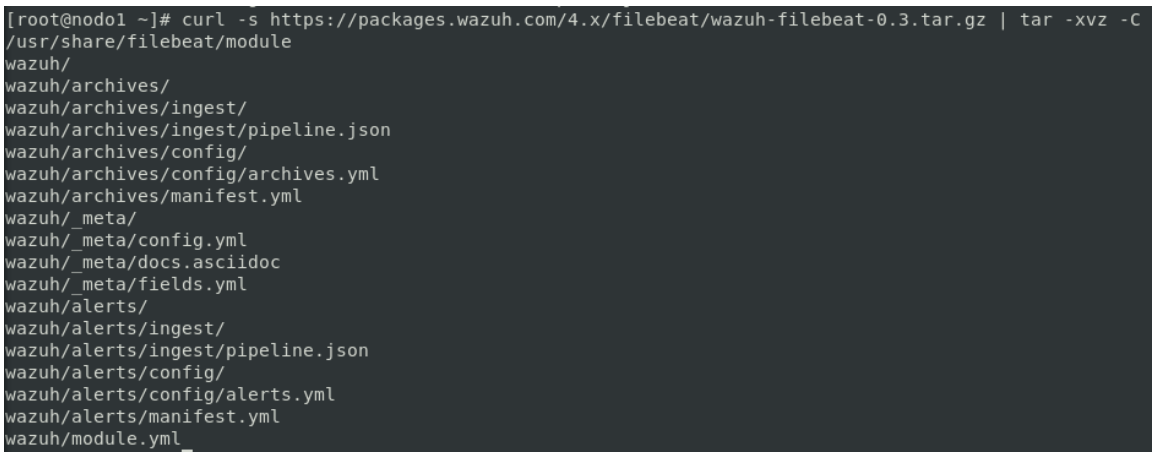


Imagen 62. *Instalación módulo de Wazuh para Filebeat*

4.9.5 Despliegue de certificados

Se asegurará que la copia del fichero *Wazuh-certificates.tar* creado durante la configuración inicial de *Wazuh indexer* se encuentra en el directorio de trabajo.

Se moverán los certificados a su correspondiente localización usando el nombre del nodo tal y como se encuentra configurado en el fichero *config,yml*. En el caso de que la instalación se esté realizando en el *Nodo1* se pondrá el nombre *nodo1*, si se realiza en el *Nodo2* el nombre que se pondrá será *nodo2* y si se realiza en el *Nodo3*

el nombre que se pondrá será *nodo3*. Ejemplo si se realiza en la máquina virtual *Nodo1*:

- `# NODE_NAME=nodo1`
- `# mkdir /etc/filebeat/certs`
- `# tar -xf ./Wazuh-certificates.tar -C /etc/filebeat/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem`
- `# mv -n /etc/filebeat/certs/$NODE_NAME.pem /etc/filebeat/certs/filebeat.pem`
- `# mv -n /etc/filebeat/certs/$NODE_NAME-key.pem /etc/filebeat/certs/filebeat-key.pem`
- `# chmod 500 /etc/filebeat/certs`
- `# chmod 400 /etc/filebeat/certs/*`
- `# chown -R root:root /etc/filebeat/certs`

4.9.6 Inicialización del servicio Filebeat

Se habilitará y se iniciará el servicio Filebeat:

- `# systemctl daemon-reload`
- `# systemctl enable filebeat`
- `# systemctl start filebeat`

A continuación, se habilitará el puerto 9200 en el *Wazuh indexer* que se encuentra en la máquina virtual *indexer* para la conexión con Filebeat. Primero se deben encontrar las zonas activas ya que solo habrá que aplicar las reglas para la zona o zonas que se encuentren activas.

- `# firewall-cmd --get-active-zones`

Ya que solo se encuentra la zona pubic, se procederá a abrir el puerto 9200 de la siguiente manera:

- `# firewall-cmd --zone=public --add-port=9200/tcp --permanent`
- `# firewall-cmd --reload`

Para terminar, en los nodos del *Wazuh server*, tanto en el Nodo1, Nodo2 como en el Nodo3, se comprobará que la instalación de Filebeat se ha realizado satisfactoriamente [Imagen 63].

- `# filebeat test output`

```
[root@localhost ~]# filebeat test output
elasticsearch: https://192.168.22.10:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 192.168.22.10
    dial up... OK
  TLS...
    security: server's certificate chain verification is enabled
    handshake... OK
    TLS version: TLSv1.3
    dial up... OK
  talk to server... OK
  version: 7.10.2
```

Imagen 63. *Filebeat test output*

4.10 Configuración *Wazuh server*

Después de completar la instalación de *Wazuh server* en cada uno de los nodos *Nodo1*, *Nodo2* y *Nodo3*, la configuración de *Wazuh server* se realizará por separado y de diferente manera en los nodos, ya que Nodo1 configurará como nodo maestro y los nodos Nodo2 y Nodo3 se configurarán como trabajadores.

Para la generación de este punto se ha utilizado la guía de instalación de *Wazuh server* que se encuentra en la documentación de *Wazuh* [5].

4.10.1 Configuración de *Wazuh server* en Nodo1

Se elegirá el Nodo1 como nodo maestro de *Wazuh server*. Se editará el fichero de configuración */var/ossec/etc/ossec.conf* con los siguientes parámetros [Imagen 64]:

- `# gedit /var/ossec/etc/ossec.conf &`

```
<cluster>
  <name>wazuh</name>
  <node_name>nodo1</node_name>
  <node_type>master</node_type>
  <key>380286a0a0482d5ab7bee6b1d40d187c</key>
  <port>1516</port>
  <bind_addr>0.0.0.0</bind_addr>
  <nodes>
      <node>192.168.22.11</node>
  </nodes>
  <hidden>no</hidden>
  <disabled>no</disabled>
</cluster>
```

Imagen 64. *Fichero configuración master Wazuh server*

- *name*: indica el nombre del clúster. En este caso *Wazuh*.
- *node_name*: indica el nombre del nodo actual. En este caso *nodo1*.
- *node_type*: especifica el rol del nodo. En este caso será *master*.

- *key*: la clave que se usa para encriptar la comunicación entre los nodos del clúster. Se usará el comando *uuidgen* y la clave generada será "380286a0a0482d5ab7bee6b1d40d187c"
- Port: indica el puerto para la comunicación del clúster, en este caso el 1516.
- *bind_addr*: es la IP de la red a la que el nodo debe escuchar las solicitudes. 0.0.0.0 para cualquier IP.
- *nodes*: Es la dirección IP del nodo maestro. En este caso *192.168.22.11*.
- *hidden*: muestra o esconde la información del clúster en las alertas generadas. En este caso debe ser *no*.
- *disabled*: indica si el nodo está habilitado o no en el clúster. En este caso deber ser *no*.

Se reiniciará el *Wazuh manager* para guardar los cambios en el clúster.

- `# systemctl restart Wazuh-manager`

4.10.2 Configuración de *Wazuh server* en Nodo2

Se elegirá el *Nodo2* como uno de los nodos trabajadores de *Wazuh server*. Se editará el fichero de configuración */var/ossec/etc/ossec.conf* con los siguientes parámetros [Imagen 65]:

```
<cluster>
  <name>wazuh</name>
  <node_name>node2</node_name>
  <node_type>worker</node_type>
  <key>380286a0a0482d5ab7bee6b1d40d187c</key>
  <port>1516</port>
  <bind_addr>0.0.0.0</bind_addr>
  <nodes>
      <node>192.168.22.11</node>
  </nodes>
  <hidden>no</hidden>
  <disabled>no</disabled>
</cluster>
```

Imagen 65. *Fichero configuración worker Nodo2 Wazuh server*

- *name*: indica el nombre del clúster. En este caso *Wazuh*.
- *node_name*: indica el nombre del nodo actual. En este caso *nodo2*.
- *node_type*: especifica el rol del nodo. En este caso será *worker*.
- *key*: la clave que se usa para encriptar la comunicación entre los nodos del clúster. La clave tiene que ser la misma en todos los nodos del clúster, así que se copiará la que se configuró para el nodo maestro.
- Port: indica el puerto para la comunicación del clúster, en este caso el 1516.
- *bind_addr*: es la IP de la red a la que el nodo debe escuchar las solicitudes. 0.0.0.0 para cualquier IP.
- *nodes*: Es la dirección IP del nodo maestro. En este caso *192.168.22.11*.

- *hidden*: muestra o esconde la información del clúster en las alertas generadas. En este caso debe ser *no*.
- Disabled: indica si el nodo está habilitado o no en el clúster. En este caso deber ser *no*.

Se reiniciará el *Wazuh manager* para guardar los cambios en el clúster.

- `# systemctl restart wazuh-manager`

4.10.3 Configuración de *Wazuh server* en Nodo3

Se elegirá el *Nodo3* como uno de los nodos trabajadores de *Wazuh server*. Se editará el fichero de configuración */var/ossec/etc/ossec.conf* con los siguientes parámetros [Imagen 66]:

```
<cluster>
  <name>wazuh</name>
  <node_name>nodo3</node_name>
  <node_type>worker</node_type>
  <key>380286a0a0482d5ab7bee6b1d40d187c</key>
  <port>1516</port>
  <bind_addr>0.0.0.0</bind_addr>
  <nodes>
      <node>192.168.22.11</node>
  </nodes>
  <hidden>no</hidden>
  <disabled>no</disabled>
</cluster>
```

Imagen 66. *Fichero configuración worker Nodo3 Wazuh server*

- *name*: indica el nombre del clúster. En este caso *Wazuh*.
- *node_name*: indica el nombre del nodo actual. En este caso *nodo3*.
- *node_type*: especifica el rol del nodo. En este caso será *worker*.
- *key*: la clave que se usa para encriptar la comunicación entre los nodos del clúster. La clave tiene que ser la misma en todos los nodos del clúster, así que se copiará la que se configuró para el nodo maestro.
- Port: indica el puerto para la comunicación del clúster, en este caso el 1516.
- *bind_addr*: es la IP de la red a la que el nodo debe escuchar las solicitudes. 0.0.0.0 para cualquier IP.
- *nodes*: Es la dirección IP del nodo maestro. En este caso *192.168.22.11*.
- *Hidden*: muestra o esconde la información del clúster en las alertas generadas. En este caso debe ser *no*.
- disabled: indica si el nodo está habilitado o no en el clúster. En este caso deber ser *no*.

Se reiniciará el *Wazuh manager* para guardar los cambios en el clúster.

- `# systemctl restart Wazuh-manager`

4.10.4 Testeo del clúster de *Wazuh server*

Se tendrá que habilitar el puerto 1516, que es el puerto por donde se realiza la conexión entre nodo maestro y los trabajadores en el clúster. Se hará en cada uno de los nodos del *Wazuh server*, tanto en el *Nodo1* como en los nodos *Nodo2* y *Nodo3*.

Primero se deben encontrar las zonas activas ya que solo habrá que aplicar las reglas para la zona o zonas que se encuentren activas.

- `# firewall-cmd --get-active-zones`

Ya que solo se encuentra la zona pubic, se procederá a abrir el puerto 1516 de la siguiente manera:

- `# firewall-cmd --zone=public --add-port=1516/tcp --permanent`
- `# firewall-cmd –reload`

Una vez modificado el firewall al abrir el puerto 1516 para la comunicación entre los nodos del clúster, se verificará que el clúster está habilitado y los nodos están conectados [Imagen 67]. La comprobación se puede realizar desde cualquiera de los nodos del clúster de *Wazuh server*, es decir, tanto desde *Nodo1* como desde los nodos *Nodo2* y *Nodo3*.

- `# /var/ossec/bin/cluster_control -l`

```
[root@nodo1 ~]# /var/ossec/bin/cluster_control -l
NAME   TYPE    VERSION  ADDRESS
nodo1  master  4.7.4    192.168.22.11
nodo2  worker  4.7.4    192.168.22.12
nodo3  worker  4.7.4    192.168.22.13
```

Imagen 67. *Output verificación del clúster Wazuh server*

Realizada esta comprobación final, el *Wazuh server* quedaría correctamente instalado y configurado en los nodos *Nodo1*, *Nodo2* y *Nodo3*. A continuación se pasará a la instalación y configuración de *Wazuh dashboard* en el nodo *Indexer*.

4.11 Instalación y configuración de *Wazuh dashboard*

La instalación y configuración de *Wazuh dashboard* se realizará en la máquina virtual *Indexer*. Para la generación de este punto se ha utilizado la guía de instalación de *Wazuh dashboard* que se encuentra en la documentación de *Wazuh* [17].

Instalación de *Wazuh dashboard*

Primero se instalarán los siguientes paquetes:

- `# yum install libcap`

A continuación, se instalará el paquete de *Wazuh dashboard*.

- `# yum -y install Wazuh-dashboard`

4.11.1 Configuración de *Wazuh dashboard*

Se editará el fichero de configuración */etc/Wazuh-dashboard/opensearch_dashboards.yml* modificando los siguientes valores [Imagen 68]:

- `# gedit /etc/Wazuh-dashboard/opensearch_dashboards.yml &`

```
server.host: 192.168.22.10
server.port: 443
opensearch.hosts: https://192.168.22.10:9200
opensearch.ssl.verificationMode: certificate
```

Imagen 68. *Fichero configuración Wazuh dashboard*

- *server.host*: especifica el host del servidor de *Wazuh dashboard*. Para permitir la conexión de usuarios remotos, se configurará con la dirección IP de *Wazuh dashboard* que será la dirección *192.168.22.10*. La dirección 0.0.0.0 aceptará todas la IP disponibles del host.
- *opensearch hosts*: La URL del *Wazuh indexer*. En este caso como se encuentra en el mismo nodo que el *Wazuh dashboard* será *https://192.168.22.10:9200*.

4.11.2 Despliegue de certificados

Se asegurará que la copia del fichero *Wazuh-certificates.tar* creado durante la configuración inicial de *Wazuh indexer* se encuentra en el directorio de trabajo.

Se moverán los certificados a su correspondiente localización usando el nombre del nodo tal y como se encuentra configurado en el fichero *config,yml*, en este caso será *dashboard*.

- `# NODE_NAME=dashboard`
- `# mkdir /etc/Wazuh-dashboard/certs`
- `# tar -xf ./Wazuh-certificates.tar -C /etc/Wazuh-dashboard/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem`
- `# mv -n /etc/Wazuh-dashboard/certs/$NODE_NAME.pem /etc/Wazuh-dashboard/certs/dashboard.pem`
- `# mv -n /etc/Wazuh-dashboard/certs/$NODE_NAME-key.pem /etc/Wazuh-dashboard/certs/dashboard-key.pem`
- `# chmod 500 /etc/Wazuh-dashboard/certs`
- `# chmod 400 /etc/Wazuh-dashboard/certs/*`
- `# chown -R Wazuh-dashboard:Wazuh-dashboard /etc/Wazuh-dashboard/certs`

4.11.3 Inicialización del servicio *Wazuh dashboard*

Se habilitará y se inicializará el servicio de *Wazuh dashboard*:

- `# systemctl daemon-reload`
- `# systemctl enable Wazuh-dashboard`
- `# systemctl start Wazuh-dashboard`

A continuación, se accederá a la interfaz web de *Wazuh* poniendo la dirección IP de *Wazuh dashboard* en la URL de un buscador web [Imagen 69].

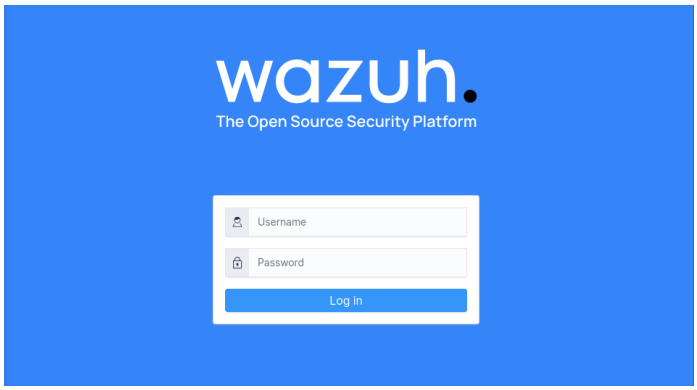


Imagen 69. *Interfaz web Wazuh dashboard*

Se editará el fichero */usr/share/Wazuh-dashboard/data/Wazuh/config/Wazuh.yml* y se reemplazará el valor de url por la dirección IP del nodo maestro de *Wazuh server* [Imagen 70].

- `# gedit /usr/share/Wazuh-dashboard/data/Wazuh/config/Wazuh.yml &`

```
hosts:
  - default:
      url: https://192.168.22.11
      port: 55000
      username: wazuh-wui
      password: wazuh-wui
      run_as: false
```

Imagen 70. *Fichero Wazuh.yml de Wazuh dashboard*

Para la correcta conexión con la API del *Wazuh server*, en el Nodo1, que es el que está configurado como nodo maestro, se deberá habilitar el puerto 55000 en el firewall.

- `# firewall-cmd --zone=public --add-port=55000/tcp --permanent`
- `# firewall-cmd --reload`

Configurado el firewall se volverá al buscador de web, en este caso Firefox, y se escribirá en la URL la dirección *192.168.22.10*, que es la correspondiente con la dirección IP del *Wazuh dashboard*. Aparecerá un inicio de sesión donde se

requerirán unas credenciales. En este caso serán *admin* para el usuario y *admin* para la contraseña [Imagen 71].

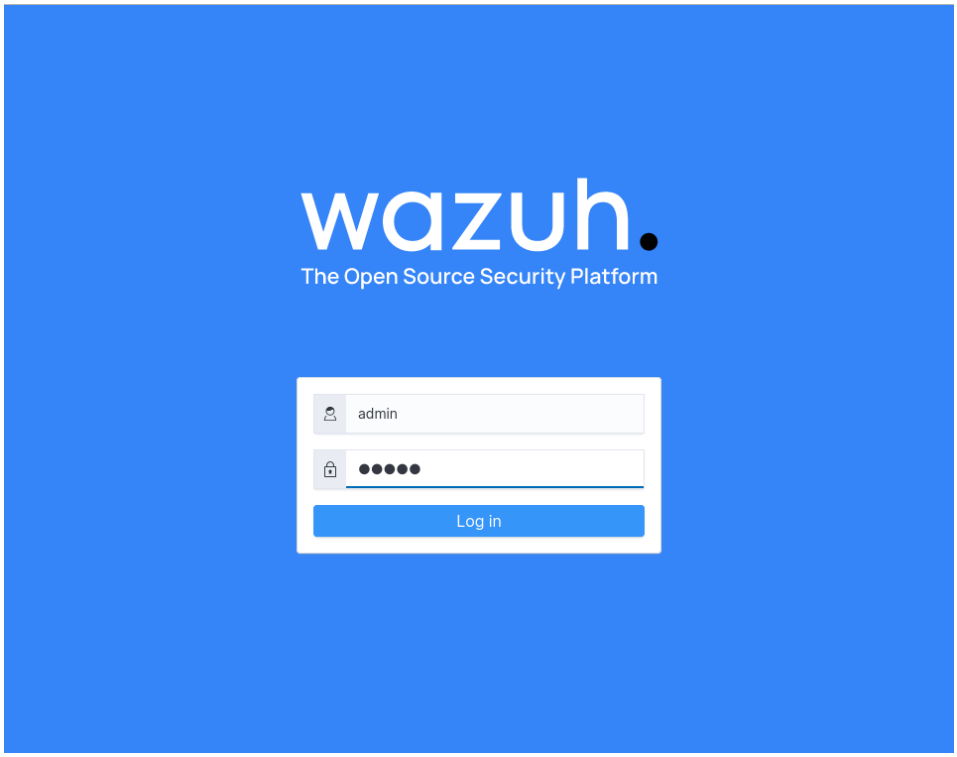


Imagen 71. *Inicio de sesión Wazuh dashboard*

Una vez se acceda al iniciar sesión correctamente, se realizará la comprobación de conexión de la API de *Wazuh* [Imagen 72].

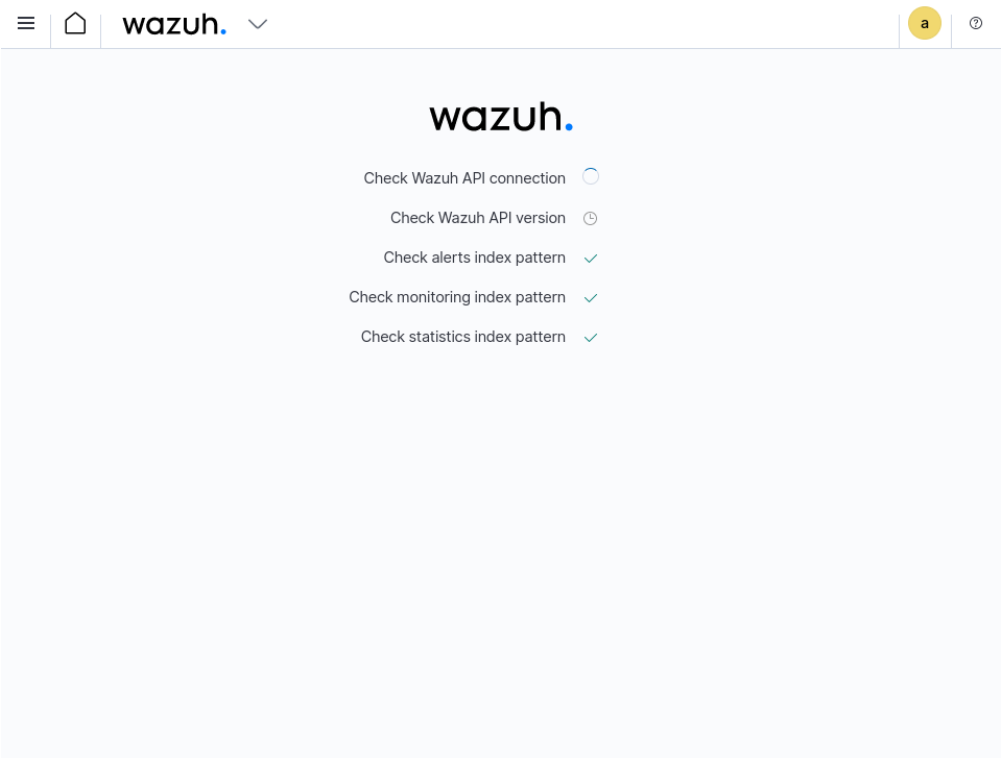


Imagen 72. *Conexión con la API de Wazuh*

Una vez comprobada y realizada la conexión con la API de *Wazuh* significará que la instalación y configuración de *Wazuh* se ha realizado satisfactoriamente, aparecerá el menú principal de *Wazuh dashboard* [Imagen 73].

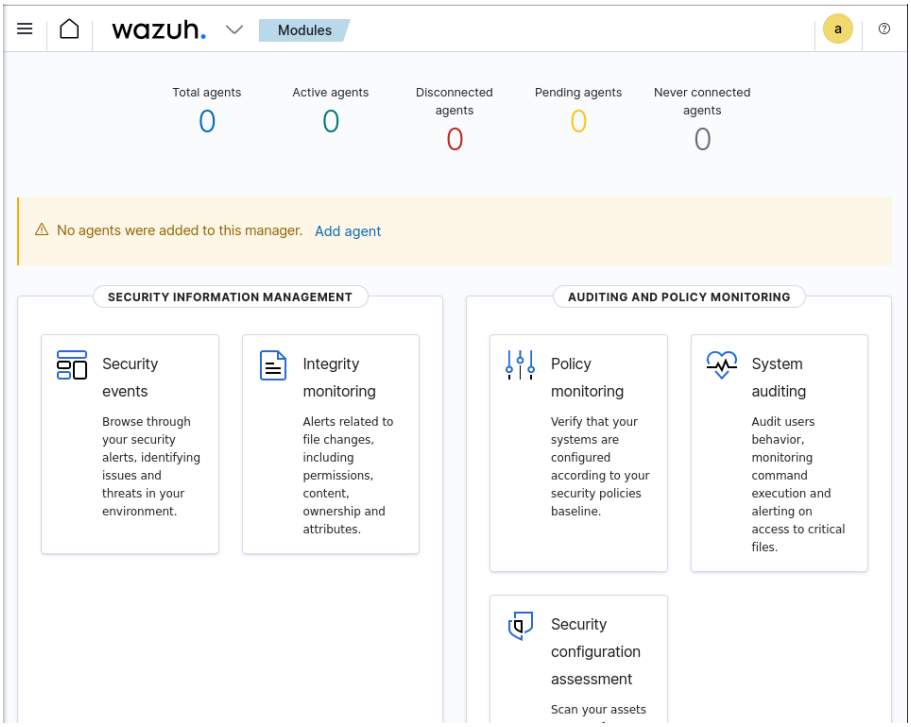


Imagen 73. *Menú principal de Wazuh dashboard*

4.12 Configuración firewall y *SElinux*

Para la correcta conexión de los agentes con el servidor se deberá habilitar en todos los nodos de *Wazuh server* el puerto 1514 tanto en el nodo maestro Nodo1 como en los trabajadores Nodo2 y Nodo3.

- `# firewall-cmd --zone=public --add-port=1514/tcp --permanent`
- `#firewall-cmd --reload`

En el nodo maestro del *Wazuh server*, que es el Nodo1, se deberá habilitar el puerto 1515 con protocolo TCP, el cual se utiliza para el registro de un nuevo agente al *Wazuh server*.

- `# firewall-cmd --zone=public --add-port=1515/tcp --permanent`
- `#firewall-cmd --reload`

En las máquinas Nodo1, Nodo2 y Nodo3 habrá que aplicar ciertas reglas de *SElinux* para permitir el tráfico en los puertos necesarios. Para ello se permitirá la conexión del servicio HTTP para la futura conexión con el balanceador de carga de *Nginx*.

- `# setsebool -P httpd_can_network_connect 1`

A continuación, se aplican las reglas para permitir el tráfico de los diferentes puertos necesarios 1514, 1515, 1516:

- `# semanage port -a -t http_port_t -p tcp 1514`
- `# semanage port -a -t http_port_t -p tcp 1515`
- `# semanage port -a -t http_port_t -p tcp 1516`

5. Balanceador de carga *Nginx*

Un balanceador de carga [18], también conocido como *Load Balancer* en inglés, es una herramienta crucial que se utiliza para distribuir de manera equitativa la carga de trabajo entre varios servidores, que actúan como nodos en una red.

Este dispositivo desempeña el papel de intermediario entre los clientes que realizan solicitudes y los servidores o recursos que procesan dichas solicitudes. Cuando recibe las solicitudes entrantes, el balanceador de carga las redirige a los servidores disponibles siguiendo una estrategia de distribución preconfigurada, asegurando así un rendimiento óptimo y evitando la sobrecarga de los servidores.

En este trabajo se usará un balanceador de carga de *Nginx* para capturar todos los eventos de los agentes y distribuirlos entre los diferentes nodos trabajadores del clúster, que son los nodos *Nodo2* y *Nodo3*.

5.1 Creación de la máquina virtual para balanceador de carga

Se clonará la máquina virtual *MV* creada en el punto *"2. Creación y configuración de una MV en VirtualBox"*. El proceso de clonación se realizará igual que en apartados anteriores. A la nueva máquina virtual se le pondrá el nombre de LB

A continuación, se pasará a la configuración de red de la máquina virtual *LB*. Para ello, se usará el comando *ifconfig* para comprobar la dirección IP del adaptador de red [Imagen 74].

- `# ifconfig`

```
[root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.22.5  netmask 255.255.255.0  broadcast 192.168.22.255
        inet6 fe80::2dd0:e120:d2a3:8cbd  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:7e:e5:df  txqueuelen 1000  (Ethernet)
        RX packets 437  bytes 250710 (244.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 468  bytes 35736 (34.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Imagen 74. *Resultado Ifconfig máquina Nginx*

Como se podrá observar, al ser una máquina clonada, tendrá la misma dirección IP que la máquina madre *MV,* y su dirección IP es la *192.168.22.5*. En este caso se cambiará a la dirección IP *192.168.22.9*. Para ello se editará el fichero de configuración del adaptador de red.

- `# gedit /etc/sysconfig/network-scripts/ifcfg-enp0s3 &`

Se editarán los siguientes campos:

- `UUID=" c4db1851-1313-4202-bab8-27680dd798e7"`
- `IPADDR="192.168.22.9"`

Para la generación del UUID se ha utilizado el siguiente comando:

- `# uuidgen`

A continuación, se actualizarán los cambios realizados en el adaptador de red reiniciando el servicio de red.

- `# systemctl restart network`

Se comprobará que se tiene conexión a internet usando el comando ping:

- `# ping google.es`

Finalmente, se editará el hostname de la máquina virtual.

- `# hostnamectl set-hostname nginx`

Se reiniciará la máquina virtual.

- `# reboot`

5.2 Instalación y configuración de *Nginx*

5.2.1 Instalación

Se instalará *Nginx* en la máquina virtual *Nginx* y luego se editará su configuración para el balanceo de carga TCP en el clúster de *Wazuh*.

Primero se descargarán los paquetes en la página web oficial de *Nginx* [19]. Una vez se haya accedido al sitio web, se hará clic en *"RHEL and derivatives"* [Imagen 75] que es la distribución para el sistema operativo de la máquina LB, que es *CentOS 7*.

Supported distributions and versions

nginx packages are available for the following Linux distributions and versions:

RHEL and derivatives

Version	Supported Platforms
7.4+	x86_64, aarch64/arm64
8.x	x86_64, aarch64/arm64, s390x
9.x	x86_64, aarch64/arm64, s390x

Debian

Version	Supported Platforms
11.x "bullseye"	x86_64, aarch64/arm64
12.x "bookworm"	x86_64, aarch64/arm64

Imagen 75. *Distribuciones y versiones de Nginx*

A continuación, se seguirá la guía de instalación ofrecida por la página oficial de *Nginx*, en la sección de *"RHEL and derivatives"*. Se instalarán los prerrequisitos:

- `# yum install yum-utils`

Para configurar el repositorio yum, se creará el archivo /etc/yum.repos.d/nginx.repo añadiendo el contenido mostrado en la siguiente tabla [Tabla 6]:

- `# gedit /etc/yum.repos.d/nginx.repo &`

Tabla 6. *Contenido archivo nginx.repo*

```
[nginx-stable]
name=nginx stable repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearc/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
module_hotfixes=true
```

Para finalizar, se instalará *Nginx* ejecutando el siguiente comando:

- `# yum install nginx`

Se arrancará el servicio de nginx y se comprobará su estado para verificar que se ha instalado correctamente [Imagen 76].

- `# systemctl start nginx`
- `# systemctl status nginx`

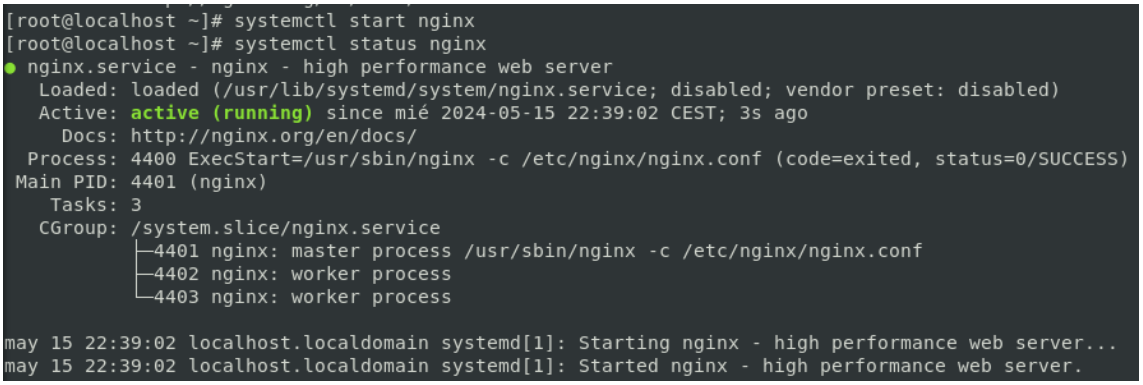


Imagen 76. *Estado del servicio nginx*

Verificada la correcta instalación se pasará a la configuración del balanceador de carga de *Nginx*.

5.2.2 Configuración

La manera en la que *Nginx* y sus módulos funcionan vienen determinados por el fichero de configuración de *Nginx* [20], */etc/nginx/nginx.conf*. Se añadirá al fichero el contenido de la siguiente tabla [Tabla 7].

- `# gedit /etc/nginx/nginx.conf &`

Tabla 7. *Contenido archivo nginx.conf*

```
stream {
    upstream cluster {
        hash $remote_addr consistent;
        server <WAZUH_WORKER1_IP_ADDRESS>:1514;
        server <WAZUH_WORKER2_IP_ADDRESS>:1514;
    }
    upstream master {
        server <WAZUH_MASTER_IP_ADDRESS>:1515;
    }
    server {
        listen 1514;
        proxy_pass cluster;
    }
    server {
        listen 1515;
        proxy_pass master;
    }
}
```

Se editará el fichero de la siguiente manera:

- <*WAZUH_MASTER*_IP_ADDRESS> se reemplazará por la dirección IP del nodo maestro *Nodo1* del clúster de *Wazuh*, la dirección *192.168.22.11*.
- <*WAZUH_WORKER*1_IP_ADDRESS> se reemplazará por la dirección IP del nodo trabajador *Nodo2* del clúster de *Wazuh*, la dirección *192.168.22.12*.
- <*WAZUH_WORKER*2_IP_ADDRESS> se reemplazará por la dirección IP del nodo trabajador *Nodo3* del clúster de *Wazuh*, la dirección *192.168.22.13*.

A continuación, se realizará una configuración más específica para mejorar el rendimiento del balanceo de carga TCP, actualizando el archivo de configuración de *Nginx* con nuevos parámetros. La actualización de la configuración asegura un balanceo de carga TCP eficiente y seguro, con mecanismos de monitoreo y chequeos de salud, llamados *Health Checks* [21] para garantizar la alta disponibilidad del servicio. Para ello se ha utilizado el manual de *Nginx* sobre el balanceo de carga TCP/UDP [22].

Primero se definirán los servidores de backend [Imagen 77]:

- *zone upstream_cluster 64k*: define una zona compartida llamada *"upstream_cluster"* con 64KB de memoria y almacenar el estado de los servidores.

- Parámetros *"max_fails"* y *"fail_timeout"*: Si el servidor falla 3 veces en un periodo de 20 segundos, éste se considerará inactivo.

```
upstream cluster {
    zone upstream_cluster 64k;
    server 192.168.22.12:1514 max_fails=3 fail_timeout=20s;  # Nodo2
    server 192.168.22.13:1514 max_fails=3 fail_timeout=20s;  # Nodo3
}
```

Imagen 77. *Servidores backend en archivo nginx.conf*

Después se configurará el servicio TCP en el servidor del balanceo de carga [Imagen 78]:

- listen 1514: el balanceador de carga escucha en el puerto 1514
- Parámetro "*proxy_pass cluster*": se redirige el tráfico al grupo de nodos definido en "*upstream_cluster*".
- Parámetro "*proxy_timeout*": tiempo de espera máximo para las conexiones proxy de 300 segundos.
- Parámetros de *health check* "*interval*", "*fails*" y "*passes*": se realiza un chequeo de salud del nodo cada 15 segundos. Si el servidor falla 3 veces consecutivas, se considerará como inactivo y debe pasar el chequeo 2 veces consecutivas para ser considerado activo nuevamente.
- Parámetro "*proxy_connect_timeout*": tiempo máximo para establecer una conexión con el servidor de *backend* de 1 segundo.

```
server {
    listen 1514;
    proxy_pass cluster;
    proxy_timeout 300s;
    health_check interval=15s fails=3 passes=2;
    proxy_connect_timeout 1s;
}
```

Imagen 78. *Servidor balanceo de carga en archivo etc.conf*

Para finalizar, se configurará también un archivo para monitorizar el balanceo de carga TCP con los siguientes parámetros [Imagen 79]:

- *"$remote_addr"*: Dirección IP del cliente.
- "*$time_local*": Hora local del servidor.
- "*$protocol*": Protocolo utilizado.
- "*$status*": Estado de la conexión.
- "*$bytes_sent*": Bytes enviados al cliente.
- "*$bytes_received*": Bytes recibidos del cliente.
- "*$session_time*": Duración de la sesión.
- "*$upstream_addr*": Dirección del servidor de backend.

```
log_format tcp_log '$remote_addr [$time_local] '
                   '$protocol $status $bytes_sent $bytes_received '
                   '$session_time $upstream_addr';

access_log /var/log/nginx/tcp_access.log tcp_log;
```

Imagen 79. *Archivo de monitorización en nginx.conf*

También se especificará la ubicación del archivo donde se devuelven los logs "*access_log /var/log/nginx/tcp_access.log tcp_log;*". Los logs de acceso para el balanceo de carga TCP se escribirán en ese archivo usando el formato definido en "*tcp_log*".

5.3 Configuración firewall y *SElinux*

Para la correcta conexión de los agentes con el servidor se deberá el puerto 1514.

- `# firewall-cmd --zone=public --add-port=1514/tcp --permanent`
- `#firewall-cmd --reload`

Se deberá habilitar el puerto 1515 con protocolo TCP, el cual se utiliza para el registro de un nuevo agente al *Wazuh server*.

- `# firewall-cmd --zone=public --add-port=1515/tcp --permanent`
- `#firewall-cmd --reload`

Hay que aplicar ciertas reglas de *SElinux* para permitir el tráfico en los puertos necesarios 1514, 1515, 1516. Para ello se permitirá la conexión del servicio HTTP.

- `# setsebool -P httpd_can_network_connect 1`

A continuación, se aplican las reglas para permitir el tráfico de los diferentes puertos necesarios para la conexión con el clúster de *Wazuh* y los agentes:

- `# semanage port -a -t http_port_t -p tcp 1514`
- `# semanage port -a -t http_port_t -p tcp 1515`
- `# semanage port -a -t http_port_t -p tcp 1516`

Una vez se hayan aplicado las reglas de *SElinux* se pasará a reiniciar el servicio de *Nginx* para aplicar los cambios de configuración del fichero *nginx*.conf.

- *# systemctl restart nginx*

6. Instalación y configuración de *Wazuh agent*

A continuación, se configurarán los agentes que se conectarán al servidor de *Wazuh*. Primero habrá que configurar varias máquinas virtuales para la creación de diferentes agentes que se configurarán como *Wazuh agents*. Para ello se clonará la máquina virtual *MV* creada en el punto *"2. Creación y configuración de una MV en VirtualBox"*. El proceso de clonación se realizará igual que en apartados anteriores.

En este caso se crearán 4 máquinas virtuales a partir de la clonación de la máquina virtual *MV*. Se llamarán *Agente1*, *Agente2*, *Agente3* y *Agente4* respectivamente. Se procederá a la configuración de red de las máquinas clonadas teniendo en cuenta que deberán estar en la misma red que las maquinas que contienen los componentes principales de *Wazuh*, que es la *Red Nat*, creada también en apartados anteriores.

6.1 Configuración e instalación en Agente1

6.1.1 Red

Se pasará a la configuración de red de la máquina virtual *Agente1*. Para ello, se usará el comando *ifconfig* para comprobar la dirección IP del adaptador de red [Imagen 80].

- `# ifconfig`

```
[root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.22.5  netmask 255.255.255.0  broadcast 192.168.22.255
        inet6 fe80::2dd0:e120:d2a3:8cbd  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:7e:e5:df  txqueuelen 1000  (Ethernet)
        RX packets 437  bytes 250710 (244.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 468  bytes 35736 (34.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Imagen 80. *Resultado Ifconfig Agente1*

Como se podrá observar, al ser una máquina clonada, tendrá la misma dirección IP que la máquina madre *MV,* y su dirección IP es la *192.168.22.5*. En este caso se cambiará a la dirección IP *192.168.22.14*. Para ello se editará el fichero de configuración del adaptador de red.

- `# gedit /etc/sysconfig/network-scripts/ifcfg-enp0s3 &`

Se editarán los siguientes campos:

- `UUID="39b201b0-dad4-4899-9e94-c2b3f355dbf5"`
- `IPADDR="192.168.22.14"`

Para la generación del UUID se ha utilizado el siguiente comando:

- `# uuidgen`

A continuación, se actualizarán los cambios realizados en el adaptador de red reiniciando el servicio de red.

- `# systemctl restart network`

Se comprobará que se tiene conexión a internet usando el comando ping:

- `# ping google.es`

Finalmente, se editará el hostname de la máquina virtual.

- `# hostname set-hostname agente1`

Se reiniciará la máquina virtual.

- `# reboot`

6.1.2 Configuración firewall y *SElinux*

Para la correcta conexión de los agentes con el servidor se deberá el puerto 1514.

- `# firewall-cmd --zone=public --add-port=1514/tcp --permanent`
- `#firewall-cmd --reload`

Se deberá habilitar el puerto 1515 con protocolo TCP, el cual se utiliza para el registro de un nuevo agente al *Wazuh server*.

- `# firewall-cmd --zone=public --add-port=1515/tcp --permanent`
- `#firewall-cmd --reload`

Hay que aplicar ciertas reglas de *SElinux* para permitir el tráfico en los puertos necesarios 1514, 1515, 1516. Para ello se permitirá la conexión del servicio HTTP.

- `# setsebool -P httpd_can_network_connect 1`

A continuación, se aplican las reglas para permitir el tráfico de los diferentes puertos necesarios para la conexión con el clúster de *Wazuh* y los agentes:

- `# semanage port -a -t http_port_t -p tcp 1514`
- `# semanage port -a -t http_port_t -p tcp 1515`

6.1.3 Instalación *Wazuh* agent

Primero habrá que situarse en la máquina virtual *Indexer*, que es donde se encuentra instalado el *Wazuh dashboard*. Se accederá a este último mediante el buscador Web y se iniciará sesión para entrar en el panel de control de nuestro servidor de *Wazuh*.

Al entrar después del inicio de sesión aparecerá un *warning* indicando: "*No agents were added to this manager"* [Imagen 81]. Esto quiere decir que no se ha añadido ningún agente al servidor de *Wazuh*. Se hará *clic* en *"Add agent"* para añadir un nuevo agente.

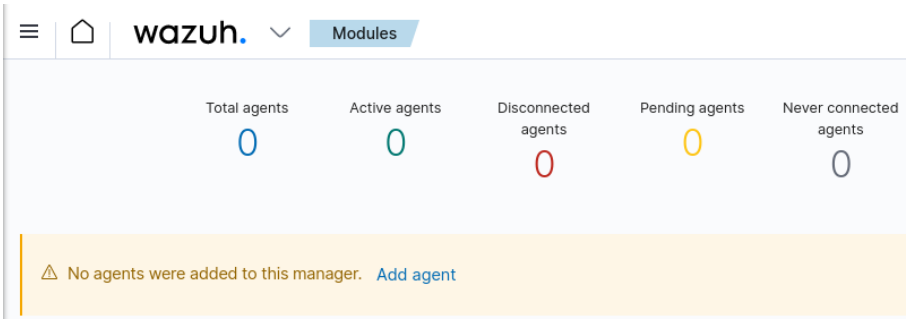


Imagen 81. *Añadir agente warning*

Se abrirá una ventana llamada *"Deploy new Agent"* donde habrá que rellenar ciertos parámetros [Imagen 82]:

- 1. Seleccionar el paquete para descargar e instalar en su sistema. Se seleccionará en *Linux* el paquete "*RPM amd64"*.
- 2. *Server address*. Se introducirá la dirección IP que el agente usa para comunicarse con el servidor, en este caso *192.168.22.11* que es la dirección del nodo *master* del *Wazuh server*.
- Configuraciones opcionales. Se asignará un nombre al agente, que en este caso será agente1 y se dejará la opción de grupo en "*Default"*.

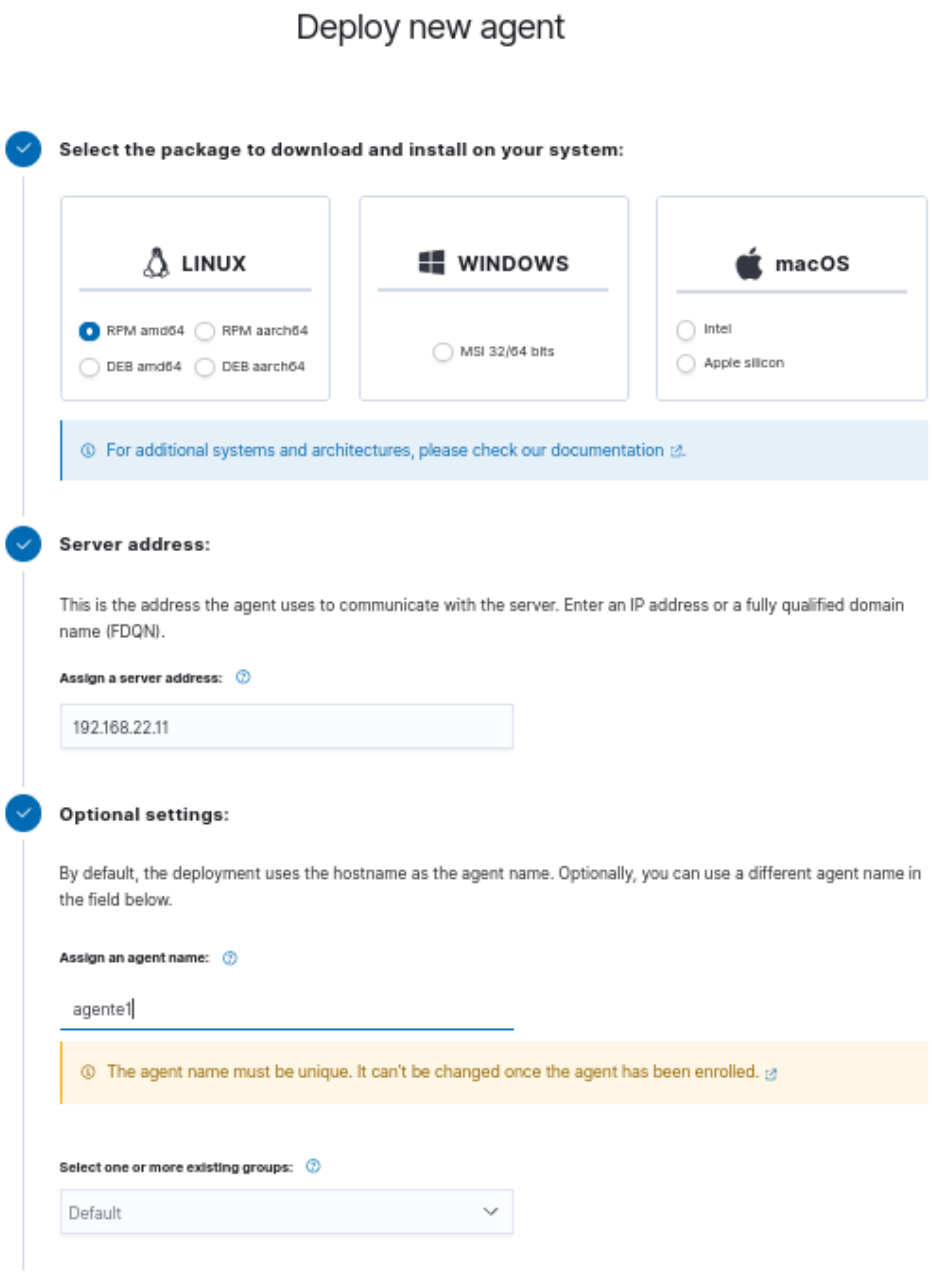


Imagen 82. *Deploy new agent configuración*

Editados los parámetros anteriores, se generará un comando [Imagen 83] que incluye datos añadidos y que se ejecutará en la máquina virtual donde se quiere instalar el nuevo agente, es decir, en la máquina virtual *Agente1*.

- `# curl -o Wazuh-agent-4.7.4-1.x86_64.rpm https://packages.Wazuh.com/4.x/yum/Wazuh-agent-4.7.4-1.x86_64.rpm && sudo WAZUH_MANAGER='192.168.22.11' WAZUH_AGENT_NAME='agente1' rpm -ihv Wazuh-agent-4.7.4-1.x86_64.rpm`

Run the following commands to download and install the agent:

```
curl -o wazuh-agent-4.7.4-1.x86_64.rpm https://packages.wazuh.com/4.x/yum/wazuh-agent-4.7.4-1.x86_64.rpm && sudo WAZUH_MANAGER='192.168.22.11' WAZUH_AGENT_NAME='agente1' rpm -ihv wazuh-agent-4.7.4-1.x86_64.rpm
```

Imagen 83. *Comando generado en Deploy a new agent*

Se editará el archivo de configuración ossec.conf [20] del agente cambiando la dirección IP del servidor a la del balanceador de carga de *Nginx* que es la *192.168.22.9* [Imagen 84].

- `# gedit /var/ossec/etc/ossec.conf &`

```
<server>
  <address>192.168.22.9</address>
  <port>1514</port>
  <protocol>tcp</protocol>
</server>
```

Imagen 84. *Archivo configuración Wazuh agent en agente1*

Una vez se haya instalado correctamente se habilitará y se iniciará el agente.

- `# systemctl daemon-reload`
- `# systemctl enable Wazuh-agent`
- `# systemctl start Wazuh-agent`

Para finalizar, de vuelta en la máquina virtual *Indexer*, en el panel de control de *Wazuh*, se hará *clic* en *"Close"* en la ventana que se encontraba aún abierta *"Deploy a new agent"*. Se abrirá la ventana de *"Agents"* del panel de control de *Wazuh*, donde se podrá verificar que se ha añadido correctamente un nuevo agente llamado *agente1* [Imagen 85].

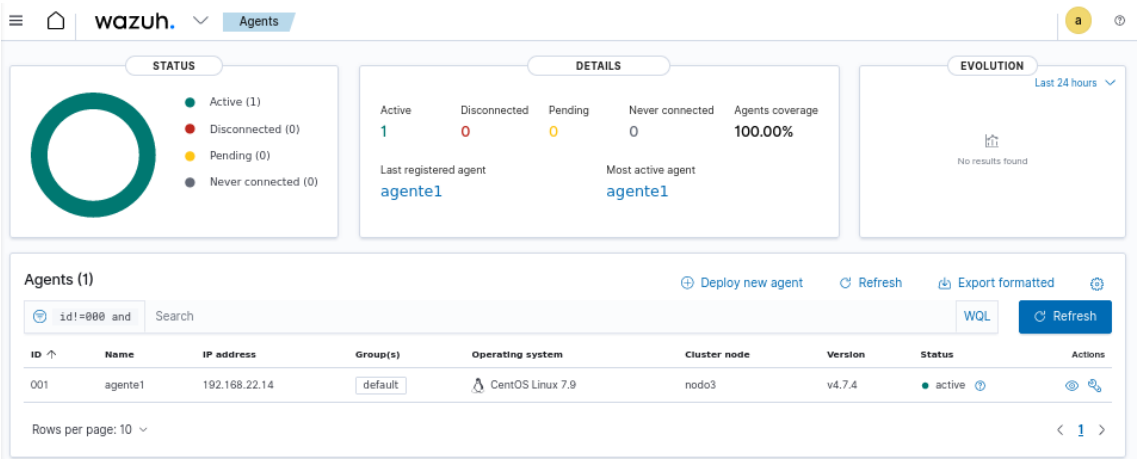


Imagen 85. *Agente agente1 en apartado Agents*

6.2 Configuración e instalación en Agente2

6.1.1 Red

Se pasará a la configuración de red de la máquina virtual *Agente2*. Para ello, se usará el comando *ifconfig* para comprobar la dirección IP del adaptador de red [Imagen 86].

- `# ifconfig`

```
[root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.22.5  netmask 255.255.255.0  broadcast 192.168.22.255
        inet6 fe80::2dd0:e120:d2a3:8cbd  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:7e:e5:df  txqueuelen 1000  (Ethernet)
        RX packets 437  bytes 250710 (244.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 468  bytes 35736 (34.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Imagen 86. *Resultado Ifconfig Agente2*

Como se podrá observar, al ser una máquina clonada, tendrá la misma dirección IP que la máquina madre *MV,* y su dirección IP es la *192.168.22.5*. En este caso se cambiará a la dirección IP *192.168.22.15*. Para ello se editará el fichero de configuración del adaptador de red.

- `# gedit /etc/sysconfig/network-scripts/ifcfg-enp0s3 &`

Se editarán los siguientes campos:

- `UUID="37846c53-e551-4523-b991-697abc68d4f1"`
- `IPADDR="192.168.22.15"`

Para la generación del UUID se ha utilizado el siguiente comando:

- `# uuidgen`

A continuación, se actualizarán los cambios realizados en el adaptador de red reiniciando el servicio de red.

- `# systemctl restart network`

Se comprobará que se tiene conexión a internet usando el comando ping:

- `# ping google.es`

Finalmente, se editará el hostname de la máquina virtual.

- `# hostname set-hostname agente2`

Se reiniciará la máquina virtual.

- `# reboot`

6.2.2 Configuración firewall y *SElinux*

Para la correcta conexión de los agentes con el servidor se deberá el puerto 1514.

- `# firewall-cmd --zone=public --add-port=1514/tcp --permanent`
- `#firewall-cmd --reload`

Se deberá habilitar el puerto 1515 con protocolo TCP, el cual se utiliza para el registro de un nuevo agente al *Wazuh server*.

- `# firewall-cmd --zone=public --add-port=1515/tcp --permanent`
- `#firewall-cmd --reload`

Hay que aplicar ciertas reglas de *SElinux* para permitir el tráfico en los puertos necesarios 1514, 1515, 1516. Para ello se permitirá la conexión del servicio HTTP.

- `# setsebool -P httpd_can_network_connect 1`

A continuación, se aplican las reglas para permitir el tráfico de los diferentes puertos necesarios para la conexión con el clúster de *Wazuh* y los agentes:

- `# semanage port -a -t http_port_t -p tcp 1514`
- `# semanage port -a -t http_port_t -p tcp 1515`

6.2.3 Instalación *Wazuh* agent

A continuación, habrá que situarse en la máquina virtual *Indexer*, que es donde se encuentra instalado el *Wazuh dashboard*. Se accederá a este último mediante el buscador Web y se iniciará sesión para entrar en el panel de control de nuestro servidor de *Wazuh*.

Se hará *clic* en la barra de navegación sobre el simbolito que representa una flecha hacia abajo, que abrirá un menú desplegable y seguidamente se hará *clic* en el apartado *"Agents"* [Imagen 87].

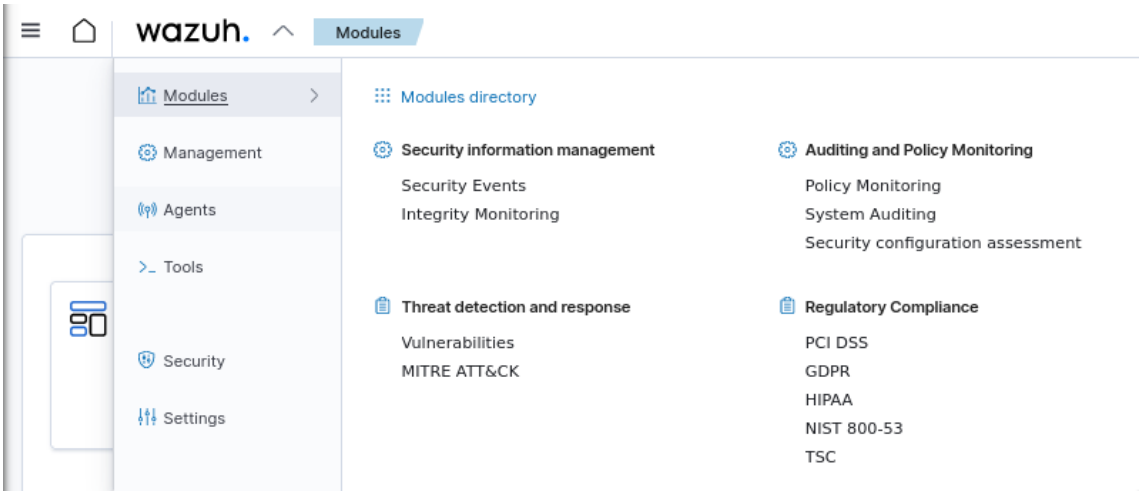


Imagen 87. *Menú desplegable Wazuh*

Una vez accedido al apartado *"Agents"* que es donde se encuentran reflejados los agentes, se irá a la parte de abajo donde se encuentran listados los agentes y se hará *clic* en *"Deploy new agent"* [Imagen 88].

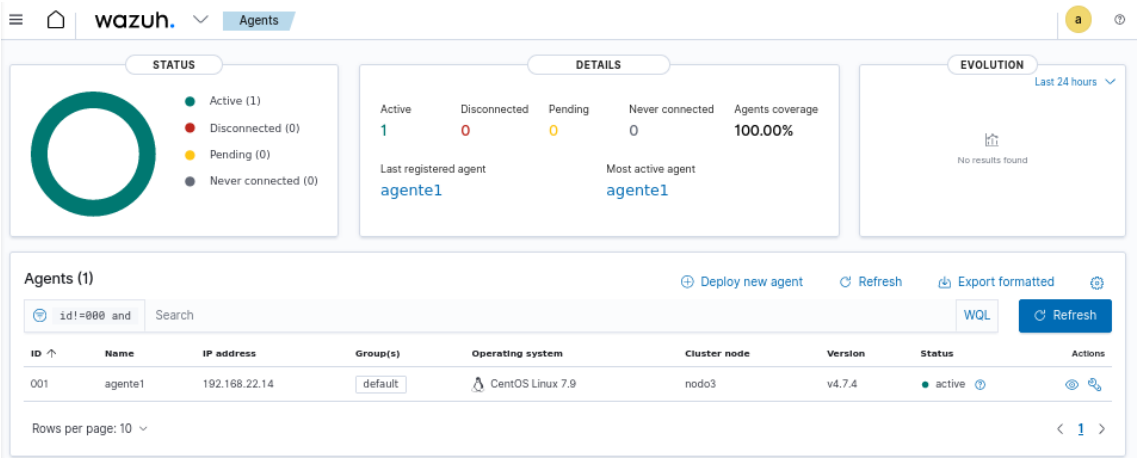


Imagen 88. *Lista de agentes en Wazuh*

Se abrirá la ventana llamada *"Deploy new agent"* donde habrá que rellenar ciertos parámetros [Imagen 89]:

- 1. Seleccionar el paquete para descargar e instalar en su sistema. Se seleccionará en *Linux* el paquete "*RPM amd64"*.
- 2. *Server* address. Se introducirá la dirección IP que el agente usa para comunicarse con el servidor, en este caso *192.168.22.11* que es la dirección del nodo *master* del *Wazuh server*.
- Configuraciones opcionales. Se asignará un nombre al agente, que en este caso será agente2 y se dejará la opción de grupo en "*Default"*.

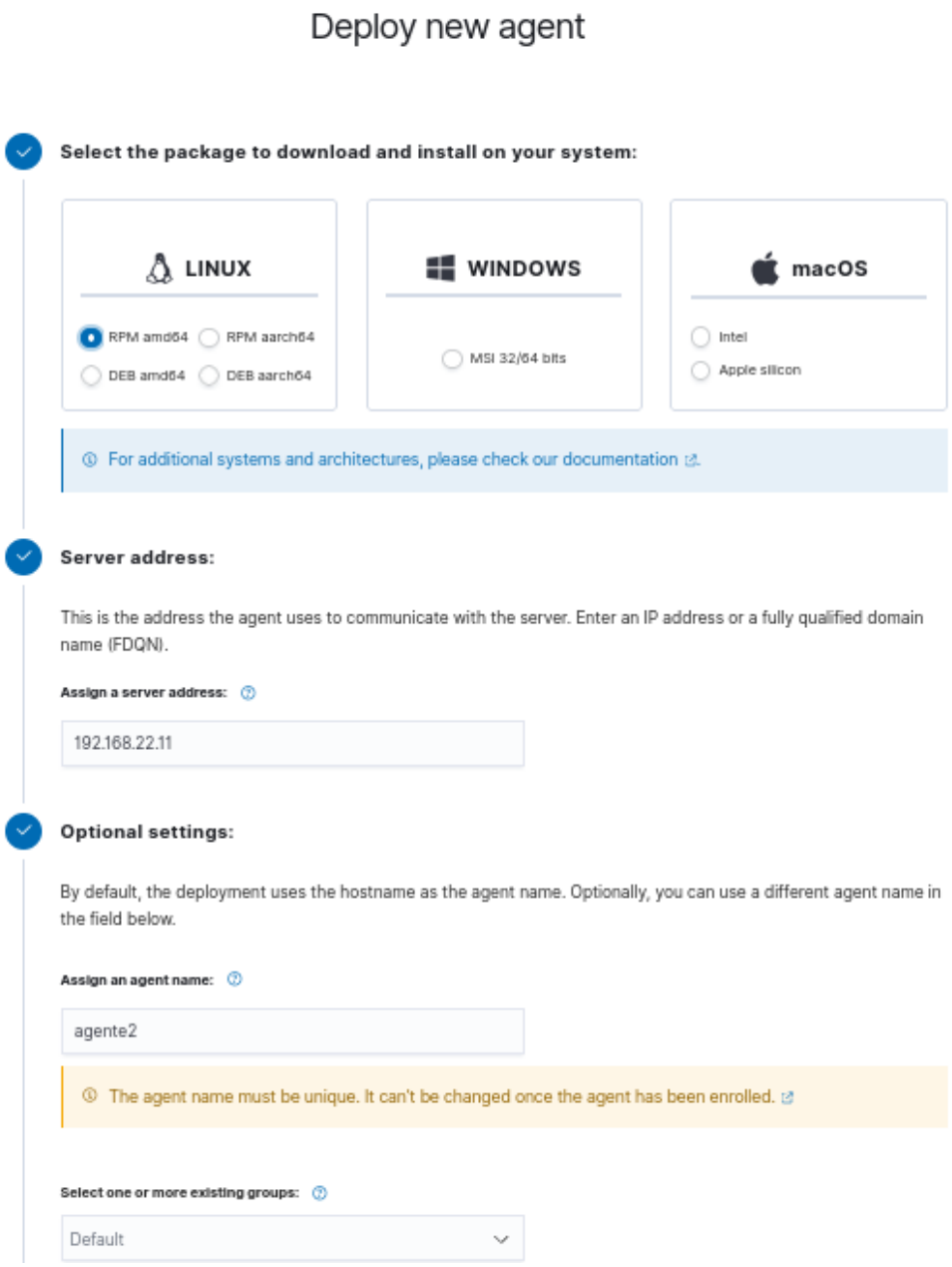


Imagen 89. *Deploy new agent parámetros agente2*

Editados los parámetros anteriores, se generará un comando [Imagen 90] que incluye datos añadidos y que se ejecutará en la máquina virtual donde se quiere instalar el nuevo agente, es decir, en la máquina virtual *Agente2*.

- `# curl -o Wazuh-agent-4.7.4-1.x86_64.rpm https://packages.Wazuh.com/4.x/yum/Wazuh-agent-4.7.4-1.x86_64.rpm && sudo WAZUH_MANAGER='192.168.22.11' WAZUH_AGENT_NAME='agente2' rpm -ihv Wazuh-agent-4.7.4-1.x86_64.rpm`

Run the following commands to download and install the agent:

```
curl -o wazuh-agent-4.7.4-1.x86_64.rpm https://packages.wazuh.com/4.x/yum/wazuh-agent-
4.7.4-1.x86_64.rpm && sudo WAZUH_MANAGER='192.168.22.11' WAZUH_AGENT_NAME='agente2' rpm -ihv
wazuh-agent-4.7.4-1.x86_64.rpm
```

Imagen 90. *Comando generado para instalar agente2*

Se editará el archivo de configuración ossec.conf [20] del agente cambiando la dirección IP del servidor a la del balanceador de carga de *Nginx* que es la *192.168.22.9* [Imagen 91].

- `# gedit /var/ossec/etc/ossec.conf &`

```
<server>
  <address>192.168.22.9</address>
  <port>1514</port>
  <protocol>tcp</protocol>
</server>
```

Imagen 91. *Archivo configuración Wazuh agent en agente2*

Una vez se haya instalado correctamente se habilitará y se iniciará el agente.

- `# systemctl daemon-reload`
- `# systemctl enable Wazuh-agent`
- `# systemctl start Wazuh-agent`

Para finalizar, de vuelta en la máquina virtual *Indexer*, en el panel de control de *Wazuh*, se hará *clic* en *"Close"* en la ventana que se encontraba aún abierta *"Deploy a new agent"*. Se abrirá la ventana de *"Agents"* del panel de control de *Wazuh*, donde se podrá verificar que se ha añadido correctamente un nuevo agente llamado *Agente2* a la lista de agentes en nuestro servidor de *Wazuh* [Imagen 91].

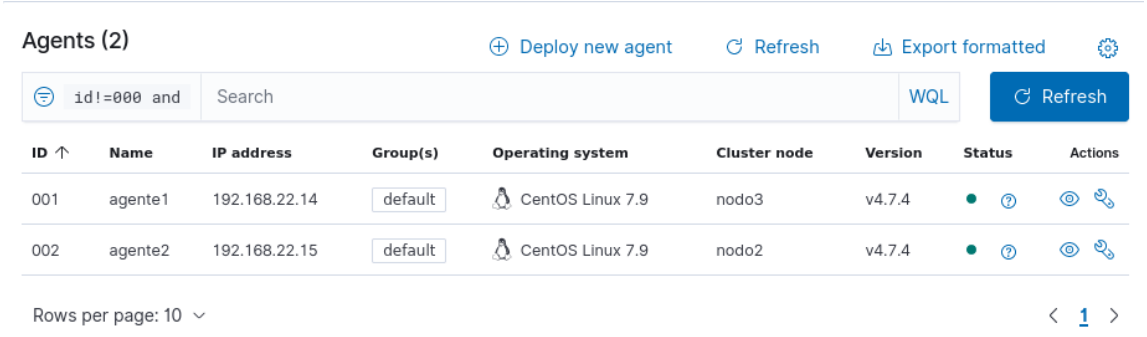


Imagen 92. *Lista de agentes agente1 y agente2 en Wazuh dashboard*

Para la configuración e instalación de los agentes Agente3 y Agente4 se seguirá el mismo procedimiento que se realizó para el Agente2. Las direcciones IP que se asignarán a los agentes Agente3 y Agente4 serán *192.168.22.16* y 192.168.22.17

respectivamente. También se les asignará el hostname agente3 y agente4 respectivamente. Una vez finalizada la configuración de red, firewall, *SElinux* y la instalación de *Wazuh* agent en ambos agentes, se verificará en el panel de control de *Wazuh* que aparecen listados los agentes agente3 y agente4 [Imagen 93].

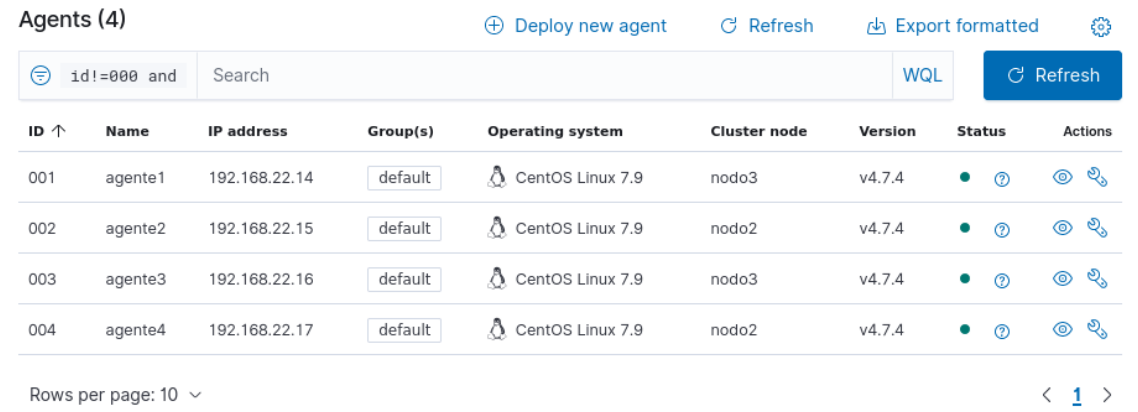


Imagen 93. *Lista de los 4 agentes en el servidor Wazuh*

Como se podrá observar el balanceador de carga ha distribuido automáticamente a los agentes conectando 2 agentes a uno de los nodos trabajadores y los otros 2 agentes al otro nodo, repartiendo así la carga entre los dos nodos trabajadores del *Wazuh server*.

7. Testeo del Clúster

A continuación, después de configurar el clúster de *Wazuh*, el balanceador de carga de *Nginx* y una vez se hayan configurado y conectado correctamente los agentes al clúster de *Wazuh* a través del balanceador de carga, se pasará a testear que el balanceador de carga funciona correctamente balanceando la carga de los agentes conectados a los nodos trabajadores y que se ofrece un servicio de alta disponibilidad en caso de que caiga uno de los nodos trabajadores.

Primero se pasará a detener uno de los nodos *worker* del clúster de *Wazuh*, en este caso el Nodo2. Una vez situados en el Nodo2 se parará el servicio de *Wazuh manager*.

- `# systemctl stop Wazuh-manager`

Se verificará en las máquinas virtuales Agente2 y Agente4 que contienen a los agentes agente2 y agente4 respectivamente, que no hay errores de conexión [Imagen 94][Imagen 95].

- `# cat /var/ossec/logs/ossec.log`

```
2024/05/25 17:31:58 wazuh-agentd: ERROR: (1137): Lost connection with manager. Setting lock.
2024/05/25 17:31:58 wazuh-agentd: INFO: Closing connection to server ([192.168.22.9]:1514/tcp).
2024/05/25 17:31:58 wazuh-agentd: INFO: Trying to connect to server ([192.168.22.9]:1514/tcp).
2024/05/25 17:31:58 wazuh-agentd: INFO: (4102): Connected to the server ([192.168.22.9]:1514/tcp).
2024/05/25 17:31:58 wazuh-agentd: INFO: Server responded. Releasing lock.
```

Imagen 94. *Resultado de los logs ossec.log en agente2*

```
2024/05/25 17:31:58 wazuh-agentd: ERROR: (1137): Lost connection with manager. Setting lock.
2024/05/25 17:31:58 wazuh-agentd: INFO: Closing connection to server ([192.168.22.9]:1514/tcp).
2024/05/25 17:31:58 wazuh-agentd: INFO: Trying to connect to server ([192.168.22.9]:1514/tcp).
2024/05/25 17:31:58 wazuh-agentd: INFO: (4102): Connected to the server ([192.168.22.9]:1514/tcp).
2024/05/25 17:31:58 wazuh-agentd: INFO: Server responded. Releasing lock.
```

Imagen 95. *Resultado de los logs ossec.log en agente4*

Se puede observar que los agentes perdieron conexión con el *Wazuh manager* a través del balanceador de carga, realizando una reconexión con el balanceador de carga que debería reconectarlos al nodo *worker* que se encuentre disponible.

Desde los agentes agente2 y agente4 se usará *netcat* verificar que la conectividad al balanceador de carga sigue siendo exitosa.

- `# nc -zv 192.168.22.9 1514`

Se ejecutará en todos los agentes un script mediante consola de comandos para generar tráfico con el balanceador de carga y comprobar a que nodo los redirige el balanceador de carga.

- `# while true; do echo "Prueba balanceo" | nc 192.168.22.9 1514; sleep 1; done`

Desde el balanceador de carga de *Nginx* se comprueban los logs de acceso [Imagen 96] para y error [Imagen] para verificar la redirección.

- `# # tail -f /var/log/nginx/access.log`
- `# tail -n 100 /var/log/nginx/error.log`

```
192.168.22.15 - - [25/May/2024:18:07:45 +0200] "HEAD / HTTP/1.1" 200 0 "-" "curl/7.29.0" "-"
192.168.22.17 - - [25/May/2024:18:07:50 +0200] "HEAD / HTTP/1.1" 200 0 "-" "curl/7.29.0" "-"
```

Imagen 96. *Resultado de los logs de access.log*

```
2024/05/25 18:04:47 [error] 7460#7460: *15 connect() failed (111: Connection refused) while connecting t
o upstream, client: 192.168.22.17, server: 0.0.0.0:1514, upstream: "192.168.22.12:1514", bytes from/to c
lient:0/0, bytes from/to upstream:0/0
2024/05/25 18:04:47 [warn] 7460#7460: *15 upstream server temporarily disabled while connecting to upstr
eam, client: 192.168.22.17, server: 0.0.0.0:1514, upstream: "192.168.22.12:1514", bytes from/to client:0
/0, bytes from/to upstream:0/0
2024/05/25 18:04:47 [error] 7460#7460: *17 connect() failed (111: Connection refused) while connecting t
o upstream, client: 192.168.22.15, server: 0.0.0.0:1514, upstream: "192.168.22.12:1514", bytes from/to c
lient:0/0, bytes from/to upstream:0/0
2024/05/25 18:04:47 [warn] 7460#7460: *17 upstream server temporarily disabled while connecting to upstr
eam, client: 192.168.22.15, server: 0.0.0.0:1514, upstream: "192.168.22.12:1514", bytes from/to client:0
/0, bytes from/to upstream:0/0
```

Imagen 97. *Resultado de los logs de error.log*

Se puede observar se reflejan las redirecciones en los logs "*access.log*" y los errores de conexión con el nodo2 en los logs de "*error.log*".

Se situará en la máquina virtual *Indexer*, en el apartado "Agents" del *Wazuh dashboard* se verificará que los agentes agente2 y agente4 se han conectado al nodo *worker* nodo3 [Imagen 98].

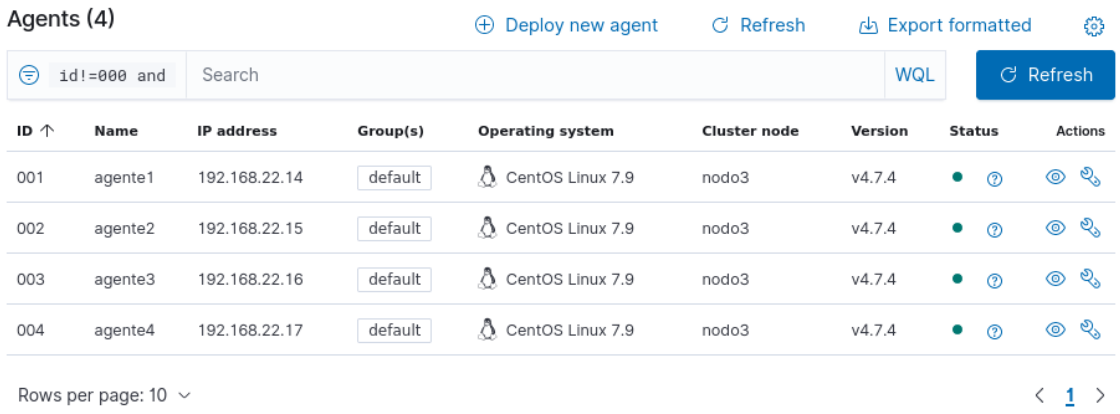


Imagen 98. *Agentes en Wazuh dashboard conectados a nodo3 comprobación 1*

De esta manera se verifica que el balanceador de carga *Nginx* ha redirigido correctamente las conexiones de los agentes conectados al nodo2 y los ha conectado al nodo3 al detectar que el servicio de *Wazuh manager* del nodo2 ha sido parado, ofreciendo un servicio de alta disponibilidad.

Se volverá a iniciar el servicio de *Wazuh manager* en el nodo *worker* nodo2.

- `# systemctl start Wazuh-manager`

En el *Wazuh dashboard* se contempla que, aunque se haya vuelto a iniciar el nodo2, los agentes siguen manteniendo las conexiones con el nodo3 [Imagen 99].

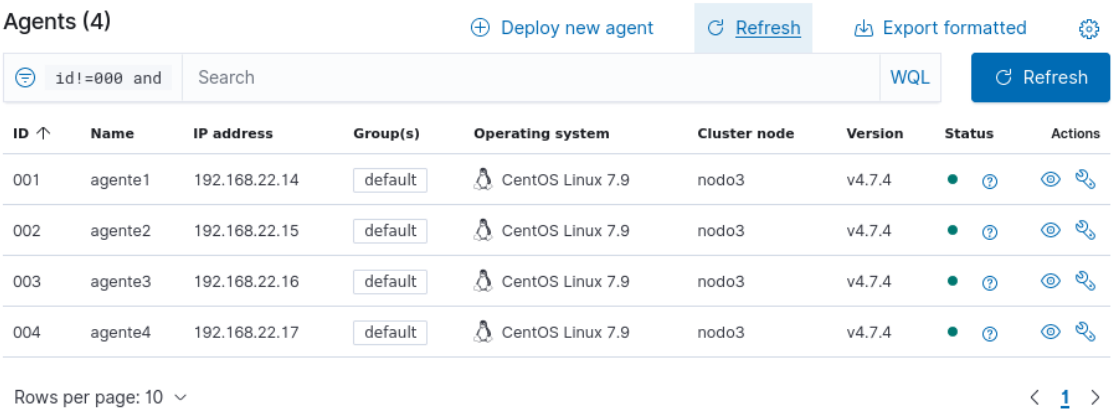


Imagen 99. *Agentes en Wazuh dashboard conectados a nodo3 comprobación 2*

Esto se debe a que los agentes de *Wazuh* mantienen una conexión persistente al nodo al que están conectados y esa conexión no se interrumpe a menos que el agente se reinicie o se pierda la conexión. Esto se debe al comportamiento general de la arquitectura cliente-servidor de *Wazuh* y su implementación de conexiones TCP.

Se comprobará que, aunque todos los agentes mantienen su conexión con el nodo3, el balanceador de carga de *Nginx* balancea las conexiones y el tráfico se equilibrará entre los nodos nodo2 y nodo3.

Para comprobarlo primero se generará tráfico con la herramienta netcat hacia el balanceador de carga desde los agentes. Para ello se usará un script en la consola de comandos que continuamente genere tráfico hacia el balanceador de carga mediante el puerto 1514.

- `# while true; do echo "Prueba balanceo" | nc 192.168.22.9 1514; sleep 1; done`

A continuación, en la máquina *Nginx*, se verificarán los logs para ver hacia donde se redirige el tráfico generado por los agentes y así comprobar si se balancean las conexiones entre los nodos trabajadores [Imagen 100].

- `# tail -f /var/log/nginx/tcp_access.log`

```
[root@nginx ~]# sudo tail -f /var/log/nginx/tcp_access.log
192.168.22.14 [31/May/2024:22:43:08 +0200] TCP 200 0 22 0.001 192.168.22.13:1514
192.168.22.16 [31/May/2024:22:43:08 +0200] TCP 200 0 22 0.000 192.168.22.12:1514
192.168.22.17 [31/May/2024:22:43:09 +0200] TCP 200 0 22 0.001 192.168.22.13:1514
192.168.22.15 [31/May/2024:22:43:09 +0200] TCP 200 0 22 0.000 192.168.22.12:1514
192.168.22.14 [31/May/2024:22:43:09 +0200] TCP 200 0 22 0.000 192.168.22.13:1514
192.168.22.16 [31/May/2024:22:43:09 +0200] TCP 200 0 22 0.000 192.168.22.12:1514
192.168.22.17 [31/May/2024:22:43:10 +0200] TCP 200 0 22 0.001 192.168.22.13:1514
192.168.22.15 [31/May/2024:22:43:10 +0200] TCP 200 0 22 0.001 192.168.22.12:1514
192.168.22.14 [31/May/2024:22:43:10 +0200] TCP 200 0 22 0.000 192.168.22.13:1514
192.168.22.16 [31/May/2024:22:43:10 +0200] TCP 200 0 22 0.001 192.168.22.12:1514
192.168.22.17 [31/May/2024:22:43:11 +0200] TCP 200 0 22 0.000 192.168.22.13:1514
192.168.22.15 [31/May/2024:22:43:11 +0200] TCP 200 0 22 0.001 192.168.22.12:1514
192.168.22.14 [31/May/2024:22:43:11 +0200] TCP 200 0 22 0.000 192.168.22.13:1514
192.168.22.16 [31/May/2024:22:43:11 +0200] TCP 200 0 22 0.000 192.168.22.12:1514
192.168.22.17 [31/May/2024:22:43:12 +0200] TCP 200 0 22 0.000 192.168.22.13:1514
192.168.22.15 [31/May/2024:22:43:12 +0200] TCP 200 0 22 0.000 192.168.22.12:1514
192.168.22.16 [31/May/2024:22:43:12 +0200] TCP 200 0 22 0.001 192.168.22.13:1514
192.168.22.14 [31/May/2024:22:43:12 +0200] TCP 200 0 22 0.000 192.168.22.12:1514
192.168.22.17 [31/May/2024:22:43:13 +0200] TCP 200 0 22 0.000 192.168.22.13:1514
192.168.22.15 [31/May/2024:22:43:13 +0200] TCP 200 0 22 0.000 192.168.22.12:1514
192.168.22.16 [31/May/2024:22:43:13 +0200] TCP 200 0 22 0.000 192.168.22.13:1514
192.168.22.14 [31/May/2024:22:43:13 +0200] TCP 200 0 22 0.000 192.168.22.12:1514
192.168.22.17 [31/May/2024:22:43:14 +0200] TCP 200 0 22 0.001 192.168.22.13:1514
192.168.22.15 [31/May/2024:22:43:14 +0200] TCP 200 0 22 0.000 192.168.22.12:1514
192.168.22.16 [31/May/2024:22:43:14 +0200] TCP 200 0 22 0.000 192.168.22.13:1514
```

Imagen 100. *Resultado de los logs de tcp_access.log*

Se puede observar que efectivamente se balancea la carga a ambos nodos. Ahora se pasará a reiniciar todos los agentes para que vuelvan a reconectarse y se balanceen las conexiones persistentes a ambos nodos.

- `# systemctl restart Wazuh-agent`

Se podrá observar en *Wazuh dashboard* en la máquina virtual *indexer*, que efectivamente se han actualizado las conexiones persistentes de los agentes a los nodos, siendo balanceadas por el balanceador de carga [Imagen 101].

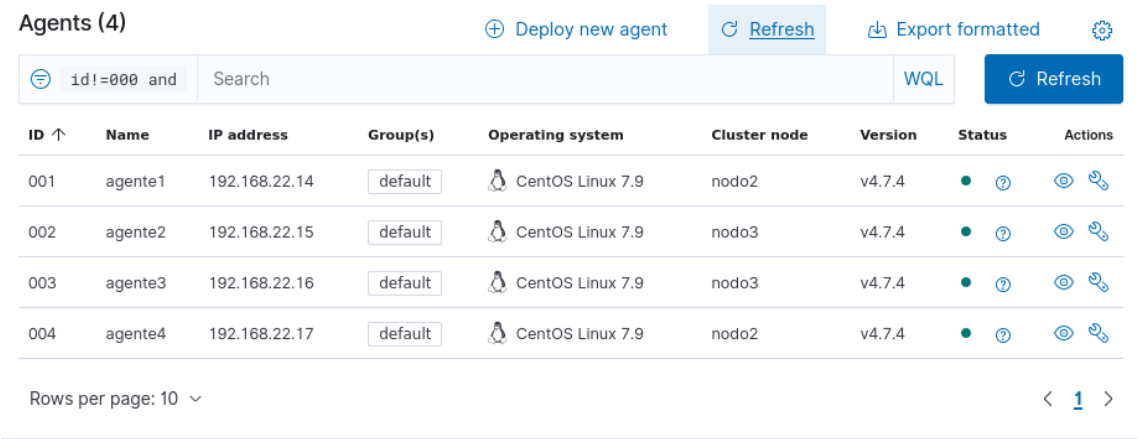


Imagen 101. *Agentes en Wazuh dashboard conectados a ambos nodos*

Para finalizar, se ha verificado que el balanceador de carga distribuye correctamente las conexiones entre ambos nodos y que se ofrece un servicio de

alta disponibilidad con la caída de un nodo. También se llega a la conclusión de que un agente mantiene las conexiones con un nodo de manera persistente, por lo que ante una caída de uno de los nodos los agentes que pierden la conexión con este último, pasarán a conectarse al que se mantiene activo manteniendo una conexión con el nodo que queda activo de manera persistente, incluso después de que el nodo caído vuelva a estar activo, hasta que se reinicie algún agente y el balanceador se encargue de conectarlo al otro nodo para balancear la carga.

8. Máquina virtual atacante con *Kali Linux*

Se creará una máquina virtual con sistema operativo *Kali Linux*, máquina desde la cual se realizarán los ataques a los agentes conectados al clúster de *Wazuh* y monitorizados por el mismo.

8.1 Descarga de una imagen *ISO* de *Kali Linux*

Para la descarga de una imagen de disco *ISO* del sistema operativo *Kali Linux* 2024.1, se accederá al navegador de internet y se introducirá la siguiente URL [23] "*https://www.kali.org/get-kali/#kali-installer-images*" donde se mostrará la siguiente página web [Imagen 102].

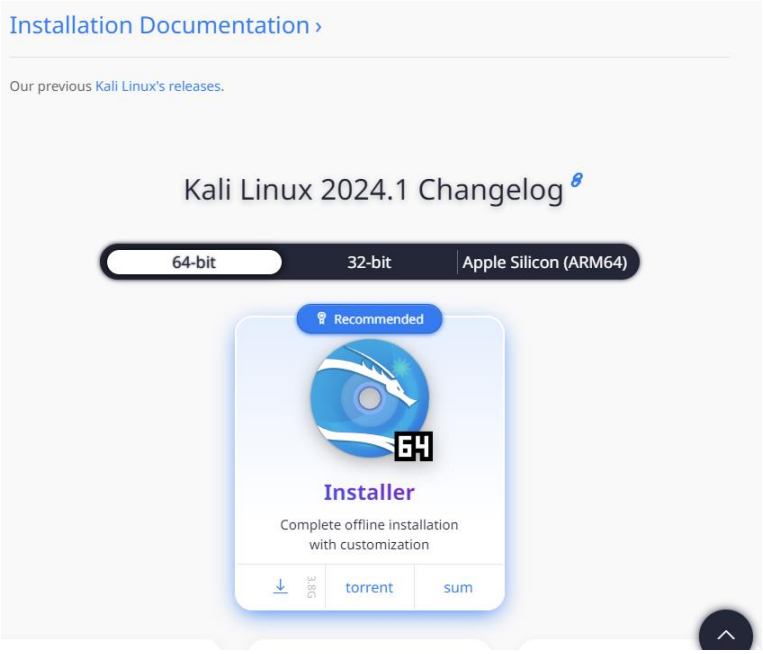


Imagen 102. *Sitio web donde se descarga la Imagen ISO de Kali Linux*

Se hará clic sobre el cuadrado donde pone "*Installer*" y se descargará la imagen *ISO* del sistema operativo.

8.2 Creación de la MV

Para la creación de la máquina virtual se utilizará el programa de virtualización *VirtualBox* igual que para la creación de una máquina virtual en CentOS 7 en otros puntos. Una vez ejecutado el programa *VirtualBox* se seleccionará la opción "Nueva" [Imagen 103].

Imagen 103. *Barra de navegación VirtualBox*

A continuación, se abrirá una ventana llamada "*Crear máquina virtual*" donde se introducirán los datos de Nombre, la carpeta donde se guardará la máquina virtual en el sistema, la imagen *ISO* previamente descargada de *Kali Linux* 2024.1 [Imagen 104]. Esta vez el *SO* detectado no puede ser instalado de forma desatendida. Una vez introducidos todos los campos se hará *clic* en el botón "*Siguiente*".

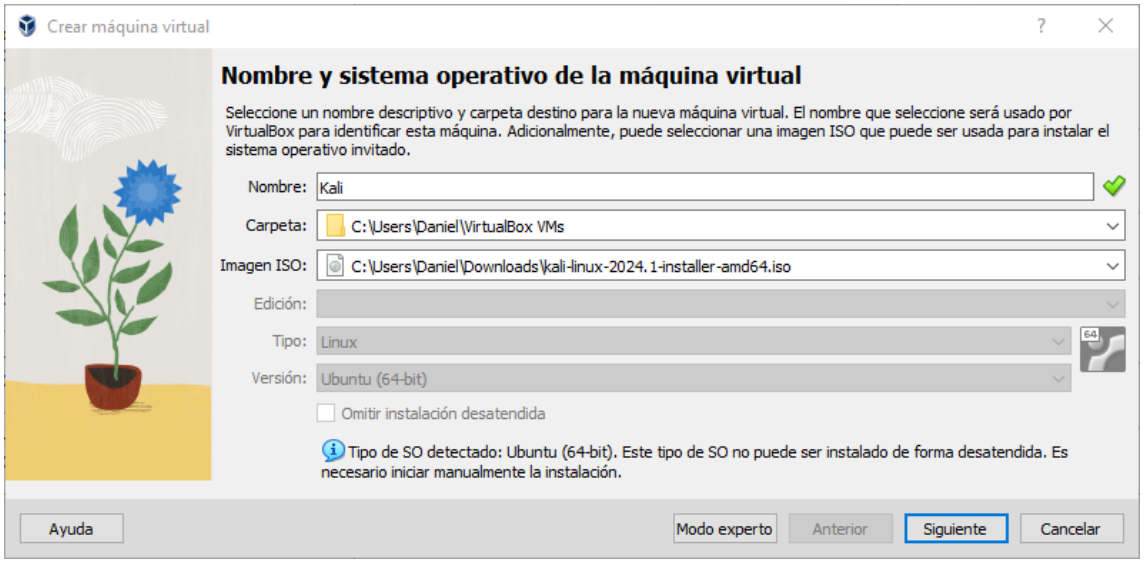


Imagen 104. *Ventana Crear máquina virtual Kali*

Se pasará a modificar el hardware de la máquina virtual cambiando la cantidad de memoria RAM y número de CPU virtuales [Imagen 105]. Para la memoria base RAM se añadirán 2048 MB y para el número de procesadores se añadirán 2 CPU. Una vez editados estos parámetros se hará *clic* en el botón "*Siguiente*".

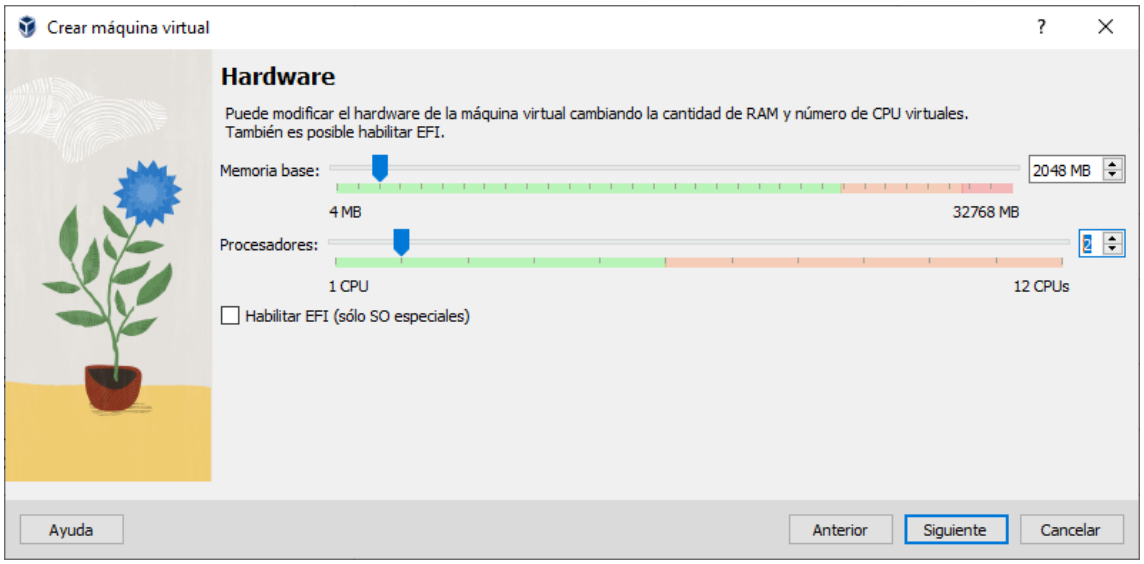


Imagen 105. *Ventana Crear máquina virtual Kali, Hardware*

Para el disco duro virtual se seleccionará la opción "Crear un disco duro virtual ahora" y se añadirán 50 GB para el tamaño del disco [Imagen 106]. Una vez introducidos estos parámetros se hará *clic* en la opción "*Siguiente*".

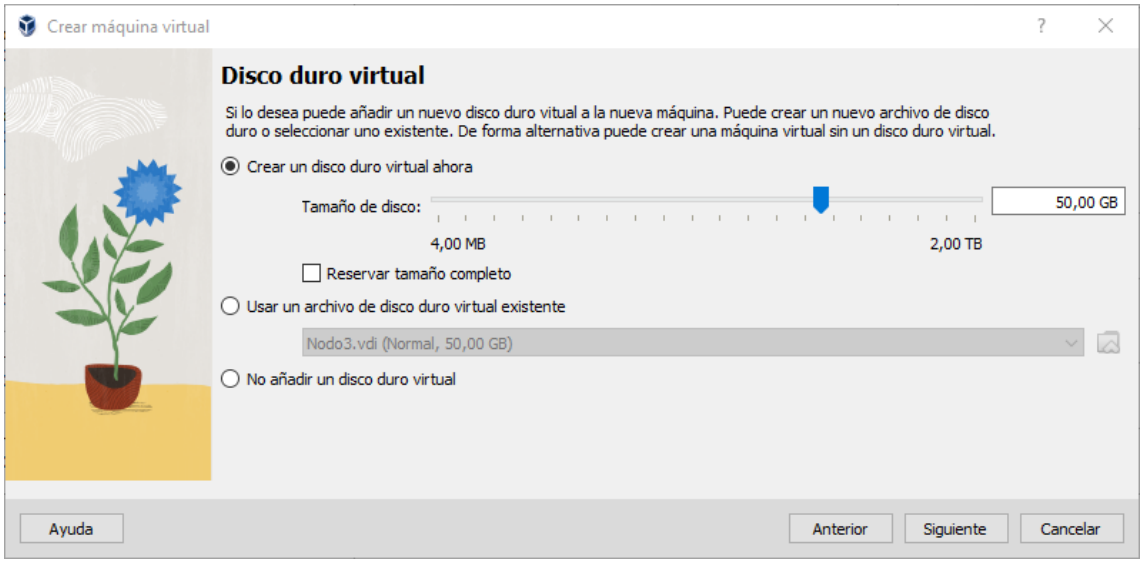


Imagen 106. *Ventana Crear máquina virtual Kali, Disco duro virtual*

Una vez completados todos los pasos anteriores aparecerá un resumen de los parámetros introducidos para la creación de nuestra nueva máquina virtual [Imagen 107] que se revisarán para ver si los datos introducidos son los correctos y una vez verificado esto se hará *clic* en el botón "*Terminar*".

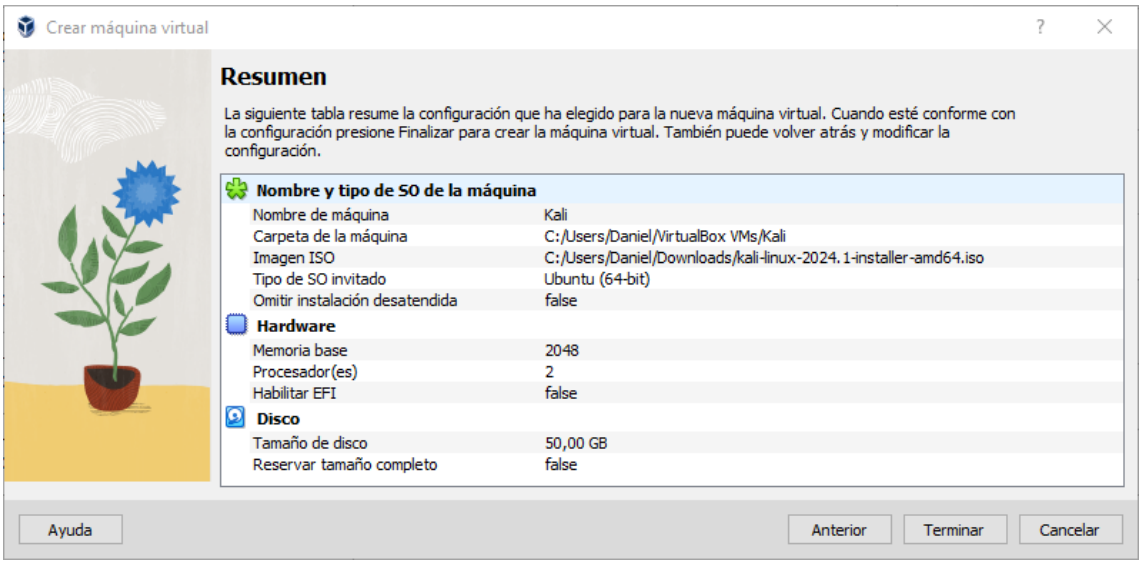


Imagen 107. *Ventana Crear máquina virtual Kali, Resumen*

Se dispondrá de una nueva máquina virtual con el nombre que se le haya elegido para su creación en el menú de inicio de *VirtualBox*, en este caso *Kali* [Imagen 108]. Para iniciar la nueva máquina virtual se hará doble *clic* sobre ella. Una vez arrancada la máquina virtual se ejecutará el disco *ISO* añadido previamente con la instalación del *SO*.

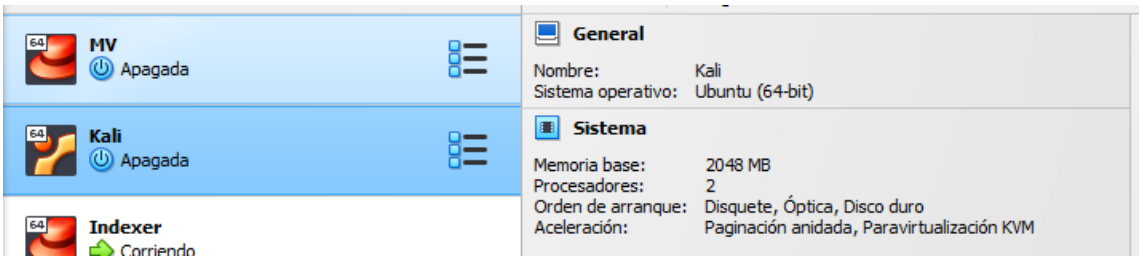


Imagen 108. *Máquina Kali creada en VirtualBox*

Una vez iniciada la máquina virtual *Kali*, aparecerá un menú llamado *"Kali Linux installer menú (BIOS mode)"*. Se moverá con las flechas y se seleccionará pulsando *Enter* sobre la opción *"Install"* [Imagen 109].

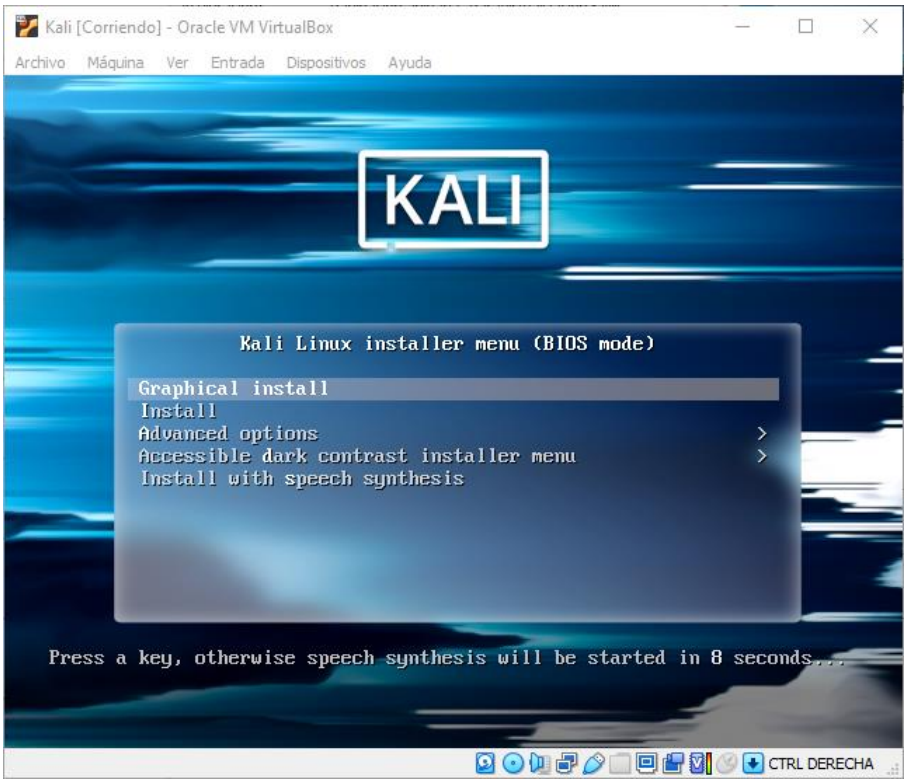


Imagen 109. *Instalación Kali Linux (BIOS mode)*

Se abrirá un menú de selección de idioma llamado *"[!!] Select a language"* donde se seleccionará un idioma con el que se realizará la instalación, en este caso se seleccionará la opción de *"Spanish"* o *"Español"* [Imagen 110].

Imagen 110. *Instalación Kali, select a language*

Luego se pasará a una ventana llamada *"[!!] Seleccione su ubicación"* donde se seleccionará la ubicación *"España"* [Imagen 111].

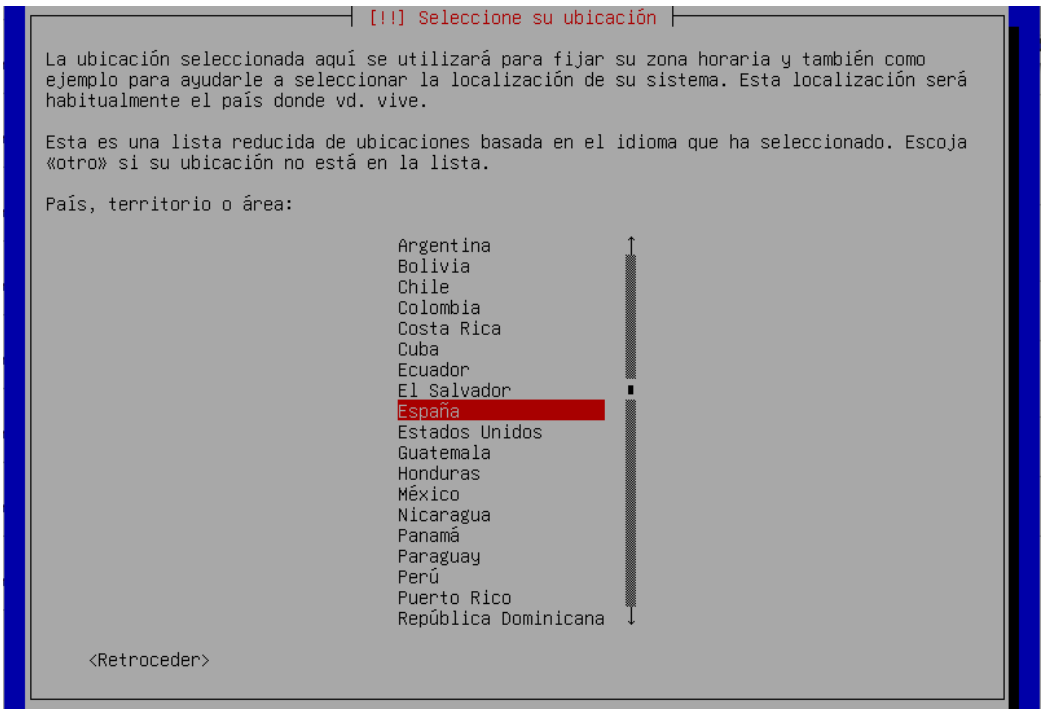


Imagen 111. *Instalación Kali, seleccione su ubicación*

Se abrirá otra ventana llamada *"[!!] Configure el teclado"* donde se seleccionará el mapa de teclado a usar, en este caso *"Español"* [Imagen 112].

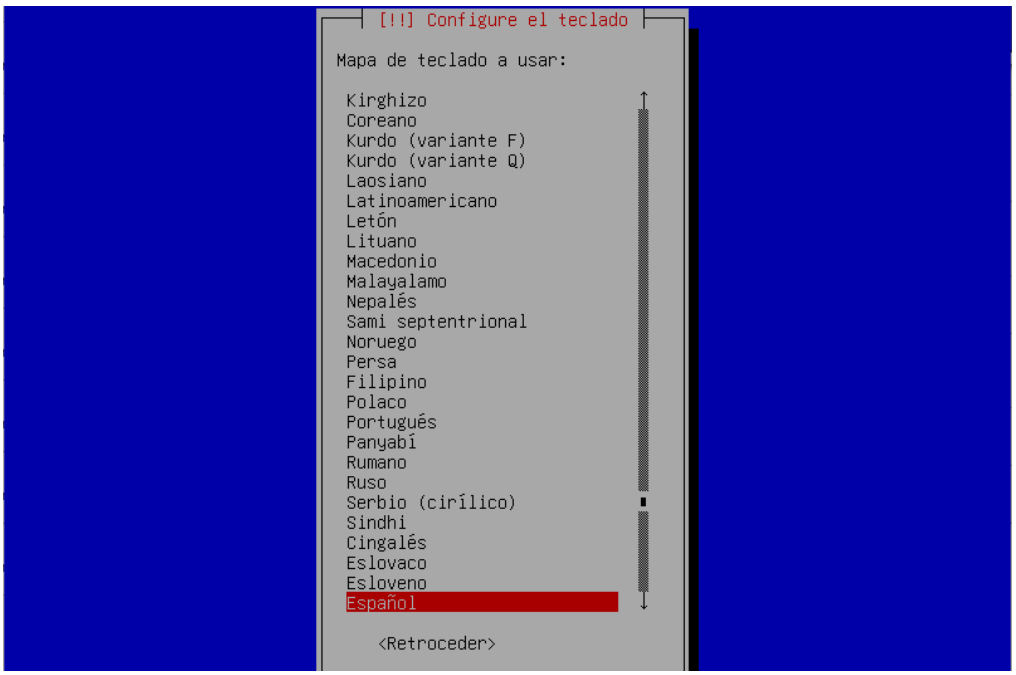


Imagen 112. *Instalación Kali, configure el teclado*

Una vez se completen los pasos anteriores se abrirá una ventana donde empezará el proceso de instalación del sistema operativo *Kali Linux* [Imagen 113].

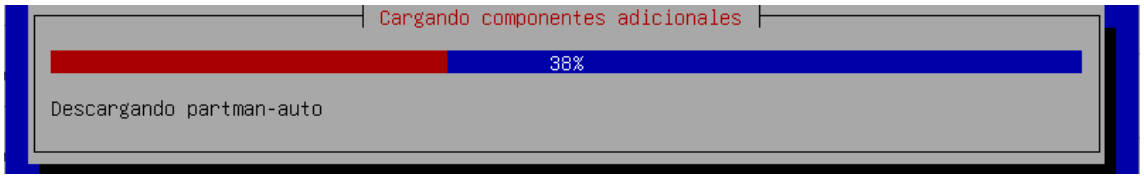


Imagen 113. *Instalación Kali, cargando componentes adicionales*

Se abrirá una ventana llamada "[!] Configurar la red" donde se introducirá el nombre de la máquina, se usará el nombre proporcionado por defecto y se presionará *Enter* en "Continuar" [Imagen 114].

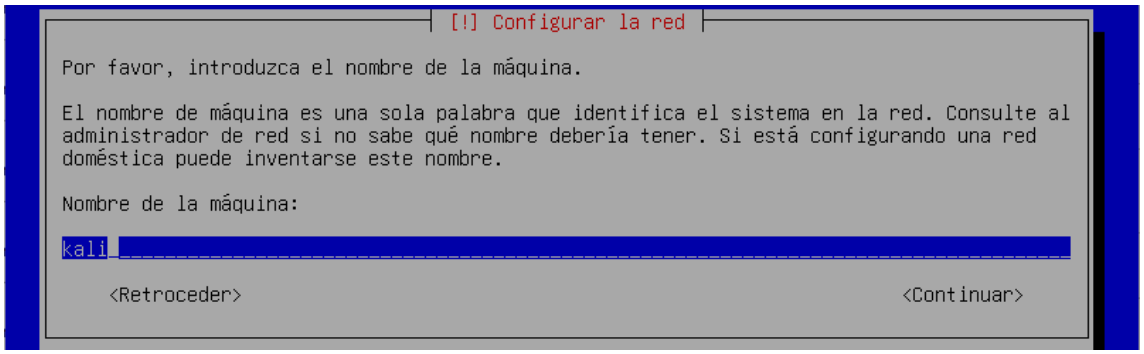


Imagen 114. *Instalación Kali, configurar la red*

Se pedirá a continuación el nombre de dominio que se dejará vacío y se presionará *Enter* en "Continuar" [Imagen 115].

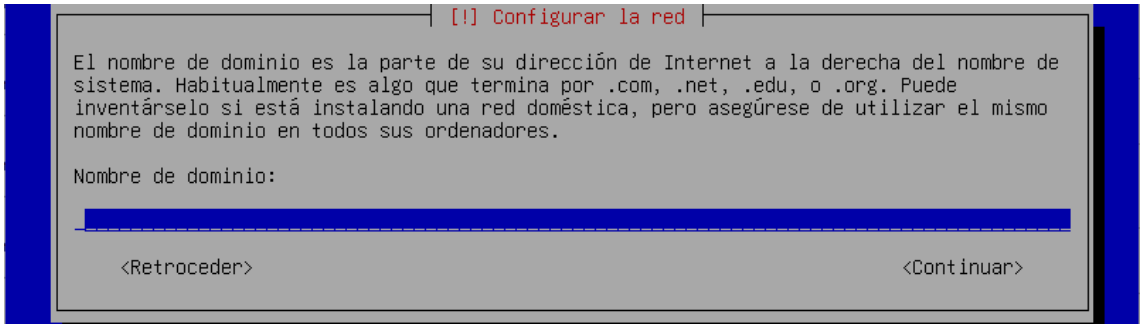


Imagen 115. *Instalación Kali, configurar la red*

Aparecerá una ventana llamada "[!!] Configurar usuarios y contraseñas" donde se creará un nombre de usuario nuevo. Se escribirá el nombre "*daniel*" y se presionará *Enter* en "Continuar" [Imagen 116].

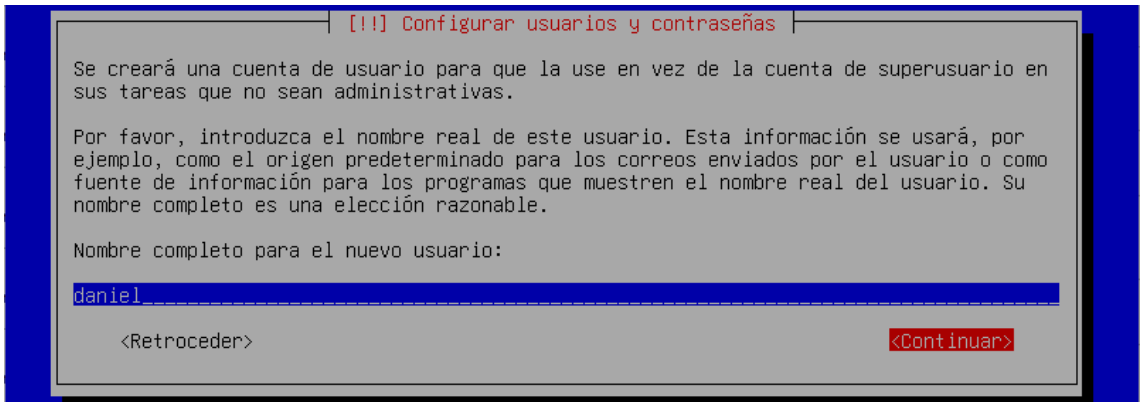


Imagen 116. *Instalación Kali, configurar usuarios y contraseñas*

Una vez elegido el nombre de usuario se pedirá una contraseña. Introducida la nueva contraseña para el usuario creado antes, se presionará *Enter* en *"Continuar"*. Aparecerá otra ventana para verificar la contraseña introducida y se presionará *Enter* en "Continuar" [Imagen 117].

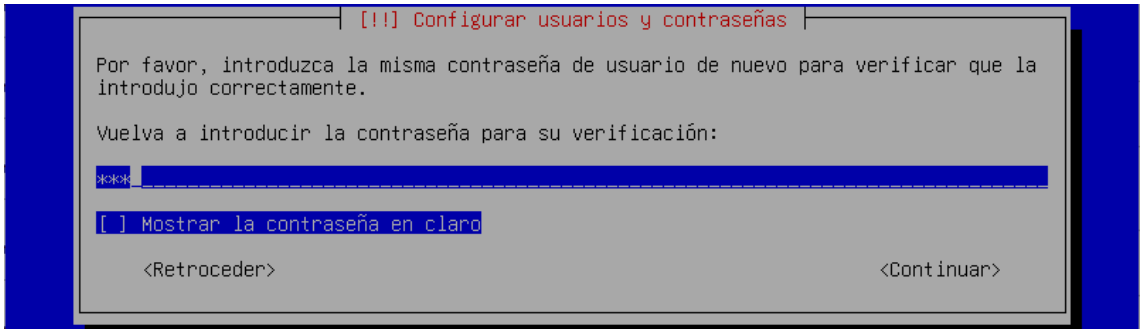


Imagen 117. *Instalación Kali, configurar usuarios y contraseñas verificar*

Se configurará la zona horaria en la ventana llamada *"[!!] Configurar el reloj"* donde se seleccionará la opción "Península" y se presionará *Enter* [Imagen 118].

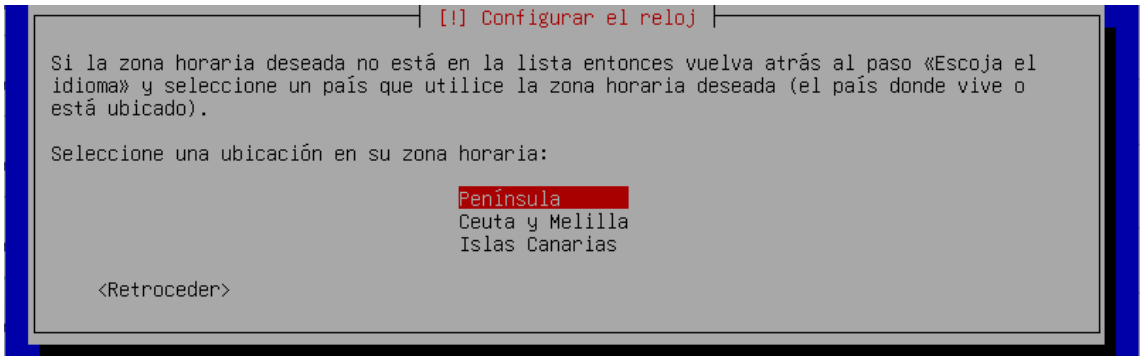


Imagen 118. *Instalación Kali, configurar el reloj*

En la ventana *"[!!] Particionado de discos"* se presionará *Enter* en la opción *"Guiado – utilizar todo el disco"* [Imagen 119].

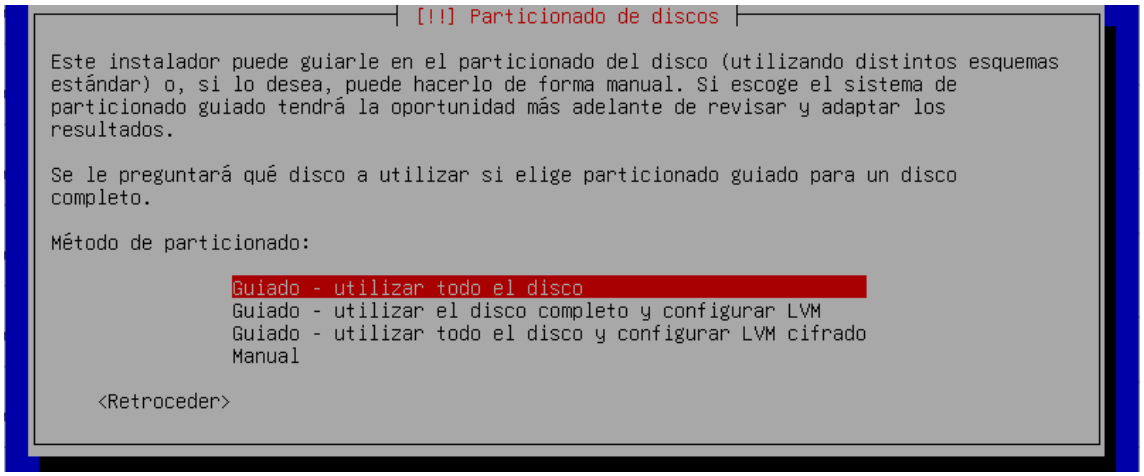


Imagen 119. *Instalación Kali, particionado de discos*

Y se presionará *Enter* en el disco que aparece a continuación [Imagen 120].

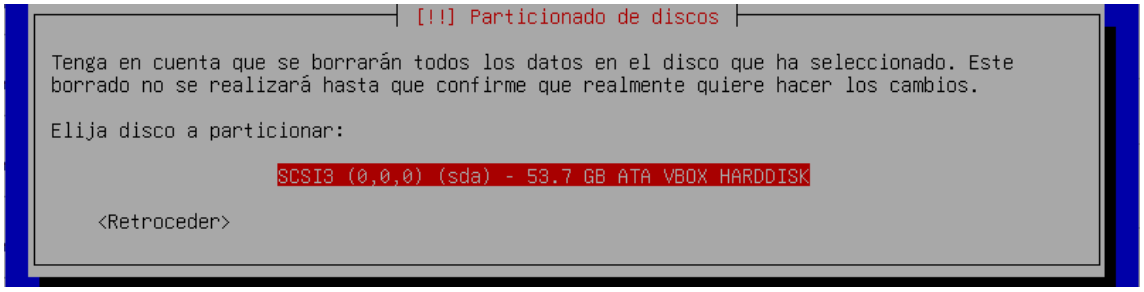


Imagen 120. *Instalación Kali, particionado de discos, elegir disco*

Se seleccionará la opción para el particionado “Todos los ficheros en una partición” [Imagen 121].

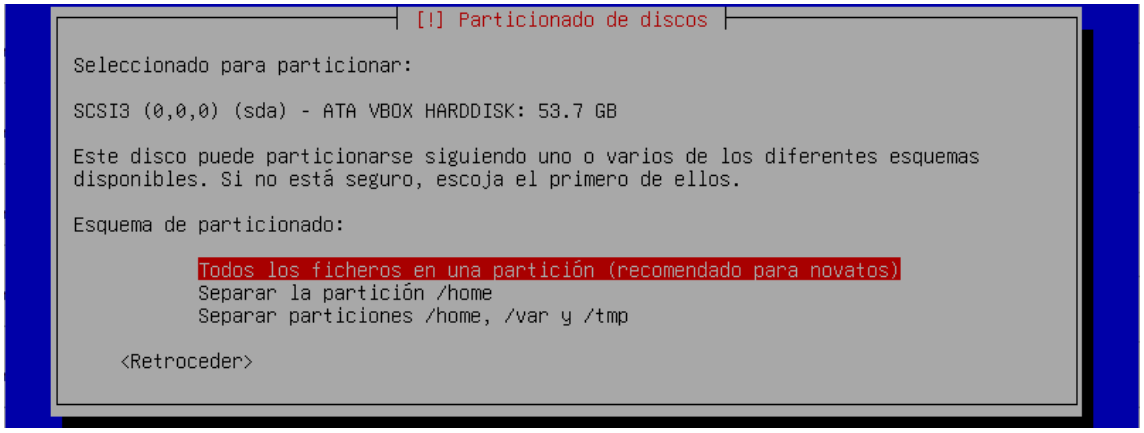


Imagen 121. *Instalación Kali, particionado de discos, elegir particionado*

Para finalizar, se presionará *Enter* en la opción *“Finalizar el particionado y escribir los cambios en el disco”* [Imagen 122] y se aceptará la escritura de cambios en los discos.

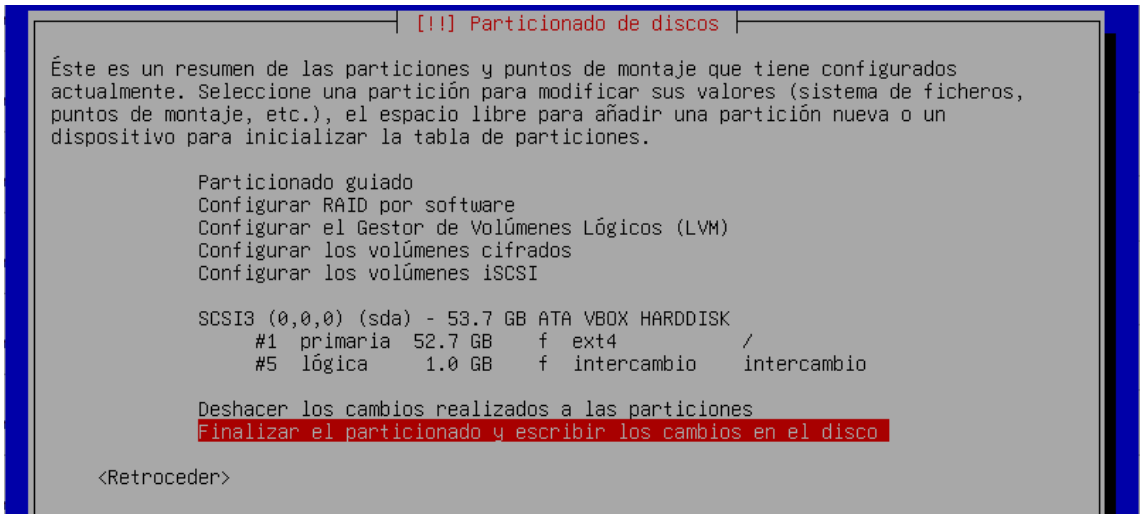


Imagen 122. *Instalación Kali, finalizar particionado*

Finalizada la configuración de particionado de discos empezará la instalación del sistema operativo [Imagen 123].

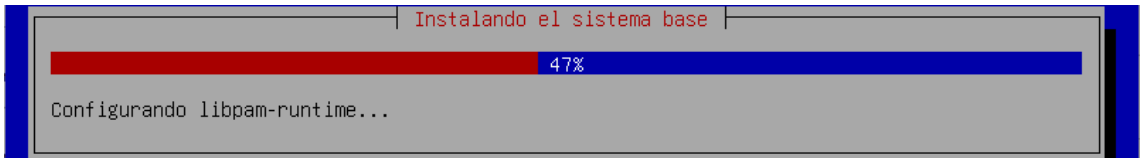


Imagen 123. *Instalación Kali, instalando sistema base*

Terminada la instalación, aparecerá una ventana con el mensaje de "Instalación completada", mostrando que la instalación ha finalizado con éxito. Se presionará *Enter* en *"Continuar"* para reiniciar.

8.3 Configuración de la máquina virtual *Kali*

Una vez reiniciada la máquina virtual después de su instalación, se accederá a ella con las credenciales proporcionadas en el proceso de instalación del sistema operativo.

8.3.1 Actualización del sistema *Kali Linux* 2024.1

Kali Linux es una distribución popular entre los profesionales de la ciberseguridad y los aficionados a la tecnología. Mantener un sistema *Kali* al día es fundamental para asegurar tanto la seguridad como el rendimiento óptimo del sistema [24].

Antes de proceder con la actualización, se actualizará la lista de repositorios [Imagen 124].

- `# sudo apt update`

```
┌──(daniel㉿kali)-[~]
└─$ sudo apt update
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.9 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [47.0 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [114 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [257 kB]
Fetched 67.3 MB in 8s (8423 kB/s)
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
771 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Imagen 124. *Comando apt update*

Una vez actualizada la lista de repositorios, puedes proceder con la actualización del sistema:

- `# sudo apt upgrade`

8.3.2 Configuración de red

Actualizada la máquina virtual, se pasará a realizar la configuración de red asignando una dirección IP a la nueva máquina de modo que se encuentre en la misma red en la que están el clúster de *Wazuh* y los agentes conectados a él.

Primero se añadirá la máquina virtual a la red Nat creada anteriormente. Para ello, en el inicio de *VirtualBox* se hará *clic* en nuestra máquina creada y de nuevo se hará *clic* en "*Configuración*" [Imagen 125].

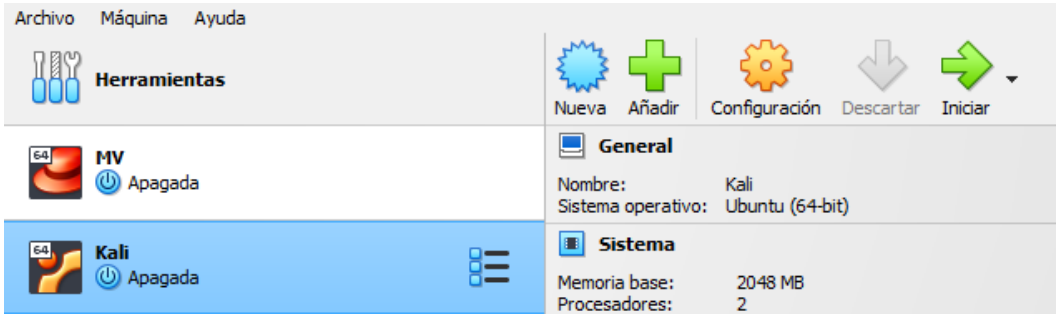


Imagen 125. *VirtualBox selección máquina Kali*

A continuación, se abrirá una ventana de configuración de la máquina virtual. En ella se irá al apartado de "*Red*" y en el "Adaptador 1" se editará la configuración cambiando el tipo de red seleccionando "*Red NAT*" y se elegirá la red "*Red Nat*" creada anteriormente [Imagen 126].

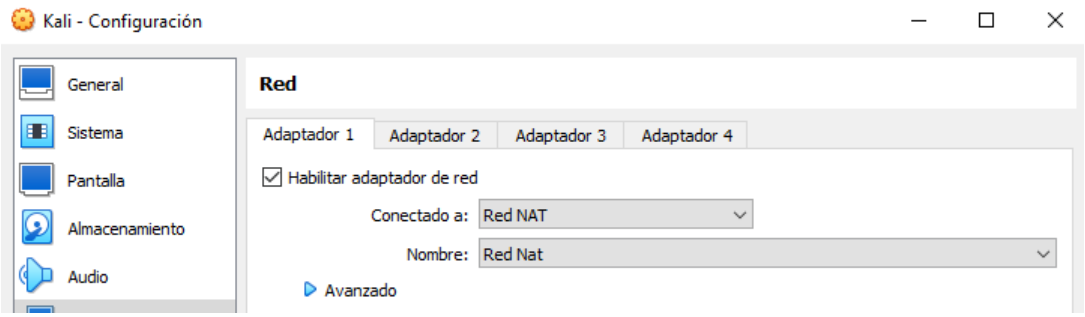


Imagen 126. *Configuración red Kali*

Una vez cambiados los parámetros, se hará *clic* en el botón "*Aceptar*" para guardar los cambios y se iniciará la máquina virtual. Iniciada la máquina virtual se escribirá en la consola el siguiente comando para ver la configuración del adaptador de red [Imagen 127].

- `# ifconfig`

```
┌──(daniel㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.22.122.6  netmask 255.255.255.0  broadcast 10.22.122.255
        inet6 fe80::a00:27ff:fe8a:2760  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:8a:27:60  txqueuelen 1000  (Ethernet)
        RX packets 3  bytes 1770 (1.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 3426 (3.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Imagen 127. *Comando ifconfig máquina Kali*

Como se puede observar, la dirección IP del adaptador de red es 10.22.122.6. Se tirará abajo la interfaz eth0 y se creará la interfaz eth1.

Se editará el archivo de configuración de red que se accederá con el siguiente comando.

- `# sudo nano /etc/network/interfaces`

Donde se añadirá la siguiente configuración [Tabla 8]:

Tabla 8. *Configuración eth1 en máquina virtual Kali*

```
auto eth1
iface eth1 inet static
    address 192.168.22.6
    netmask 255.255.255.0
    network 192.168.22.0
    broadcast 192.168.22.255
    gateway 192.168.22.1
```

- auto eth1: siendo eth1 la interfaz de red utilizada por la máquina virtual

- *iface eth0 inet static*: indicando que la configuración de red de la interfaz de red eth0 se configurará de manera estática.
- *address*: donde se definirá la IP de la máquina virtual que será *192.168.22.6*
- *netmask*: se define la máscara de red
- *network*: la red donde se conectará la máquina, 192.168.22.0
- *gateway*: será la puerta de enlace 192.168.22.1

Se reiniciará la interfaz de red para guardar y aplicar los cambios.

- `# sudo systemctl restart networking`

Para verificar que todo esté configurado correctamente y que se dispone de conexión hacia internet se ejecutará el siguiente comando:

- `# ping google.es`

9. Fase de Enumeración

Esta fase de enumeración es usada en auditorías de ciberseguridad para analizar equipos dentro de una red interna.

En este apartado, una vez conectada la máquina virtual a la red virtual donde se encuentra el clúster de *Wazuh* y los agentes, se pasará a identificar los equipos que hay dentro de la red virtual "*Red Nat*". El usuario de la máquina virtual *Kali* no conoce las direcciones IP ni los equipos que existen dentro de esa red, simplemente se ha conectado en esa red.

Desde la máquina virtual *Kali* con la ayuda de las herramientas *Nmap* y *Metasploit*, las cuales ya vienen instaladas en *Kali Linux*, se ejecutarán comandos para analizar la red y así identificar equipos existentes en esa red, así como los servicios que ofrecen, versiones y posibles vulnerabilidades que puedan presentar.

9.1 Identificación de la red 192.168.22.0

Una vez conectada la máquina virtual *Kali* a la red 192.168.22.0, con la herramienta *Nmap* se procederá a identificar todos los dispositivos que se hayan conectados en esa red, así como de los servicios que éstos ofrecen [Imagen 128] [Imagen 129]. Con la opción "-sS" se realiza un escaneo SYN, el cuál es más sigiloso, ya que no llega a completar las conexiones TCP. La opción "-Pn" se ejecuta para no realizar ping, mediante el cual el protocolo ICMP envía al host un determinado datagrama para solicitar una respuesta. La opción "-oA" interna se usa para guardar los resultados en un archivo en varios formatos admitidos. Las opciones usadas están recogidas en el manual de *Nmap*.

- `# sudo nmap -sS -Pn -F -oA interna 192.168.22.0-254`

```
┌──(daniel㉿kali)-[~]
└─$ sudo nmap -sS -Pn -F -oA interna 192.168.22.0-254
[sudo] password for daniel:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-01 17:09 CEST
Nmap scan report for 192.168.22.1
Host is up (0.00022s latency).
Not shown: 99 closed tcp ports (reset)
PORT   STATE    SERVICE
53/tcp filtered domain
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.22.2
Host is up (0.00083s latency).
Not shown: 98 filtered tcp ports (no-response)
PORT    STATE SERVICE
135/tcp open  msrpc
445/tcp open  microsoft-ds
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.22.9
Host is up (0.00074s latency).
Not shown: 90 filtered tcp ports (no-response), 9 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:19:05:07 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.22.10
Host is up (0.00067s latency).
Not shown: 90 filtered tcp ports (no-response), 9 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:84:CA:B2 (Oracle VirtualBox virtual NIC)
```

Imagen 128. *Comando Nmap escaneo*

```
Nmap scan report for 192.168.22.11
Host is up (0.00062s latency).
Not shown: 90 filtered tcp ports (no-response), 9 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:78:8A:54 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.22.12
Host is up (0.00099s latency).
Not shown: 90 filtered tcp ports (no-response), 9 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:56:82:01 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.22.13
Host is up (0.0010s latency).
Not shown: 90 filtered tcp ports (no-response), 9 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:B7:22:20 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.22.14
Host is up (0.0024s latency).
Not shown: 90 filtered tcp ports (no-response), 9 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:9B:6E:E4 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.22.15
Host is up (0.0019s latency).
Not shown: 90 filtered tcp ports (no-response), 9 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:F5:31:C8 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.22.16
Host is up (0.00057s latency).
Not shown: 90 filtered tcp ports (no-response), 9 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:41:71:DC (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.22.17
Host is up (0.00065s latency).
Not shown: 90 filtered tcp ports (no-response), 9 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:66:A7:FE (Oracle VirtualBox virtual NIC)
```

Imagen 129. *Comado Nmap escaneo 2*

Como se puede observar todos los equipos aparecen con los puertos filtrados. Se volverá a escanear la red, pero esta vez con evasión de cortafuegos con la opción "-f".

- `# sudo nmap -sS -Pn -F -f -oA interna2 192.168.22.0-254`

En cambio, se siguen obteniendo los mismos resultados. Los puertos salen como filtrados, es decir, que no se pueden alcanzar debido a las reglas firewall que bloquean el tráfico.

A continuación, se pasará a escanear cada uno de los dispositivos encontrados en busca de versiones de los servicios que ofrecen, así como del sistema operativo. La opción -sV se usa para obtener información del servicio y versión de los puertos abiertos. La opción "-O" se usa para detectar el sistema operativo [Imagen 130].

- `# sudo nmap -sS -sV -O -Pn -F -f -oA version 192.168.22.1`

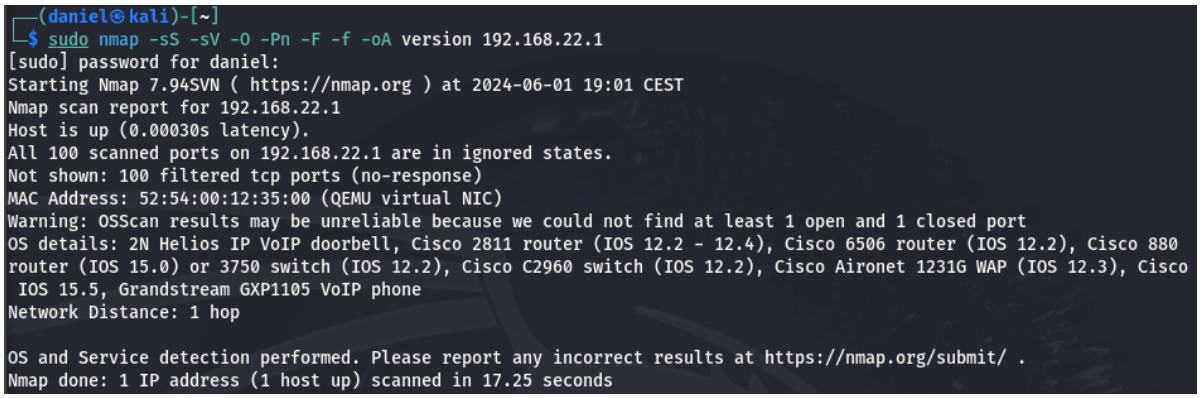


Imagen 130. *Comando Nmap escaneo version*

Se seguirá realizando la misma acción para los demás dispositivos que se encuentran en la red. Así se va guardando la información que servirá de ayuda para posteriormente pasar a identificar vulnerabilidades.

Como se han ido guardando los resultados del escaneo en varios archivos .xml, se importarán en la herramienta *Metasploit* para analizar y ver la información de los resultados de una manera más ordenada y limpia.

Se ejecutará *Metasploit* de la siguiente manera.

- `# sudo msfdb init && mfsconsole`

Se importarán los resultados obtenidos de los escaneos [Imagen 131].

- `# msf6 > db_import /home/daniel/version1.xml`

```
msf6 > db_import /home/daniel/version.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.13.10'
[*] Importing host 192.168.22.17
[*] Successfully imported /home/daniel/version.xml
```

Imagen 131. *Import Metasploit version*

Una vez importados todos los resultados se ejecutarán los siguientes comandos para ver toda la información [Imagen 132].

- `# msf6 > hosts`
- `# msf6 > services`

```
msf6 > hosts

Hosts
=====

address       mac                name  os_name   os_flavor  os_sp  purpose  info  comments
-------       ---                ----  -------   ---------  -----  -------  ----  --------
192.168.22.1  52:54:00:12:35:00        embedded               device
192.168.22.2  52:54:00:12:35:00        ASA                9.X  device
192.168.22.9  08:00:27:19:05:07        Linux              3.X  server
192.168.22.10 08:00:27:84:ca:b2        Linux              3.X  server
192.168.22.11 08:00:27:78:8a:54        Linux              3.X  server
192.168.22.12 08:00:27:56:82:01        Linux              3.X  server
192.168.22.13 08:00:27:b7:22:20        Linux              3.X  server
192.168.22.14 08:00:27:9b:6e:e4        Linux              3.X  server
192.168.22.15 08:00:27:f5:31:c8        Linux              3.X  server
192.168.22.16 08:00:27:41:71:dc        Linux              3.X  server
192.168.22.17 08:00:27:66:a7:fe        Linux              3.X  server

msf6 > services
Services
========

host          port  proto  name      state  info
----          ----  -----  ----      -----  ----
192.168.22.9  22    tcp    ssh       open   OpenSSH 7.4 protocol 2.0
192.168.22.10 22    tcp    ssh       open   OpenSSH 7.4 protocol 2.0
192.168.22.10 9200  tcp    ssl/rtsp  open
192.168.22.11 22    tcp    ssh       open   OpenSSH 7.4 protocol 2.0
192.168.22.12 22    tcp    ssh       open   OpenSSH 7.4 protocol 2.0
192.168.22.13 22    tcp    ssh       open   OpenSSH 7.4 protocol 2.0
192.168.22.14 22    tcp    ssh       open   OpenSSH 7.4 protocol 2.0
192.168.22.15 22    tcp    ssh       open   OpenSSH 7.4 protocol 2.0
192.168.22.16 22    tcp    ssh       open   OpenSSH 7.4 protocol 2.0
192.168.22.17 22    tcp    ssh       open   OpenSSH 7.4 protocol 2.0
```

Imagen 132. *Información mostrada en Metasploit*

Como se puede observar los equipos *192.168.22.9-10* son equipos con sistema operativo *Linux*. Mientras que el 192.168.22.1 que sería la puerta de enlace de la red, que puede ser un *router*, y el 192.168.22.2 que tiene sistema operativo ASA, el que parece ser una especie de firewall.

Con esta información se pasará enumerar las posibles vulnerabilidades en cada equipo que se ha visto que contiene sistema operativo *Linux*.

9.2 Vulnerabilidades equipos *192.168.22.9-17*

Se hará uso de la herramienta nmap para detectar las posibles vulnerabilidades que pueda presentar los equipos *192.168.22.9-17*. Se usará la opción "*--script vuln*" para la búsqueda de vulnerabilidades [Imagen 133].

- `# sudo nmap -sS -Pn -f --script vuln -oA vulns9 192.168.22.9`

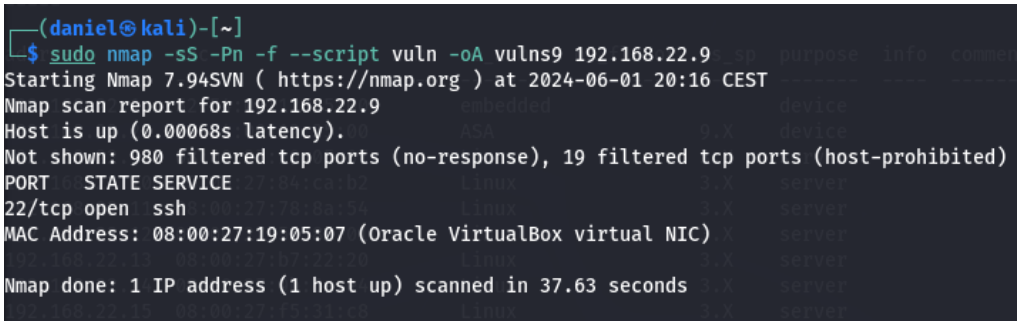


Imagen 133. *Comando Nmap vulns*

Se ejecutará lo mismo para todos los equipos y luego se importará en *Metasploit*.

- `# msf6 > db_import /home/daniel/vulns1.xml`

Una vez importados todos los resultados se reflejarán de la siguiente manera [Imagen 134].

- `# msf6 > vulns`

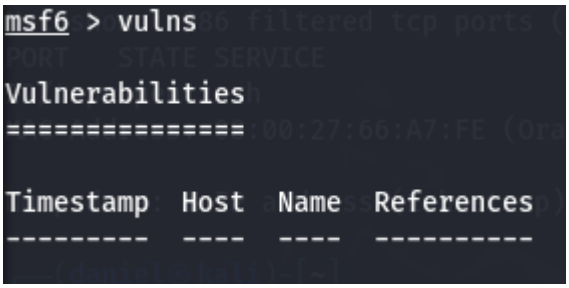


Imagen 134. *Vulns en Metasploit*

Como se puede observar no se encuentran vulnerabilidades explotables.

Una vez visto las posibles vulnerabilidades en cada uno de los equipos, se pasarán a analizar las vulnerabilidades en cada uno de los puertos abiertos.

9.2.1 Puerto 22

Este puerto se usa para conexiones seguras *SSH* y *SFTP*. La función de *SSH* es el acceso remoto a un servidor por medio de un canal seguro.

Se intentará una conexión mediante *SSH* con el dispositivo *192.168.22.9* [Imagen 135].

- `# sudo ssh root@192.168.22.9`

Imagen 135. *Comando SSH*

Como se puede comprobar se permite la conexión mediante *SSH* y se pide la contraseña del equipo *192.168.22.9* para establecer la conexión.

Se harán varios intentos de *loggeo* probando contraseñas aleatorias. Después de varios intentos se deniega el permiso y se cierra la conexión. Se probarán diferentes credenciales en todos los equipos para probar entrar.

9.2.2 Puerto 9200

Se puede comprobar que en el equipo *192.168.22.10* se ha detectado una vulnerabilidad [Imagen 136].

- `# sudo nmap -sS -Pn -f --script vuln -oA vulns10 192.168.22.10`

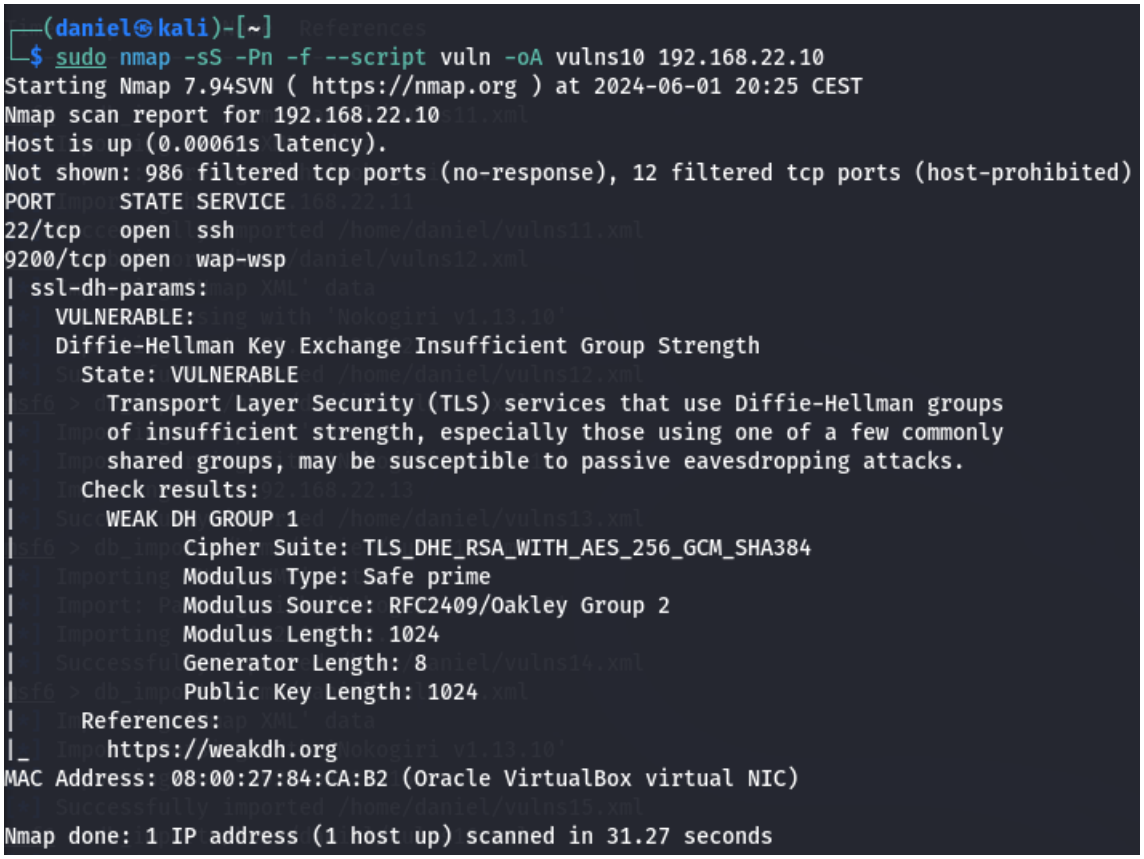


Imagen 136. *Comando Nmap vulns2*

Como se puede observar se trata de una vulnerabilidad *"Diffie-Hellman Key Exchange Insufficient Group Strengh",* pero no existe un *exploit* específico para esta vulnerabilidad. Sin embargo, esta vulnerabilidad puede ser explotada por un ataque de "*man-in-the-middle*" usando *Metasploit*. Se buscarán módulos relacionados con MITM [Imagen 137].

- `# msf6 > search mitm`

```
msf6 > search mitm

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank       Check  Description
   -  ----                                                  ---------------  ----       -----  -----------
   0  exploit/android/browser/webview_addjavascriptinterface  2012-12-21       excellent  No     Android Browser and
WebView addJavascriptInterface Code Execution
   1  auxiliary/server/jsse_skiptls_mitm_proxy              2015-01-20       normal     No     Java Secure Socket E
xtension (JSSE) SKIP-TLS MITM Proxy
   2  exploit/osx/local/libxpc_mitm_ssudo                   2018-03-15       excellent  Yes    Mac OS X libxpc MITM
 Privilege Escalation
   3  exploit/windows/browser/malwarebytes_update_exec      2014-12-16       good       No     Malwarebytes Anti-Ma
lware and Anti-Exploit Update Remote Code Execution
   4  auxiliary/server/openssl_altchainsforgery_mitm_proxy  2015-07-09       normal     No     OpenSSL Alternative
Chains Certificate Forgery MITM Proxy
   5  auxiliary/scanner/ssl/openssl_ccs                     2014-06-05       normal     No     OpenSSL Server-Side
ChangeCipherSpec Injection Scanner
   6  auxiliary/server/wpad                                                  normal     No     WPAD.dat File Server
   7  post/windows/manage/pptp_tunnel                                        normal     No     Windows Manage Remot
e Point-to-Point Tunneling Protocol

Interact with a module by name or index. For example info 7, use 7 or use post/windows/manage/pptp_tunnel
```

Imagen 137. *MITM Metasploit*

Se seleccionará el módulo "auxiliary/*server*/jsse_skiptls_mitm_proxy" que se usa para OpenSSL y este puerto 9200 ofrecía un servicio *SSL*.

- `# msf6 > use 1`
- `# msf6 > set SRVHOST 192.168.22.6`
- `# msf6 > set SRVPORT 8443`
- `# msf6 > set RHOST 192.168.22.10`
- `# msf6 > set RPORT 9200`
- `# msf6 > run`

Se ejecutará Wireshark mientras se encuentra escuchando, usando el puerto 8443, se seleccionará la interfaz de red y se filtrará lo siguiente para escuchar el puerto 9200 y dirección IP *192.168.22.10*.

- `IP.addr == 192.168.22.10 && tcp.port == 9200`

9.3 Ataques a las máquinas *192.168.22.9-17*

Enumeradas y explotadas las vulnerabilidades se pasará a ejecutar otro tipo de ataques y pruebas.

9.3.1 Ataques de Fuerza Bruta

Se realizarán ataques de fuerza bruta hacia todas las máquinas *192.168.22.9-17*. Para ello se utilizará la herramienta Hydra [Imagen 138]. Este tipo de ataque consiste en lanzar conexiones *SSH* probando múltiples credenciales.

- `# sudo hydra -l usuario -P passwords.txt ssh://192.168.22.11`

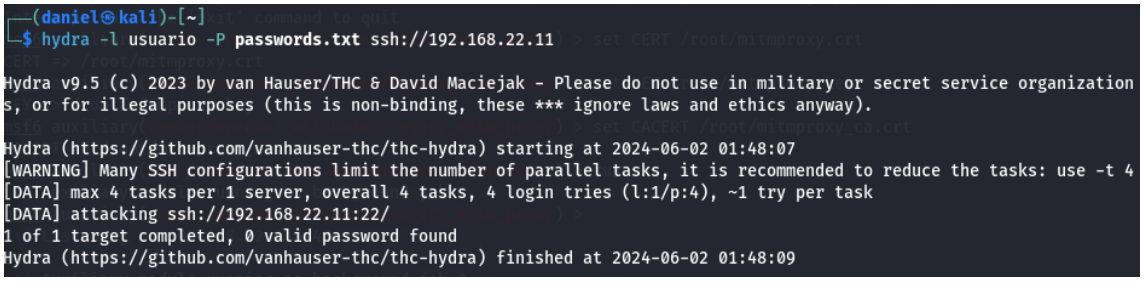


Imagen 138. *Ataque fuerza bruta*

9.3.2 Ataques de denegación de servicio

Un ataque de denegación de servicio (DOS) se basa en un ataque llamado *Flood* de paquetes, que consiste en enviar una gran cantidad de paquetes SYN a un puerto específico y sobrecargar un servicio. Se realiza este tipo de ataque debido a que en la IP *192.168.22.10* estaba abierto el puerto 9200 que se usa para *Elasticsearch*. Un *Elasticsearch* es usado por un SIEM de ciberseguridad para coleccionar documentos almacenados como *JSON*. Por lo que se llega a la conclusión que debe de haber un servidor que use alguna herramienta de monitorización.

A continuación, se realizan los ataques *DOS* sobre las máquinas *192.168.22.9-17* para intentar sobrecargar algún servicio. Para ello se usará la herramienta hping3. La opción "-S" indica que son paquetes SYN. La opción "-p" especifica el puerto. La opción "-c" indica el número de paquetes que se envían.

- `# sudo hping3 -S -p 80 -c 10000 192.168.22.10`

Se irán editando los puertos más comunes que usan protocolo TCP y también se añadirá la opción "-i u1000" para enviar 1000 cada unidad de segundo.

- `# sudo hping3 -i u1000 -S -p 443 -c 10000000000 192.168.22.11`

10. Monitorización de los eventos de seguridad

Después de realizar la fase de enumeración desde la máquina atacante *Kali*, se observarán los eventos detectados en los agentes de *Wazuh* y en los nodos servidor del clúster de *Wazuh*.

10.1 *Security* Events

Para ello se accederá el *Wazuh dashboard* en la máquina virtual *Indexer*. En la sección *"Security Information Management"* se hará clic en *"Security events"* y se abrirá una página donde se podrán observar los resultados de los logs tras la monitorización de diferentes eventos [Imagen 139].

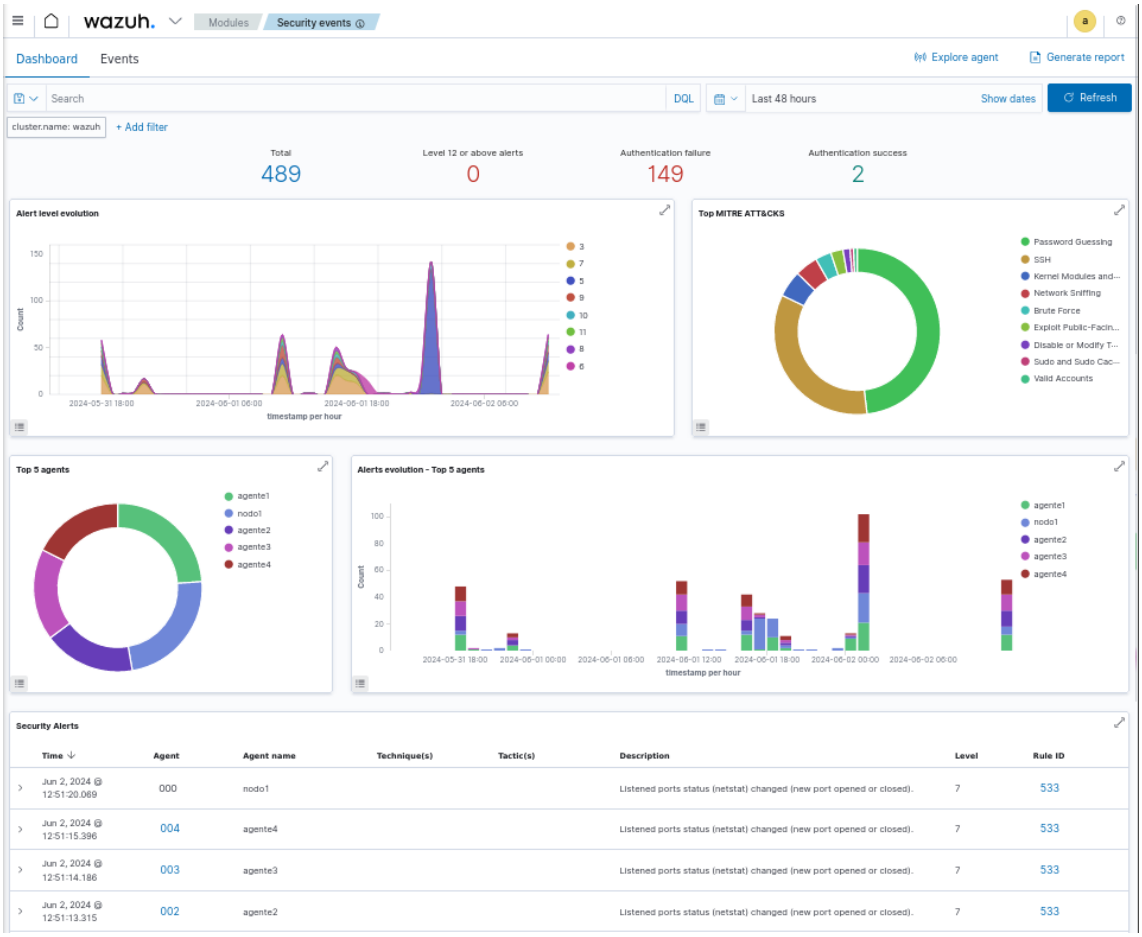


Imagen 139. *Security Events*

Como se puede observar a simple vista el *Wazuh dashboard* ha devuelto una serie de alertas que ha monitoreado. Para empezar, ha detectado 489 eventos en total en las últimas 48 horas [Imagen 140], los cuales no todo deben de ser ataques, ya que también se muestran reinicios o arranques de servicios, cambios en los sistemas o puertos y sesiones abiertas con usuario *root*.

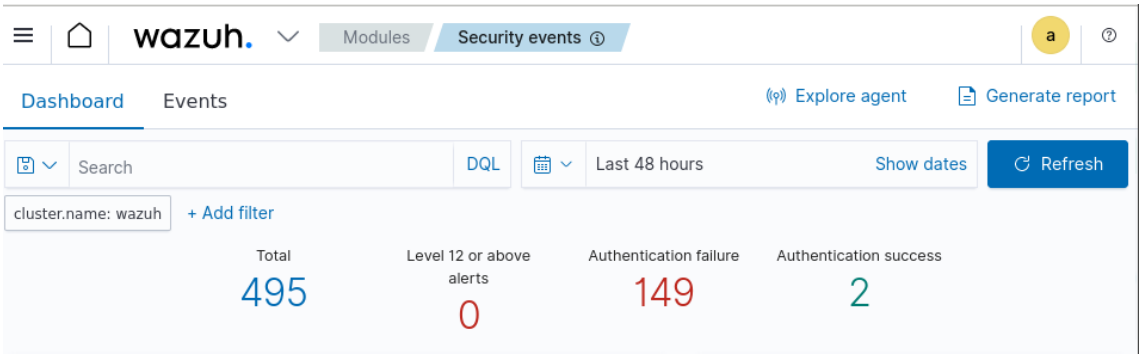


Imagen 140. *Security events alerts*

También se puede observar que ha detectado 149 intentos de autenticación fallidos. Se observará que hay una ventana llamada "Alert level evolution", que muestra una gráfica del número de alertas por niveles de alertas que existen y la evolución de estas alertas en el tiempo [Imagen 141].

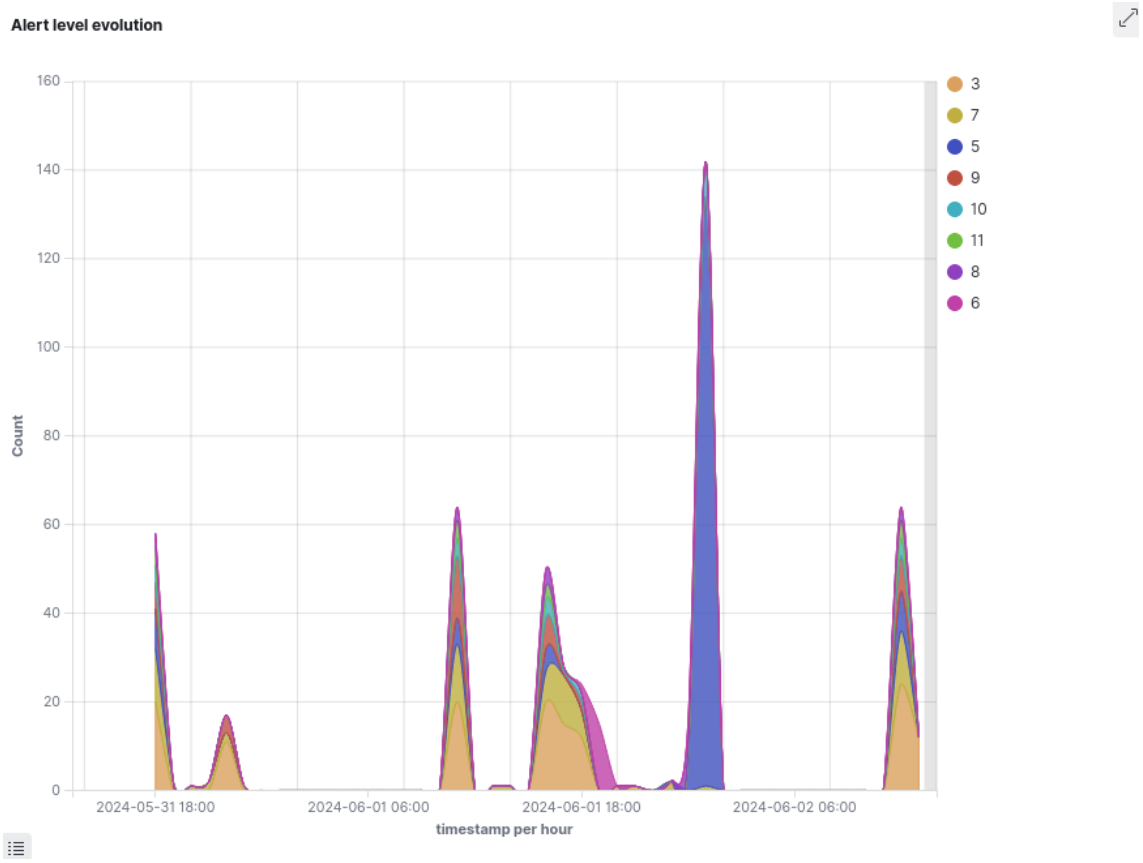


Imagen 141. *Gráfica Alert level detection*

A la derecha de esta última gráfica, se encuentra una gráfica donut chart llamada "Top *MITRE ATT&CKS*" [Imagen 142] que muestra el porcentaje de cada tipo de tácticas o técnicas de ataques. *MITRE ATT&CK* [25] es una herramienta que recoge estás técnicas o tácticas que se usan para comprometer la seguridad, proporcionando un lenguaje común en este ámbito.

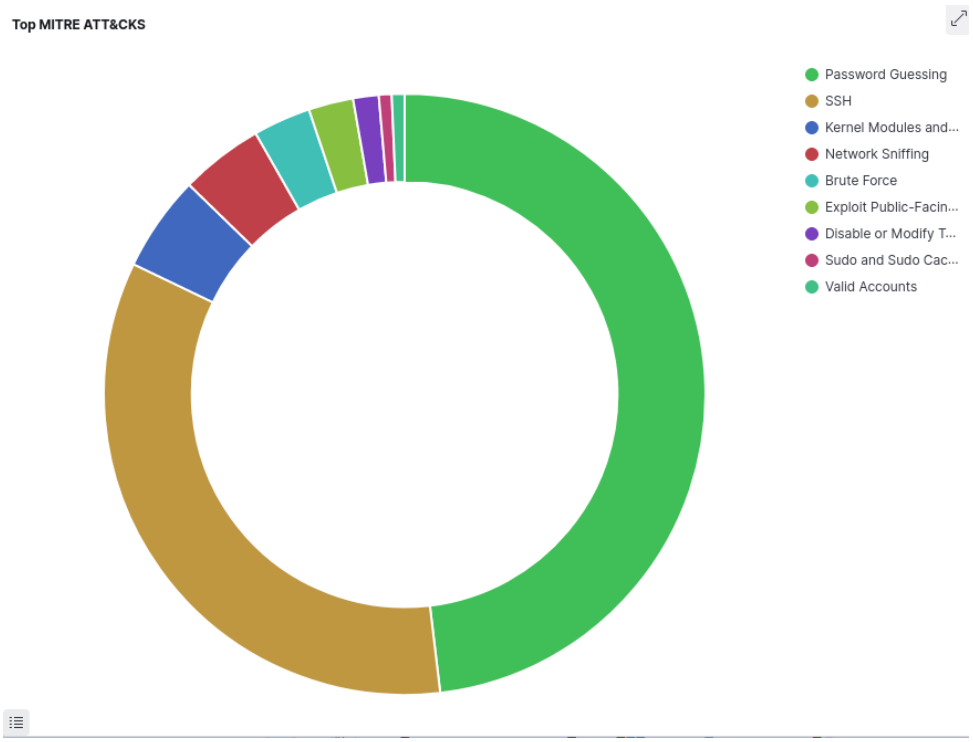


Imagen 142. *Gráfica Top MITRE ATT&CKS*

Como se puede observar se han monitorizado eventos relacionados con técnicas o tácticas recogidas en *MITRE ATT&CK* como son *Password Guessing*, *SSH*, Brute *Force*, etc.

La siguiente gráfica llamada "*Top 5 agents*", muestra el porcentaje y el número de eventos detectados en los 5 agentes donde más eventos se han detectado [Imagen 143].

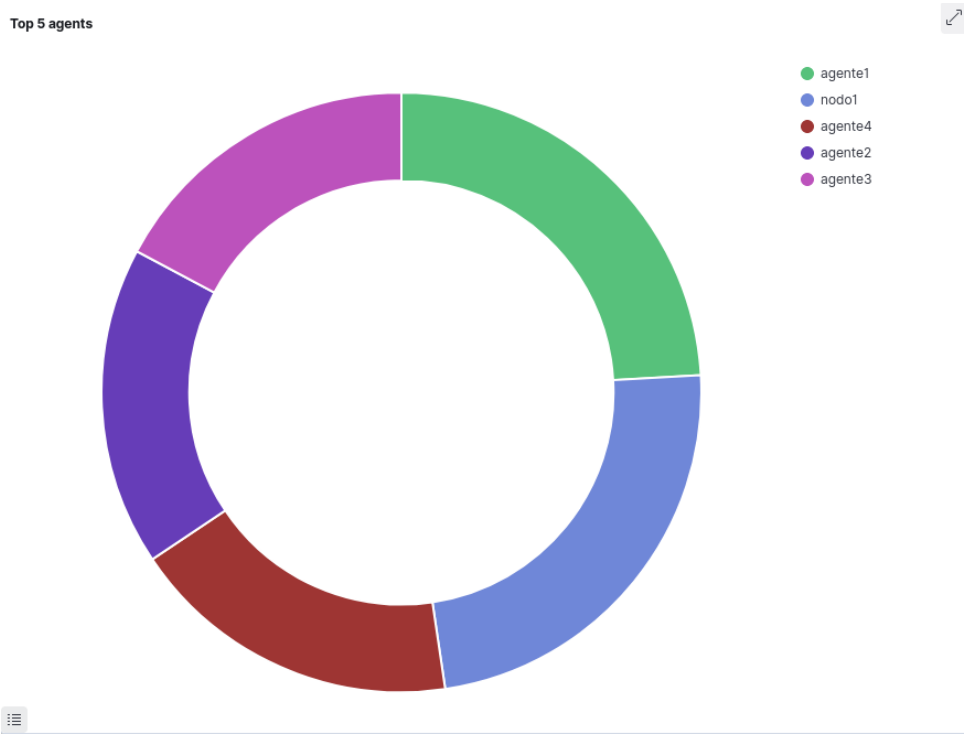


Imagen 143. *Gráfica Top 5 agents*

La última gráfica llamada "*Alerts evolution – top 5 agents*" muestra el número de eventos de los 5 agentes que más eventos se han detectado a lo largo del tiempo [Imagen 144].

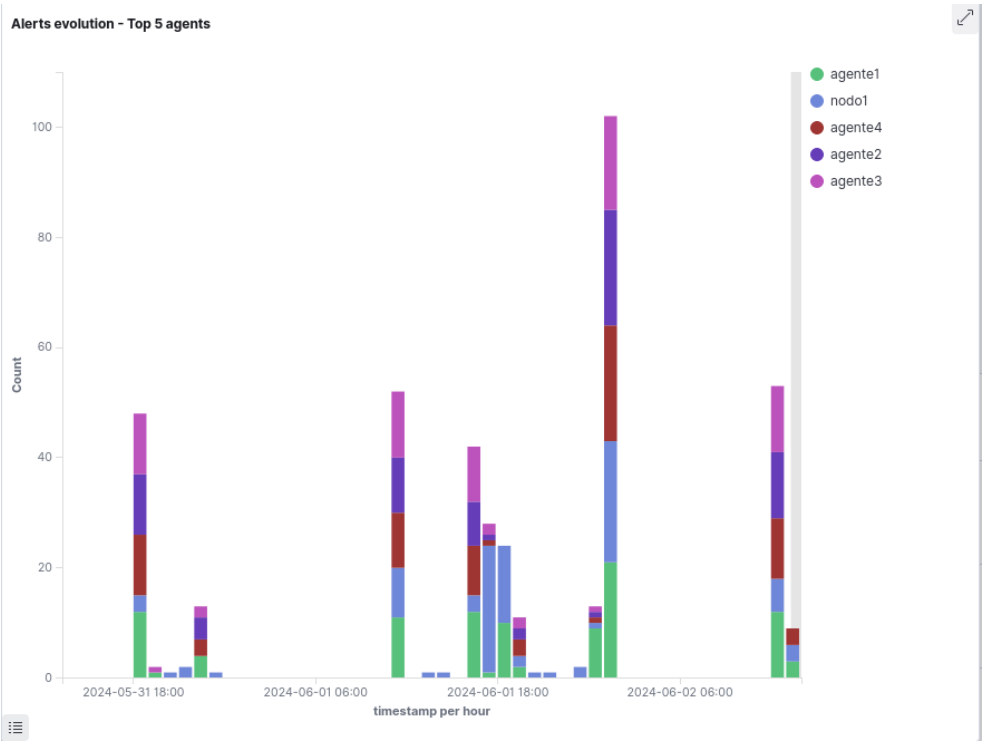


Imagen 144. *Gráfica Alerts evolution – Top 5 agents*

A continuación, se pasará a observar la lista *"Security alerts"* que son los eventos detectados por *Wazuh* ordenados por el tiempo de más recientes a más antiguos. En la sección *"Security events"* se ha indicado mostrar los eventos detectados en las últimas 48 horas, por lo que se buscarán las alertas sobre los ataques que se han realizado desde la máquina atacante *Kali*.

Para buscar las alertas de manera más rápida y fácil, en la gráfica "Top *MITRE* ATT&ACKS" se hará clic en el tipo de táctica para visualizar en la lista "*Security Alerts*" los eventos relacionados con ese tipo de técnica.

10.1.1 *Password Guessing*

Se hará clic en la gráfica "Top *MITRE ATT&CKS*" sobre la técnica "*Password Guessing*" y nos aparecerán en la lista "*Security Alerts*" los eventos monitorizados relacionados con ese tipo de técnicas [Imagen 145].

Security Alerts

Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jun 2, 2024 @ 01:49:03.155	004	agente4	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jun 2, 2024 @ 01:49:03.153	004	agente4	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jun 2, 2024 @ 01:49:03.151	004	agente4	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jun 2, 2024 @ 01:49:03.148	004	agente4	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jun 2, 2024 @ 01:49:01.200	004	agente4	T1110.001	Credential Access	PAM: User login failed.	5	5503
Jun 2, 2024 @ 01:49:01.200	004	agente4	T1110.001	Credential Access	PAM: User login failed.	5	5503
Jun 2, 2024 @ 01:49:01.200	004	agente4	T1110.001	Credential Access	PAM: User login failed.	5	5503
Jun 2, 2024 @ 01:49:01.200	004	agente4	T1110.001	Credential Access	PAM: User login failed.	5	5503
Jun 2, 2024 @ 01:49:01.180	004	agente4	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jun 2, 2024 @ 01:49:01.180	004	agente4	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710

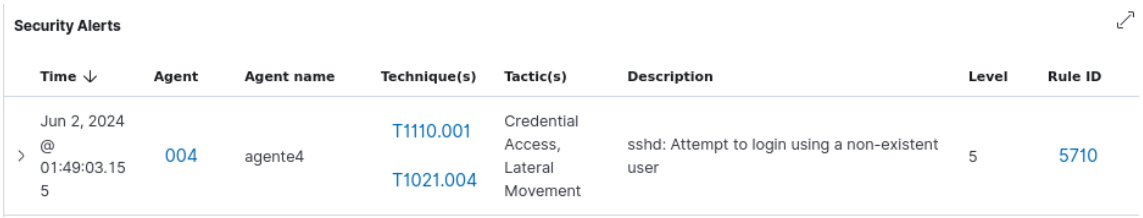
Rows per page: 10 ‹ 1 2 3 4 5 … 14 ›

Imagen 145. *Lista Security alerts*

Se podrá observar que los dispositivos donde se han detectado estos eventos son nodo1, nodo2, nodo3, agente1, agente2, agente3 y agente4. Se observa también que se han monitorizado dos tipos de eventos.

El primero indica que se trata de una táctica "*Credential Access*, *Lateral Movement*" y su descripción indica lo siguiente "*sshd: Attempt to login using a non-existent user*". Como primera interpretación se puede deducir que se trata de un intento de *loggeo* fallido a través del servicio *SSH*, por el que se ha intentado un *login* usando

un usuario que no existe en el sistema [Imagen 146]. También se indica que se trata de una alerta de nivel 5, siendo 15 el nivel más alto.



Security Alerts

Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jun 2, 2024 @ 01:49:03.155	004	agente4	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710

Imagen 146. *Evento Password Guessing 1*

Para conocer más información sobre este evento que se ha monitorizado en el agente4 se hará clic sobre él [Imagen 147].

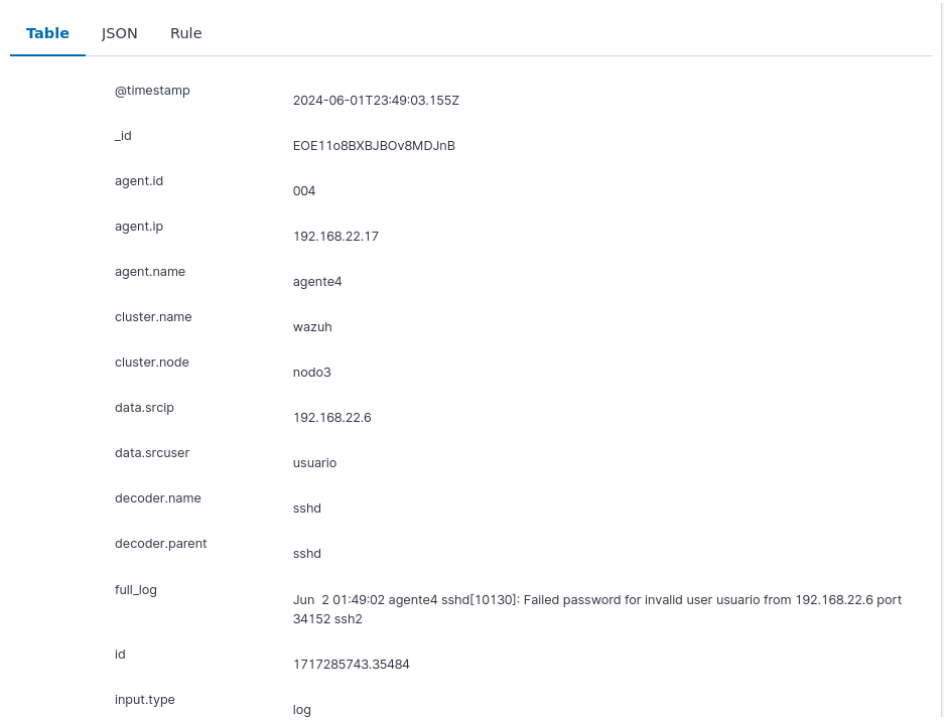


Table | JSON | Rule

@timestamp	2024-06-01T23:49:03.155Z
_id	EOE11o8BXBJBOv8MDJnB
agent.id	004
agent.ip	192.168.22.17
agent.name	agente4
cluster.name	wazuh
cluster.node	nodo3
data.srcip	192.168.22.6
data.srcuser	usuario
decoder.name	sshd
decoder.parent	sshd
full_log	Jun 2 01:49:02 agente4 sshd[10130]: Failed password for invalid user usuario from 192.168.22.6 port 34152 ssh2
id	1717285743.35484
input.type	log

Imagen 147. *Evento Password Guessing 1 info*

Como se puede observar, los resultados se recogen en esta tabla donde se visualizan de manera más cómoda, que también pueden ser visualizados como un archivo tipo *JSON*. Los resultados indican que el evento monitorizado pertenece al agente4 con dirección IP *192.168.22.17*, que este agente se encuentra conectado al clúster de *Wazuh*, específicamente manteniendo una conexión con el nodo3.

También se observa que la máquina desde la que se realizó la técnica es la *192.168.22.6*, que se corresponde con la dirección IP de la máquina atacante *Kali*. El *decoder* indica que se realizó la técnica sobre el servicio *sshd* y en el parámetro "*full_log*" se muestra el log relacionado con este *evento "Jun 2 01:49:02 agente4 sshd[10130]: Failed password for invalid user usuario from 192.168.22.6 port 34152 ssh2"*.

En el mismo evento se reflejan enlaces a las reglas en la base de datos de "*MITRE ATT&CK*". Aparecen en este evento las técnicas "*T1110.001*" y "*T1021.004*" que haciendo clic sobre alguna de ellas se muestra la información sobre estas reglas recogidas en la base de datos de "*MITRE ATT&CK*" [Imagen 148].

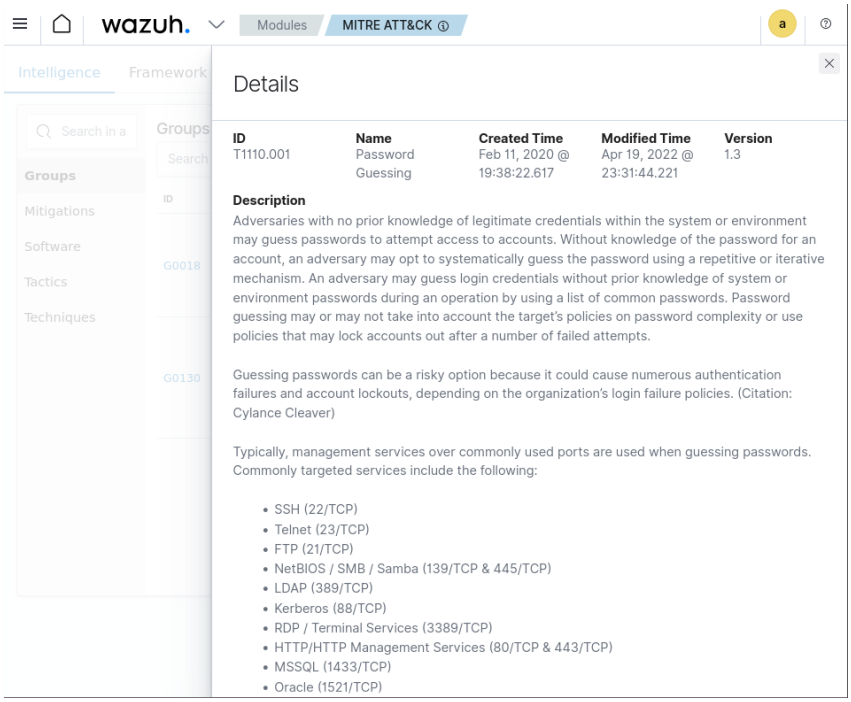


Imagen 148. *Evento Password Guessing 1 details*

El otro tipo de evento que se ha detectado se trata de una técnica "*Credential Access*" también y su descripción "*PAM: User login failed*" que indica que se trata de un intento fallido de inicio de sesión *PAM* (*Pluggable Authentication Module*), que se trata de un tipo de módulo de autenticación usado por *SSH,* por ejemplo. Este evento indica un intento fallido de autenticación *SSH*, posiblemente como parte de un ataque de adivinación de contraseñas. Se trata también de una alerta de nivel 5. Se mostrará más información acerca del evento haciendo clic sobre él [Imagen 149].

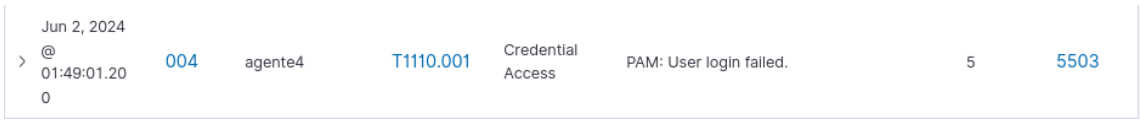


Imagen 149. *Evento Password Guessing 2*

Se observa que se trata de un evento monitorizado en el agente4 que se encuentra conectado al clúster de *Wazuh*, específicamente manteniendo una conexión con el nodo3. Se trata de una técnica utilizada por una máquina con dirección IP *192.168.22.6*, que se corresponde con la máquina desde la que se han realizado los ataques. El log entero relacionado con el evento es *"Jun 2 01:49:00 agente4*

sshd[10130]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.22.6” Se puede observar que la técnica de “*MITRE ATT&CK*” es la misma que en el evento recogido anteriormente [Imagen 150].

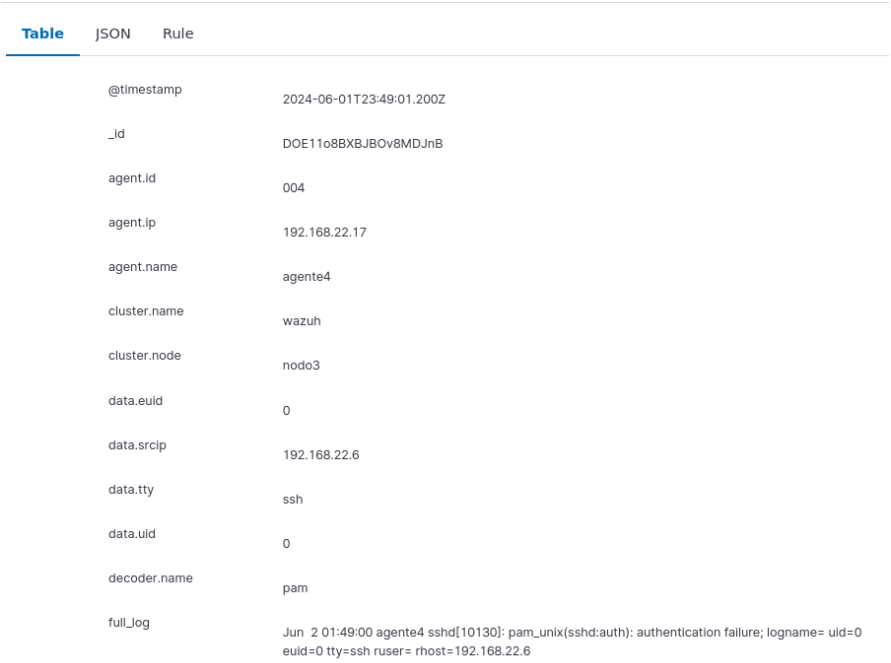
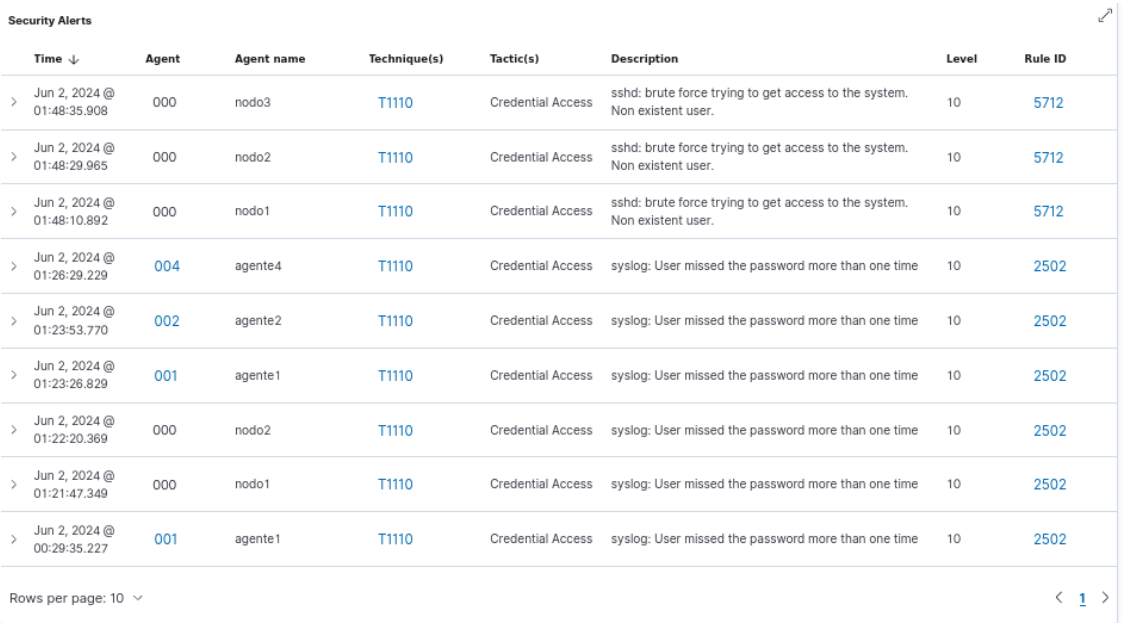


Imagen 150. *Evento Password Guessing 2 info*

Se observa que estos dos tipos de eventos se repiten varias veces y monitorizados en todos los agentes y nodos del servidor de *Wazuh*. Por lo que se podría interpretar como un intento de ataque de fuerza bruta ya que se producen varios eventos similares en un espacio corto de tiempo. O incluso un intento de conexión mediante *SSH* realizado manualmente repetido continuamente.

10.1.2 Brute Force

También conocido como ataque de fuerza bruta. Se hará clic en la gráfica “*Top MITRE ATT&CKS*” sobre la técnica “*Brute Force*” y nos aparecerán en la lista “*Security Alerts*” los eventos monitorizados relacionados con ese tipo de técnica [Imagen 151].



Security Alerts

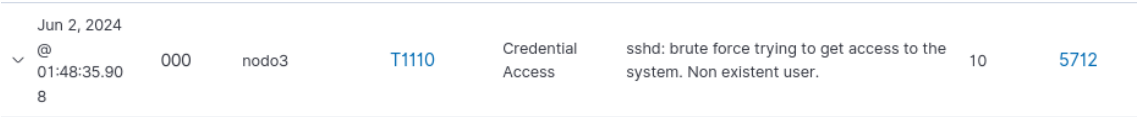
Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jun 2, 2024 @ 01:48:35.908	000	nodo3	T1110	Credential Access	sshd: brute force trying to get access to the system. Non existent user.	10	5712
Jun 2, 2024 @ 01:48:29.965	000	nodo2	T1110	Credential Access	sshd: brute force trying to get access to the system. Non existent user.	10	5712
Jun 2, 2024 @ 01:48:10.892	000	nodo1	T1110	Credential Access	sshd: brute force trying to get access to the system. Non existent user.	10	5712
Jun 2, 2024 @ 01:26:29.229	004	agente4	T1110	Credential Access	syslog: User missed the password more than one time	10	2502
Jun 2, 2024 @ 01:23:53.770	002	agente2	T1110	Credential Access	syslog: User missed the password more than one time	10	2502
Jun 2, 2024 @ 01:23:26.829	001	agente1	T1110	Credential Access	syslog: User missed the password more than one time	10	2502
Jun 2, 2024 @ 01:22:20.369	000	nodo2	T1110	Credential Access	syslog: User missed the password more than one time	10	2502
Jun 2, 2024 @ 01:21:47.349	000	nodo1	T1110	Credential Access	syslog: User missed the password more than one time	10	2502
Jun 2, 2024 @ 00:29:35.227	001	agente1	T1110	Credential Access	syslog: User missed the password more than one time	10	2502

Rows per page: 10 1

Imagen 151. *Brute Force events*

Se podrá observar que los dispositivos donde se han detectado estos eventos son nodo1, nodo2, nodo3, agente1, agente2 y agente4. Se observa también que se han monitorizado dos tipos de eventos.

El primer evento indica que se trata de una táctica "*Credential Access*" y su descripción indica lo siguiente "*sshd: brute force trying to get access to the system. Non existent user*". Como primera interpretación se puede deducir que se trata de un intento de ataque de fuerza bruta para acceder al sistema a través de *SSH* utilizando un usuario no existente [Imagen 152].



Jun 2, 2024 @ 01:48:35.908	000	nodo3	T1110	Credential Access	sshd: brute force trying to get access to the system. Non existent user.	10	5712

Imagen 152. *Evento Brute force*

Para conocer más información sobre este evento que se ha monitorizado en el *nodo3* se hará clic sobre él [Imagen 153].

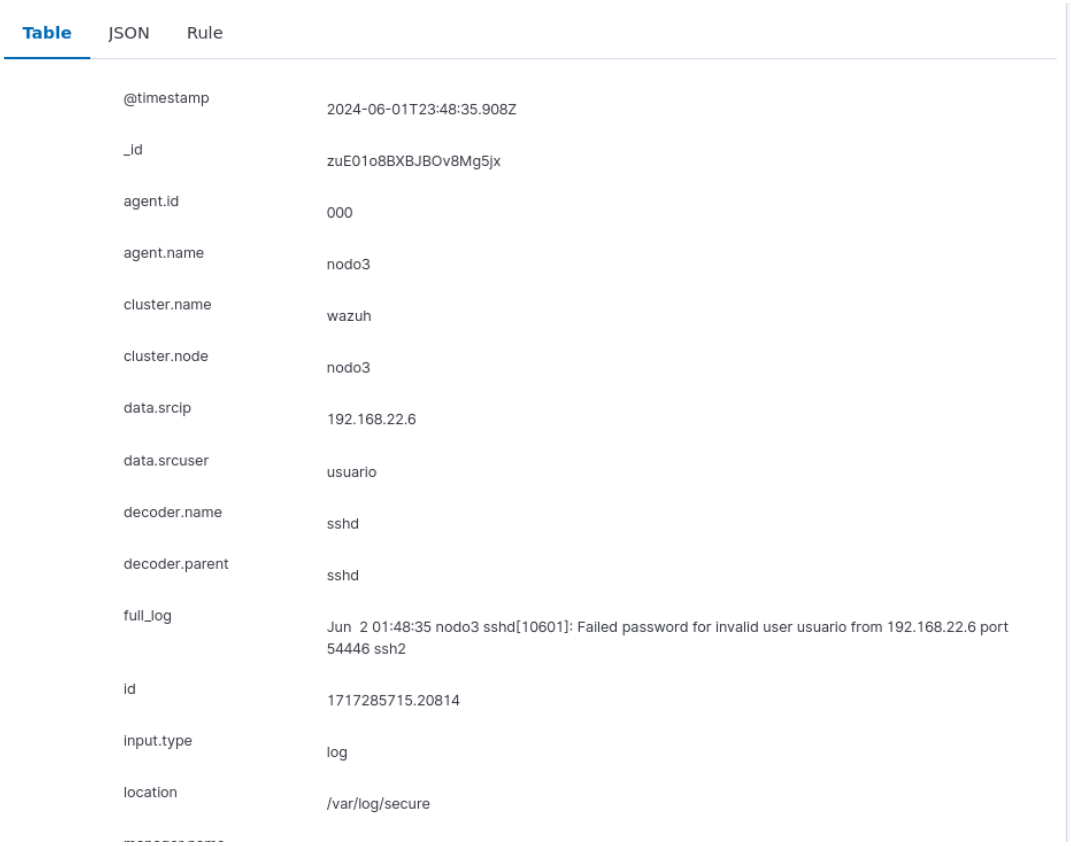


Imagen 153. *Evento Brute force info*

También se observa que la máquina desde la que se realizó la técnica es la *192.168.22.6*, que se corresponde con la dirección IP de la máquina atacante *Kali*. El decoder indica que se realizó la técnica sobre el servicio *sshd* y en el parámetro "*full_log*" se muestra el log relacionado con este *evento "Jun 2 01:48:35 nodo3 sshd[10601]: Failed password for invalid user usuario from 192.168.22.6 port 54446 ssh2"*.

Se describe una regla con la técnica "*T1110*" que si se accede a ella se muestra la información sobre esta técnica en la base de datos de "*MITRE ATT&CK*" [Imagen 154].

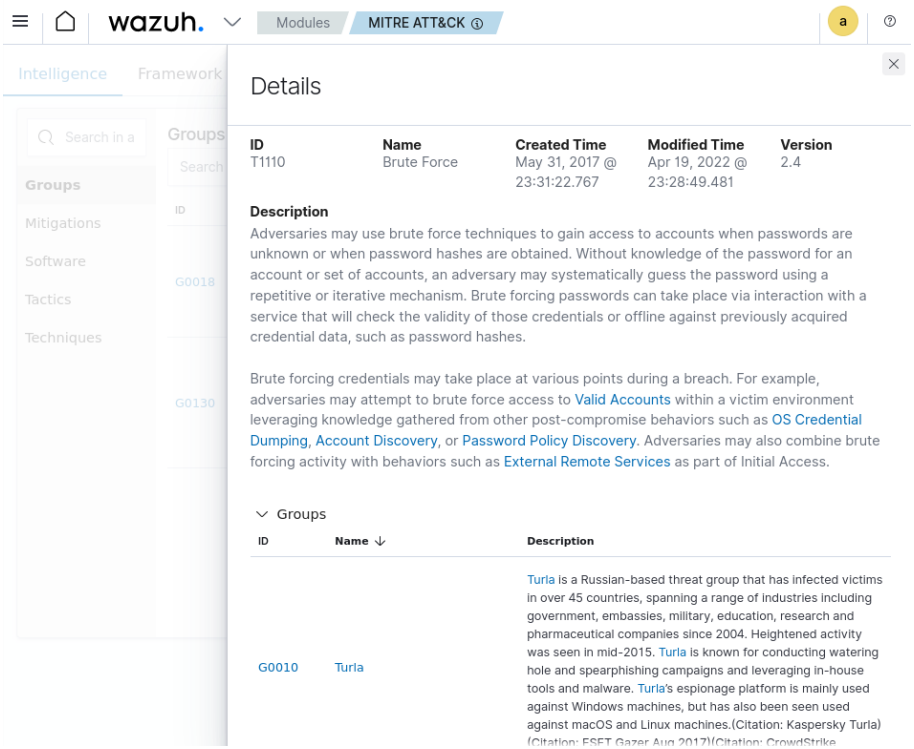


Imagen 154. *Evento Brute force details*

Este tipo de evento sólo se ha monitorizado en los nodos del servidor de *Wazuh*.

El segundo evento mostrado con esta técnica "*Credential Acces*" y su descripción "*syslog: User missed the password more than once*" que se trata de un intento fallido de autenticación *SSH* repetido utilizando un usuario, fallando en el intento de contraseña. Se trata de una alerta de nivel 10, ya que permitió la conexión SHH, pero se falló al autenticarse con la contraseña [Imagen 155].

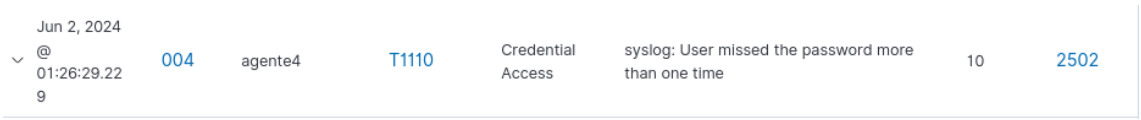


Imagen 155. *Evento Brute force 2*

Se hará clic sobre el evento para visualizar más información acerca de él. Se puede observar que es un evento monitorizado en el agente4 con dirección IP *192.168.22.17* [Imagen 156]. Este agente se conecta al clúster de *Wazuh*, específicamente manteniendo una conexión con el nodo3. Se ha realizado la técnica desde la máquina con dirección *192.168.22.6*, máquina desde la que se han realizado los ataques. Se ha realizado un ataque a través del servicio *SSH* y el log entero sería "*Jun 2 01:26:28 agente4 sshd[9887]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.22.6 user=root*" donde se puede observar que se ha realizado un intento de autenticación con el usuario *root*, realizando una autenticación fallida al desconocer la contraseña.

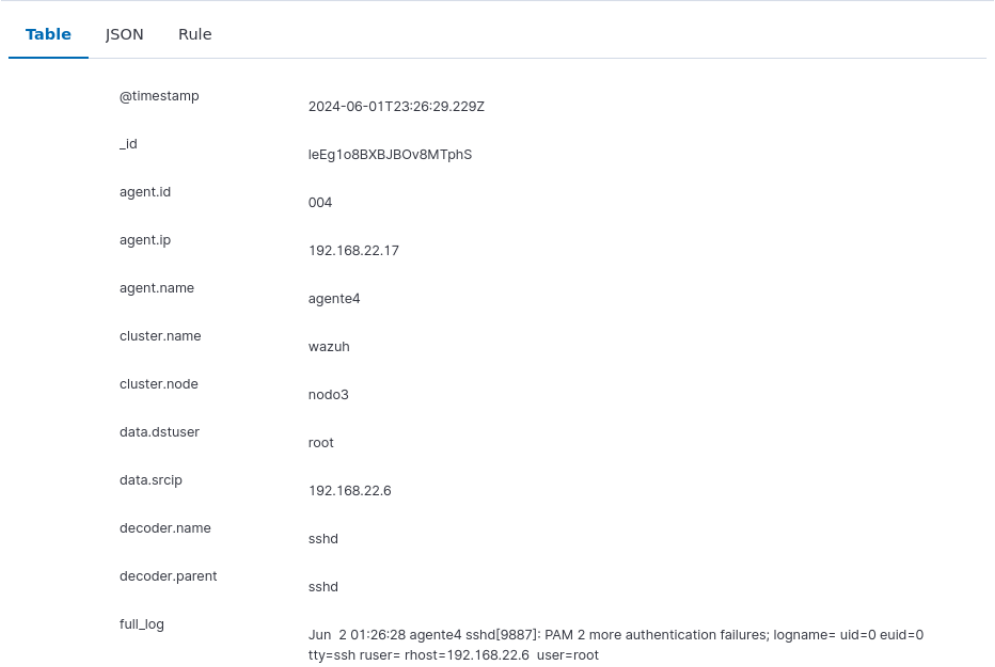
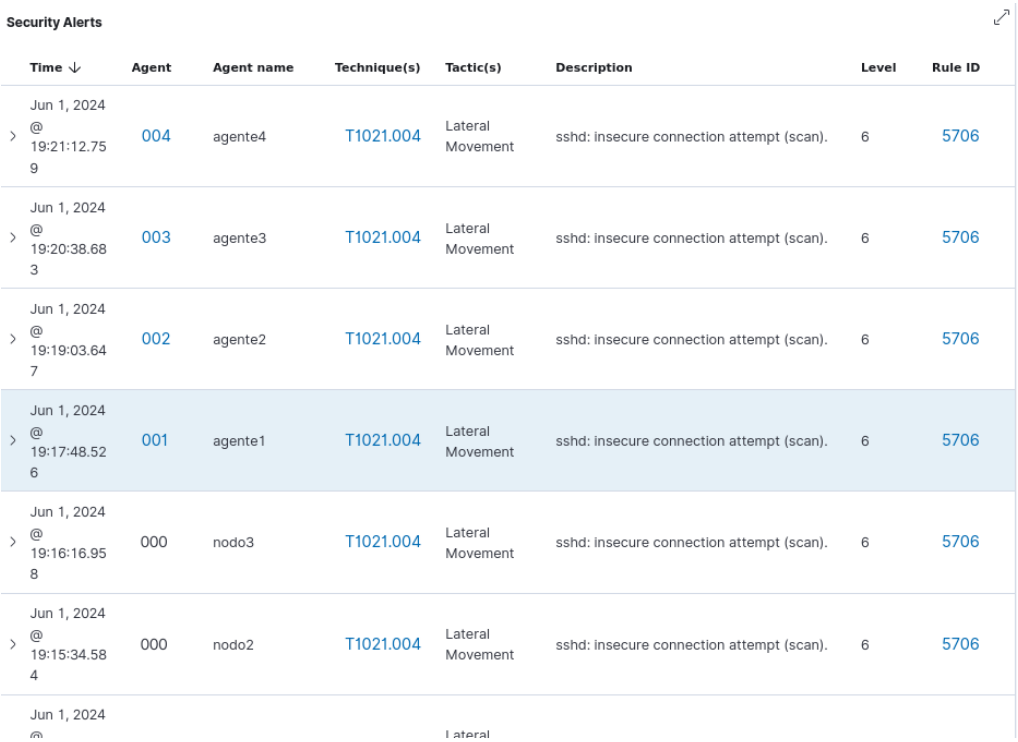


Imagen 156. *Evento Brute force 2 info*

A la vista de los resultados, este tipo de evento sólo se ha monitorizado en los agentes conectados al clúster de *Wazuh*.

10.1.3 *SSH*

Se hará clic en la gráfica "Top *MITRE ATT&CK*S" sobre la técnica "*SSH*" y nos aparecerán en la lista "*Security Alerts*" los eventos monitorizados relacionados con ese tipo de técnica [Imagen 157].

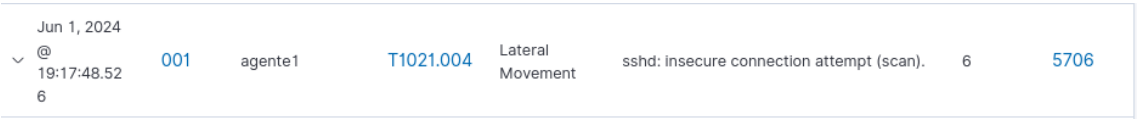


Security Alerts

Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jun 1, 2024 @ 19:21:12.759	004	agente4	T1021.004	Lateral Movement	sshd: insecure connection attempt (scan).	6	5706
Jun 1, 2024 @ 19:20:38.683	003	agente3	T1021.004	Lateral Movement	sshd: insecure connection attempt (scan).	6	5706
Jun 1, 2024 @ 19:19:03.647	002	agente2	T1021.004	Lateral Movement	sshd: insecure connection attempt (scan).	6	5706
Jun 1, 2024 @ 19:17:48.526	001	agente1	T1021.004	Lateral Movement	sshd: insecure connection attempt (scan).	6	5706
Jun 1, 2024 @ 19:16:16.958	000	nodo3	T1021.004	Lateral Movement	sshd: insecure connection attempt (scan).	6	5706
Jun 1, 2024 @ 19:15:34.584	000	nodo2	T1021.004	Lateral Movement	sshd: insecure connection attempt (scan).	6	5706
Jun 1, 2024 @				Lateral			

Imagen 157. *SHH events*

Se ha identificado un tipo de evento nuevo con una táctica "*Lateral Movement*" y descripción "*ssh: insecure connection attempt (scan).*", el cual parece ser un intento de conexión inseguro a través de SHH, identificado como un escaneo [Imagen 158].



Jun 1, 2024 @ 19:17:48.526	001	agente1	T1021.004	Lateral Movement	sshd: insecure connection attempt (scan).	6	5706

Imagen 158. *Evento SSH*

El evento ha sido monitorizado en el agente1 y se hará clic en el para ver más información acerca de este evento [Imagen 159].

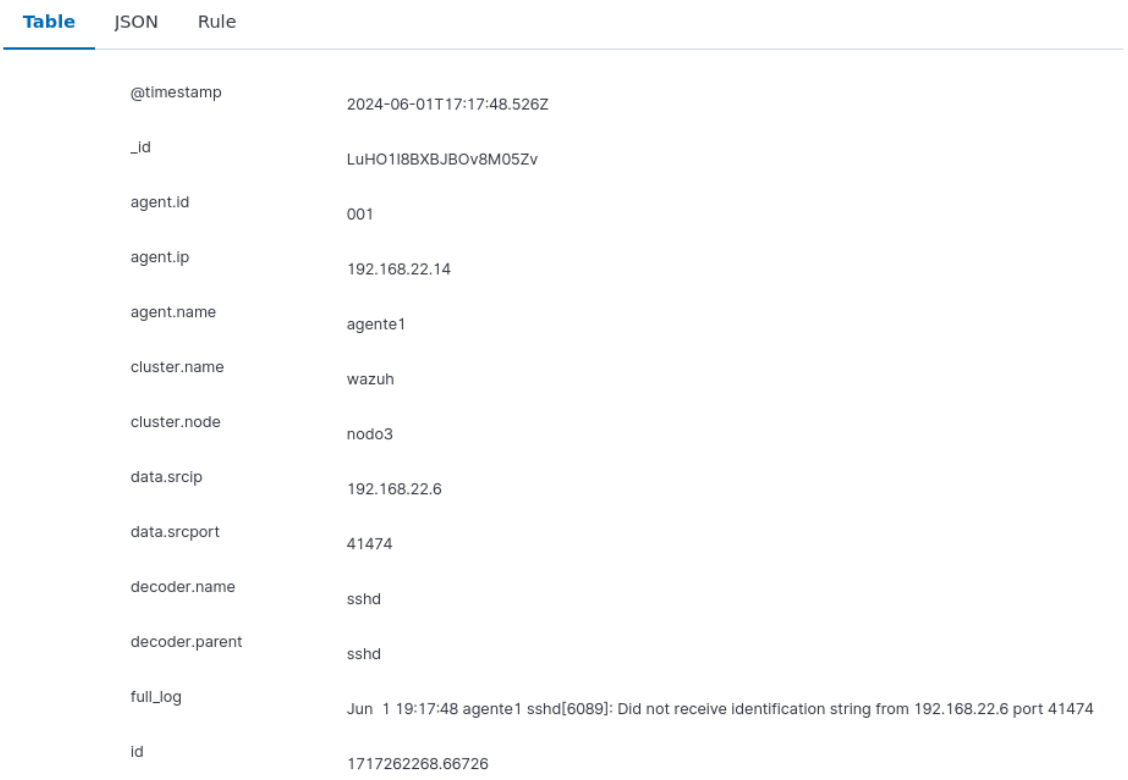


Imagen 159. *Evento SSH info*

Efectivamente se podrá observar que el evento ha sido monitorizado en el agente1 con dirección IP *192.168.22.14*, el cual se encuentra conectado al clúster de *Wazuh*, específicamente a través del nodo3. La dirección IP desde la que se ha realizado este escaneo es la *192.168.22.6*, que se corresponde con la máquina *Kali*, desde la que se han realizado los ataques. Se ha realizado esta técnica a través del servicio *SSH* y el log completo es el siguiente *"Jun 1 19:17:48 agente1 sshd[6089]: Did not receive identification string from 192.168.22.6 port 41474"* , por lo que este evento indica un intento de conexión inseguro a través de *SSH*, probablemente un escaneo de puertos.

A la vista de los resultados obtenidos, se podrán observar diversos eventos del mismo tipo monitorizados en todos los agentes y nodos del clúster de *Wazuh*, llegando a la conclusión de que este evento se relaciona con el escaneo realizado con *Nmap* por parte de la máquina *Kali* para escanear los puertos y servicios que ofrecen cada uno de los equipos en la red 192.168.22.0.

10.2 Conclusiones

Se han podido monitorizar satisfactoriamente muchas de las técnicas y ataques realizados por parte de la máquina *Kali*. Se verifica el correcto funcionamiento de la monitorización por parte de *Wazuh*.

También cabe destacar que la herramienta *Wazuh* permite ver lo eventos de muchas maneras diferentes, tanto como se hizo en este trabajo, como también

seleccionando cualquiera de los agentes o nodos donde se encuentra instalado *Wazuh* y así ver solo los eventos relacionados solo con uno de los equipos.

11. Conclusiones y trabajo futuro

En este TFT se han logrado el objetivo de desplegar una infraestructura de clúster con la herramienta Wazuh para monitorizar eventos de seguridad en alta disponibilidad. El uso de Nginx como balanceador de carga ha asegurado una distribución eficiente de las conexiones de los agentes al clúster y la continuidad del servicio incluso si un nodo falla.

El proceso de instalación y configuración ha demostrado que es posible usar herramientas de código abierto para crear soluciones de seguridad robustas y sin costo. Los eventos monitorizados, incluyendo los ataques desde las máquina virtual con Kali Linux, han confirmado la eficacia de Wazuh en detección y análisis de eventos de seguridad, demostrando su capacidad para proteger los sistemas informáticos de una organización.

De cara a un futuro trabajo, el siguiente paso clave para mejorar el proyecto sería actualizar el sistema operativo de CentOS 7 a una versión más moderna y soportada de Linux, garantizando que la infraestructura se mantenga protegida con las últimas actualizaciones de seguridad.

Tener un sistema operativo actualizado mejorará la seguridad, la compatibilidad con nuevas herramientas y ofrecerá un soporte mas sólido. Esta actualización sería fundamental para mantener un entorno seguro y eficiente.

12. Bibliografía

[1] «¿Qué significa SIEM y cómo funciona?,» Ambit BST | Consultoría regulatoria y de calidad en sector salud, [En línea]. Available: https://www.ambit-bst.com/blog/qu%C3%A9-significa-siem-y-c%C3%B3mo-funciona. [Último acceso: 02 06 2024].

[2] «Getting started with Wazuh - Wazuh documentation,» Wazuh documentation, [En línea]. Available: https://documentation.wazuh.com/current/getting-started/index.html. [Último acceso: 02 06 2024].

[3] «Components - Getting started with Wazuh - Wazuh documentation,» Wazuh documentation, [En línea]. Available: https://documentation.wazuh.com/current/getting-started/components/index.html. [Último acceso: 02 06 2024].

[4] «OpenVAS - Open Vulnerability Assessment Scanner,» OpenVAS - Open Vulnerability Assessment Scanner, [En línea]. Available: https://www.openvas.org/. [Último acceso: 02 06 2024].

[5] «AlienVault OSSIM is trusted by security professionals across the globe,» AlienVault OSSIM, [En línea]. Available: https://cybersecurity.att.com/products/ossim. [Último acceso: 02 06 2024].

[6] O. Fernandez, «Qué es Splunk y por qué Debería Importarte,» Aprender Big Data, 30 12 2022. [En línea]. Available: https://aprenderbigdata.com/splunk/. [Último acceso: 02 06 2024].

[7] «Cisco SecureX Software de seguridad,» Cisco, [En línea]. Available: https://www.cisco.com/site/mx/es/products/security/securex-platform/index.html. [Último acceso: 02 06 2024].

[8] «Download,» The CentOS Project, [En línea]. Available: https://www.centos.org/download/. [Último acceso: 02 06 2024].

[9] «¿Cómo Actualizar CentOS 7?,» Steemit, 2018. [En línea]. Available: https://steemit.com/technology/@ndnthor/como-actualizar-centos-7. [Último acceso: 02 06 2024].

[10] «¿Qué son las VirtualBox Guest Additions?,» Slice of *Linux*, 04 05 2009. [En línea]. Available: https://sliceoflinux.wordpress.com/2009/05/04/%C2%BFque-son-las-virtualbox-guest-additions/. [Último acceso: 02 06 2024].

[11] «Wazuh indexer - Installation guide - Wazuh documentation,» Wazuh documentation, [En línea]. Available: https://documentation.wazuh.com/current/installation-guide/wazuh-indexer/index.html. [Último acceso: 02 06 2024].

[12] «Wazuh dashboard - Installation guide - Wazuh documentation,» Wazuh documentation, [En línea]. Available:

https://documentation.wazuh.com/current/installation-guide/wazuh-dashboard/index.html. [Último acceso: 02 06 2024].

[13] «Wazuh server - Installation guide - Wazuh documentation,» Wazuh documentation, [En línea]. Available: https://documentation.wazuh.com/current/installation-guide/wazuh-server/index.html. [Último acceso: 02 04 2024].

[14] «Architecture - Getting started with Wazuh - Wazuh documentation,» Wazuh documentation, [En línea]. Available: https://documentation.wazuh.com/current/getting-started/architecture.html. [Último acceso: 02 04 2024].

[15] «Installing the Wazuh indexer step by step - Wazuh indexer,» Wazuh documentation, [En línea]. Available: https://documentation.wazuh.com/current/installation-guide/wazuh-indexer/step-by-step.html. [Último acceso: 02 04 2024].

[16] «Installing the Wazuh server step by step - Wazuh server,» Wazuh documentation, [En línea]. Available: https://documentation.wazuh.com/current/installation-guide/wazuh-server/step-by-step.html. [Último acceso: 02 06 2024].

[17] «Installing the Wazuh dashboard step by step - Wazuh dashboard,» Wazuh documentation, [En línea]. Available: https://documentation.wazuh.com/current/installation-guide/wazuh-dashboard/step-by-step.html. [Último acceso: 02 06 2024].

[18] «¿Qué es un balanceador de carga?,» Medium, [En línea]. Available: https://medium.com/@diego.coder/qu%C3%A9-es-un-balanceador-de-carga-6ca76a4b123e. [Último acceso: 02 06 2024].

[19] «nginx: *Linux* packages,» nginx , [En línea]. Available: https://nginx.org/en/linux_packages.html. [Último acceso: 02 06 2024].

[20] «Agents connections - Wazuh server cluster - Wazuh documentation,» Wazuh documentation, [En línea]. Available: https://documentation.wazuh.com/current/user-manual/manager/configuring-cluster/advanced-settings.html. [Último acceso: 02 06 2024].

[21] «TCP Health Checks | NGINX Documentation,» NGINX Documentation, [En línea]. Available: https://docs.nginx.com/nginx/admin-guide/load-balancer/tcp-health-check/. [Último acceso: 02 06 2024].

[22] «TCP and UDP Load Balancing | NGINX Documentation,» NGINX Documentation, [En línea]. Available: https://docs.nginx.com/nginx/admin-guide/load-balancer/tcp-udp-load-balancer/. [Último acceso: 02 06 2024].

[23] «Get *Kali* | *Kali Linux*,» *Kali Linux*, [En línea]. Available: https://www.kali.org/get-kali/#kali-installer-images. [Último acceso: 02 06 2024].

[24] «Updating *Kali* | *Kali Linux* Documentation,» *Kali Linux*, [En línea]. Available: https://www.kali.org/docs/general-use/updating-kali/. [Último acceso: 02 06 2024].

[25] «ATT&CK,» MITRE | ATTC&CK, [En línea]. Available: https://attack.mitre.org/. [Último acceso: 03 06 2024].