



CIPFP AUSIÀS MARCH
CENTRE INTEGRAT PÚBLIC
DE FORMACIÓ PROFESSIONAL

Despliegue, administración y securización de una infraestructura con un servidor Wazuh multi-node

David López Saorín

Grado Superior de Administración de Sistemas Informáticos en red

Tutores:

- **Raül V. Lerma-Blasco - Ausiàs**
- **Alfredo Marco Moreno - IIS La Fe**
- **Javier Ripoll Esteve - IIS La Fe**
- **Valentín Lacuesta Melero - Ausiàs**

Curso 2023 - 2024

No importa qué tan difícil o imposible sea,
no pierdas de vista tu objetivo.

- Monkey D. Luffy.

Agradecimientos

Antes de comenzar, me gustaría agradecer a todas las personas que han formado parte de mi proyecto y me han ayudado a hacerlo realidad.

Primero que nada a mi familia, que me han estado apoyando y motivando cada día desde que empecé mi trayectoria en informática para lograr lo que a día de hoy he conseguido.

También a unas de las partes más importantes del proyecto, mis tutores y compañeros, porque sin la ayuda de Alfredo Marco y Javier Ripoll, por parte del IIS La Fe, no habría podido adquirir los conocimientos técnicos suficientes para desplegar la infraestructura, y por otra parte a todos mis tutores del Ausias, sobretodo a Raül V. Lerma-Blasco que fue mi guía además de realizar el seguimiento de mi trabajo y Valentín Lacuesta Melero me proporcionó los recursos necesarios para el despliegue de la infraestructura.

Y por último, y no menos importante, a todos mis compañeros del curso con los que he pasado 2 años aprendiendo y compartiendo experiencias.

¡Muchas gracias a todos! Todos habéis sido muy importantes para el desarrollo de mi proyecto.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Despliegue, administración y securización de una infraestructura con un servidor Wazuh multi-node</i>
Nombre del autor:	<i>David López Saorín</i>
Nombre del consultor:	<i>Raül V. Lerma-Blasco - Ausiàs Alfredo Marco Moreno - IIS La Fe Javier Ripoll Esteve - IIS La Fe Valentin Lacuesta Melero - Ausiàs</i>
Fecha de entrega:(mm/aaaa)	<i>06/2024</i>
Titulación:	<i>Grado Superior en Administración de Sistemas Informáticos en red</i>
Área del trabajo final:	<i>TFGS - Seguridad Informática</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave:	<i>Indexador, XDR, SIEM, Multi-node</i>
Resumen del trabajo:	
<p>El objetivo del proyecto es desplegar una infraestructura securizada en un entorno de virtualización en la nube con Proxmox VE. De esta manera, se logra una facilidad de acceso inmensa ya que si la infraestructura estuviera desplegada en un servidor virtual local de la propia máquina física costaría desplegarla el doble de tiempo. La razón por la cuál he elegido Wazuh para securizar el sistema es porque es el SIEM Open Source por excelencia hoy en día. La idea principal es montar un escenario donde se desplegarán 4 servidores principales, uno de ellos será el propio Wazuh Server, y una vez desplegados serán monitorizados con Wazuh mediante la instalación de un agente por cada servidor. Wazuh es un XDR y SIEM capaz de recolectar cantidades ingentes y de calidad de información de los clientes monitorizados, y no solo eso, si se configura correctamente es capaz de ejercer de active-responser, es decir, se pueden establecer triggers para encadenar acciones activas como puede ser el bloqueo por IP para evitar ataques de Fuerza Bruta. El resultado final de la infraestructura es un sistema multi funcional con acceso remoto centralizado y monitorizado/securizado con Wazuh y fácil de administrar para la organización donde se decida instalar.</p>	

Abstract:

The objective of the project is to deploy a secure infrastructure in a cloud virtualization environment using Proxmox VE. This approach offers immense ease of access since if the infrastructure were deployed on a local virtual server on the physical machine itself, it would have taken twice the time to deploy. The reason I have chosen Wazuh to secure the system is because it is the quintessential open-source SIEM today. The main idea is to set up a scenario where 4 main servers will be deployed, one of them being the Wazuh Server itself. Once deployed, they will be monitored with Wazuh by installing an agent on each server. Wazuh is an XDR and SIEM capable of collecting large amounts of high-quality information from the monitored clients. Additionally, if configured correctly, it can act as an active responder, meaning triggers can be set up to chain active actions such as IP blocking to prevent brute force attacks. The final result of the infrastructure is a multifunctional system with centralized remote access, monitored and secured with Wazuh, and easy to manage for the organization where it is installed.

Índice

1. Introducción.....	13
1.1 Contexto y justificación de trabajo.....	14
1.2 Objetivos del trabajo.....	15
1.3 Enfoque y método seguido.....	15
2. Gestión y presupuesto del proyecto.....	16
2.1 Metodología.....	16
2.2 Planificación del proyecto.....	16
2.3 Presupuesto.....	17
3. Descripción y diseño del proyecto.....	19
3.1 Descripción.....	19
3.2 Esquema de red IMPEL-DOWN.....	22
3.3 Funcionamiento.....	23
3.4 Información de recursos de la red.....	23
3.5 Configuración Hardware de los servidores.....	24
4. Arquitectura y funcionamiento de Wazuh.....	27
4.1 Arquitectura.....	27
4.2 Distribución de puertos.....	29
4.3 Comunicación agente - servidor.....	30
4.4 Comunicación servidor - indexador.....	30
4.5 Casos de implementación Wazuh.....	30
Despliegue de IMPEL-DOWN.....	31
5. Desplegando Clúster de Indexadores.....	31
5.1 Descargar script y configuración de certificados.....	31
5.2 Configuración de certificados.....	31
5.3 Ejecutar script de certificados.....	32
5.4 Comprimir certificados.....	32
5.5 Copiar certificados en remoto a los nodos.....	32
5.6 Instalación de dependencias.....	33
5.7 Añadir repositorios de Wazuh.....	33
5.8 Instalando wazuh-indexer.....	33
5.9 Configurando los indexadores.....	34
5.10 Desplegando certificados.....	36
5.11 Iniciar servicios.....	38
5.12 Inicializar el Clúster de indexadores.....	38
5.13 Verificar el funcionamiento del Clúster.....	38
6. Desplegando Clúster de Servidores.....	40
6.1 Añadir repositorios de Wazuh.....	40
6.2 Instalando wazuh-manager.....	40
6.3 Iniciar servicios.....	40
6.4 Solución de error en PROXMOX: Wazuh-analysisd.....	42

6.5 Instalar Filebeat.....	43
6.6 Configurar Filebeat.....	43
6.7 Crear almacén de claves seguro.....	43
6.8 Configurar credenciales predeterminadas.....	44
6.9 Descargar plantilla de alertas.....	44
6.10 Descargar el módulo de Wazuh para Filebeat.....	44
6.11 Desplegando certificados.....	45
6.12 Iniciar el servicio de Filebeat.....	46
6.13 Verificar el estado de Filebeat.....	46
6.14 Inicializar el Clúster de servidores.....	48
6.15 Verificar el funcionamiento del Clúster.....	50
7. Desplegando la Dashboard de Wazuh.....	51
7.1 Instalación de dependencias.....	51
7.2 Añadir repositorios de Wazuh.....	51
7.3 Instalando wazuh-dashboard.....	51
7.4 Configurando wazuh-dashboard.....	52
7.5 Desplegando certificados.....	52
7.6 Iniciar servicios.....	53
7.7 Apuntar al Master Node.....	53
7.8 Accediendo a la interfaz web de Wazuh.....	53
7.9 Cambiar contraseña web de admin.....	53
8. Desplegando agente de Wazuh.....	54
8.1 Añadir agente.....	54
8.2 Desplegar agente.....	54
9. Generación de alertas via Gmail de nivel 12.....	57
9.1 Instalación de dependencias.....	57
9.2 Configuración de “main.cf”.....	57
9.3 Crear los ficheros de autenticación y configurar permisos.....	57
9.4 Verificar el funcionamiento de Postfix.....	58
9.5 Configurar ossec.conf.....	59
10. Reglas customizadas.....	60
10.1 Introducción y rutas de las reglas.....	60
11. Eliminación de logs antiguos con CRON.....	61
11.1 Introducción.....	61
11.2 Sintaxis de CRON.....	61
11.3 Tarea programada.....	62
12. Breve introducción a Host Bastión.....	63
12.1 Introduciendo Apache Guacamole.....	63
13. Configuración de Kumacamole.....	64
13.1 Configuración de disco de Kumacamole.....	64
13.2 Configuración de red de Kumacamole.....	65
13.3 Configuración de “/etc/hosts”.....	66
14. Configuración previa de Apache Guacamole.....	66
14.1 Instalación de Docker en Ubuntu.....	66

15. Desplegando Apache Guacamole.....	67
15.1 Instalación y configuración de Guacamole.....	67
15.2 Iniciar sesión en Apache Guacamole.....	70
15.3 Administración de Apache Guacamole.....	71
15.4 Configuración del servicio XRDP.....	75
16. Acceder remotamente a MagellanW-D-Aashboard.....	76
16.1 Iniciar sesión xRDP.....	76
16.2 Monitorizar inicios de sesión.....	77
17. Breve introducción a LDAP.....	78
18. Desplegando dominio de LDAP.....	78
18.1 Configuración de red.....	78
18.2 Configuración de hostname.....	79
18.3 Instalación de dependencias.....	79
18.4 Verificación del dominio.....	80
18.5 Historia y estructura de LDAP.....	81
18.6 Configuración del servicio de directorio.....	81
18.7 Generando la estructura de dominio.....	82
18.8 Instalar administrador gráfico de LDAP (LAM).....	86
19. Unión de Kumacamole al dominio.....	86
19.1 Configuración de hostname.....	86
19.2 Configuración de red.....	87
19.3 Instalación del software.....	87
20. Revisión de la configuración.....	90
20.1 Fichero “/etc/ldap.conf”.....	90
20.2 Fichero “/etc/nsswitch.conf”.....	91
20.3 Comando “pam-auth-update”.....	91
20.4 Iniciar sesión gráfica.....	92
21. Breve introducción a Remote Backups Server.....	94
21.1 Funcionamiento de Remote Backups Server.....	94
22. Desplegando Remote Backups Server.....	95
22.1 Configuración de disco.....	95
22.2 Configuración de red.....	96
23. Configurando accesos usuario local rsync.....	97
23.1 Generación de clave rsa.....	97
23.2 Copiar clave pública al servidor remoto.....	98
23.3 Configurar auto-conexiones al servidor remoto.....	98
23.4 Configurar acl para usuario impeladmin.....	98
23.5 Automatización de Jobs con tareas CRON.....	99
24. Conclusiones.....	104
25. Bibliografía.....	105
26. Anexos.....	106

1. Introducción

El mundo está cambiando cada vez más rápido en avances tecnológicos, desde la inteligencia artificial hasta el Internet de las cosas (IoT). Este rápido avance realmente aporta innumerables beneficios, incluida una mayor eficiencia, nuevas oportunidades comerciales y una mejor calidad de vida. Sin embargo, también se puede aprovechar para hacer el mal, lo que conlleva un desafío contra la ciberseguridad a todos los niveles.

Esta innovación tecnológica cambia gran parte de los aspectos de nuestras vidas. Las empresas dependen cada vez más de la computación en la nube, los big data y las soluciones de redes de alta velocidad. Estas tecnologías permiten a las empresas no sólo trabajar de manera más eficiente, sino también brindar servicios innovadores a sus clientes. Sin embargo, esta revolución también significa una mayor superficie de ataque para los ciberdelincuentes como antes he mencionado.

Cuanto más avanza la tecnología, los ciberataques se vuelven más sofisticados y frecuentes. Las infraestructuras modernas se han vuelto cada vez más complejas y, con ello, surge el desafío de proteger nuestros datos. Las organizaciones deben hacer frente a una amplia gama de amenazas, desde malware y ransomware hasta ataques de phishing, filtraciones de datos, reverse shells y backdoorings entre una infinita gama de vectores de ataque.

Dicho esto, lo que ofrezco en este proyecto es crear una solución de ciberseguridad mediante una herramienta tope de gama y Open Source del mercado actual. Wazuh es una plataforma de seguridad perimetral diseñada para monitoreo y análisis de seguridad en tiempo real. Wazuh es una plataforma que proporciona capacidades avanzadas de detección de intrusiones, monitoreo de integridad, análisis de registros y respuesta a incidentes. Se integra fácilmente con una variedad de infraestructuras, tanto locales como en la nube, brindando a las organizaciones una visibilidad completa de la seguridad.

Wazuh permite a las empresas monitorear continuamente su infraestructura y detectar y remediar instantáneamente cambios inesperados o no autorizados gracias a su velocidad de notificación. Esta plataforma recopila y analiza registros de varios sistemas y aplicaciones y proporciona información detallada sobre eventos de seguridad. Esto ayuda a detectar patrones o comportamientos inusuales que pueden indicar un ataque.

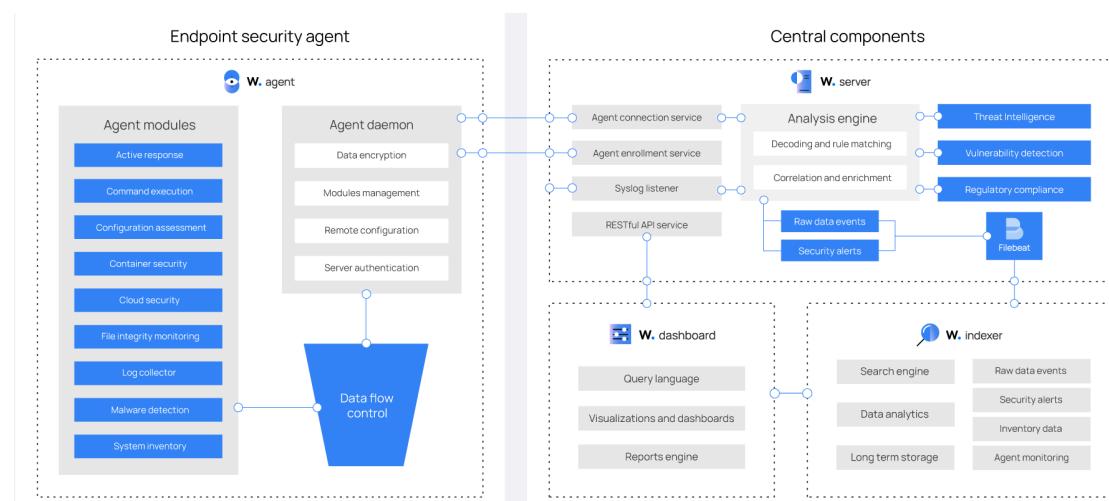
La idea final del proyecto es recrear un infraestructura securizada que contendrá un host bastión donde habrá corriendo un servicio por el que accede el administrador de manera remota al resto de la infraestructura, la cual contará con un servidor LDAP, un servidor de backups remotos y un cluster de servidores Wazuh.

1.1 Contexto y justificación del trabajo

Considerando que hasta hace poco tiempo, la seguridad en las Tecnologías de la Información y la Comunicación (TIC) no recibía la atención que merece, si realizáramos una encuesta sobre las herramientas empleadas para monitorizar la seguridad de los sistemas, muchos responderían que no disponen de este tipo de herramientas. Las razones para esto son variadas, pero las más destacadas incluyen el coste y el desconocimiento sobre cómo implementar y utilizar estas herramientas de manera efectiva. Es crucial entender que interpretar la información proporcionada por estas herramientas no es tarea sencilla y, generalmente, se requiere la intervención de expertos en seguridad TIC para analizar y gestionar los resultados de manera adecuada.

Las herramientas diseñadas para monitorizar y gestionar la seguridad en las TIC desempeñan un papel vital en la detección de amenazas, vulnerabilidades, configuraciones erróneas y brechas de seguridad. Este tipo de soluciones se conocen comúnmente como Sistemas de Gestión de Información y Eventos de Seguridad (SIEM, por sus siglas en inglés). El mercado ofrece un amplio abanico de herramientas de seguridad, cada una con sus propias características y enfoques. Entre todas ellas, destaca una herramienta de gran interés llamada "Wazuh".

"Wazuh" es una plataforma de código abierto que proporciona capacidades avanzadas para la monitorización de la seguridad, específicamente orientada a la seguridad de los endpoints, como servidores, mediante el uso de agentes ligeros. A diferencia de otros SIEM que pueden ser más generalistas, Wazuh se especializa en ofrecer una solución robusta y eficiente para la seguridad de los dispositivos finales. Esto incluye la detección de intrusiones, la monitorización de la integridad de archivos, la auditoría de la configuración de seguridad y el análisis de logs.



1.2 Objetivos del trabajo

Objetivos generales:

- Desplegar un servidor Wazuh multi-node
- Desplegar un Apache Guacamole
- Desplegar un servidor LDAP
- Desplegar un servidor de Backups Remoto
- Monitorizar cada uno de los servidores endpoint
- Conocer el funcionamiento de la infraestructura al completo

Objetivos específicos:

- Conocer la arquitectura agente-servidor
- Crear reglas avanzadas customizadas en el servidor Wazuh
- Corregir problemas concretos de Proxmox con Wazuh
- Administrar todos los nodos de manera remota con Guacamole
- Crear procesos en CRON para ejecutar rsync en remoto y de esta manera hacer copias de seguridad incrementales y remotas.

1.3 Enfoque y método seguido

Mi enfoque en el proyecto ha sido la constante investigación sobre una tecnología (Wazuh) que, en mi caso, es nueva y resulta ser muy poderosa a la hora de analizar cantidades enormes de información. Y gracias a esta herramienta se logrará instalar el agente en los servidores de la red mediante unas variables que se aplican junto a la instalación del agente y monitorizarse mediante reglas.

Cabe recalcar que la información que he utilizado al 70% ha sido extraída de la propia página web de Wazuh, el 30% restante ha sido obtenido de consultar en foros de Internet.

2. Gestión y presupuesto del proyecto

2.1 Metodología

La metodología llevada a cabo en mi proyecto ha sido mediante el esfuerzo del día a día, la prueba y error y la más importante, la investigación constante para documentarse sobre la tecnología elegida, lograr el despliegue y perfecto funcionamiento de una gran estructura mediante el mínimo de recursos disponibles. A lo largo de mi trayectoria como informático he descubierto que la clave del buen funcionamiento no está en la alta y costosa tecnología de la que uno dispone, si no de construir mucho y bien con lo justo y necesario de lo que se dispone. De esta manera se logra aprender a pensar y administrar y suministrar los recursos disponibles tanto físicos como virtuales.

2.2 Planificación del proyecto

La idea principal era crear un servidor de Wazuh “Dockerizado” en una máquina virtual que haría de host y unir dos o tres clientes con los que realizar pruebas de auditorías contra esos clientes con la creencia de que podría crear respuestas activas (active-responses) para neutralizar los ataques.

No fue hasta después de un mes en las prácticas, que después de todas las tareas realizadas en el IIS La Fe y darme cuenta de lo que realmente podría llegar a desplegar decidí modificar todo mi plan de despliegue de infraestructura.

Mi nueva idea es mucho más grande y ambiciosa que la anterior:

- Consiste en el despliegue de un Host Bastión, el cuál haría de nodo “portón” por el que entrar al resto de la infraestructura, de esta manera interpongo una barrera más entre el exterior y mi intranet. Además se conseguirá centralizar las conexiones de manera cifrada y segura de los nodos de la red con Apache Guacamole.
- El segundo paso es el despliegue de un servidor LDAP para ayudar al futuro administrador del sistema al que venda la infraestructura a centralizar las autenticaciones de los usuarios.
- El tercer paso consiste en la creación de un servidor de backups remoto que de manera semanal y en algunos casos diaria realizará copias de seguridad de rutas de ficheros importantes de todos los servidores de la infraestructura.
- Y por último deslegaré un servidor Wazuh multi-node para la securización de la infraestructura.

Esta sería una visión aproximada de el tiempo empleado en cada fase:

Actividad	Porcentaje
Investigación preliminar	25%
Desarrollo del prototipo	10%
Implementación final	40%
Pruebas y validación	15%
Evaluación de resultados	10%
Elaboración del informe final	10%

2.3 Presupuesto

Para el desarrollo de este proyecto los recursos han sido proporcionados por la propia organización, la infraestructura física ya estaba disponible. Por lo que los gastos materiales han sido nulos.

Un presupuesto partiendo desde el inicio del servidor físico necesitaría de los siguientes elementos:

Componente	Nombre	Cantidad	Precio
Chasis	Dell R740xd 24SFF	1	Incluido
Procesador	Intel Xeon Gold 6240R (24C 35.75M Cache 2.40 GHz)	2	€4 018
RAM	32GB DDR4 RDIMM 2666MHz	1	€51
Módulo de control remoto	iDRAC 9 Express	1	Incluido
Fuente de alimentación	Power supply Dell 1100w	2	€124
Adaptador LAN	ports 1GB Base-T RJ-45 NDC	4	Incluido
Sistema Operativo	No	0	No incluido
Rieles para Rack	Rack mount kit 19"	kit	€69
Bahías	SSD 1.92TB SATA 2.5" + Tray Caddy	1	€359
Total			€4621

Este proyecto que incluye el desempeño de la idea, la planificación, estructuración, la investigación de una nueva tecnología en el mercado y el despliegue de toda la infraestructura ha sido desarrollado por una única persona. Luego, el presupuesto incluirá la creación del proyecto y su diseño, el tiempo de dedicación del técnico que lo desarrolla y los posibles desplazamientos al lugar físico del montaje del proyecto si fuera necesario para su instalación y puesta en marcha.

Presupuesto final

Unidades	Importe/unidad	Descripción	Precio
1	€2.500	Estudio y diseño del proyecto	€2.500
136	€10.50	Horas dedicación técnica	€1.428
10	€13	Horas desplazamiento técnico (*)	€130
Presupuesto total			€4058

(*) Este número de horas incluye 10 visitas a la empresa, si por necesidad del cliente el número de desplazamientos tuviera que aumentar se cobrará a razón de €13 por visita/hora de desplazamiento. Del mismo modo en el caso de necesitar menos visitas se descontará a razón de €13 por visita/hora.

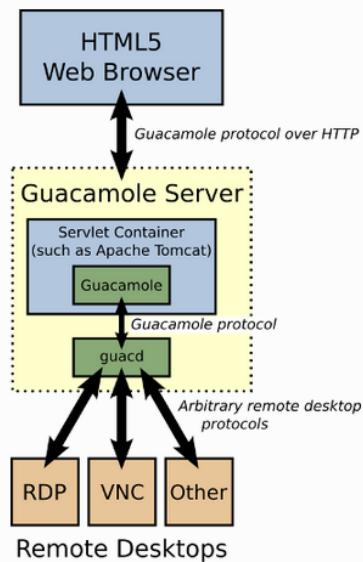
3. Descripción y diseño del proyecto

3.1 Descripción

La idea principal de este proyecto es desplegar un entorno de servidores virtualizados multifuncional con unos mínimos, como la autenticación centralizada de usuarios mediante un servidor LDAP, un servidor de backups remoto para que de manera semanal y algunos casos cada 2 o 3 días a la semana hayan copias de seguridad incrementales de las rutas de ficheros más importantes de cada servidor, más adelante las nombraré. También, habrá un equipo entre la Extranet y la Intranet que ejercerá de Host Bastión, y será el encargado de centralizar el resto de conexiones de la red mediante un proceso que correrá bajo la tecnología de Docker, Apache Guacamole. Guacamole es un proyecto de Apache, y gracias a él se logrará utilizar los protocolos adecuados para abrir sesiones tanto gráficas como por consola remotamente. Pese a algunos inconvenientes que han surgido en Apache Guacamole se ha logrado programar una configuración óptima para el entorno en cuestión.

Dado que Guacamole es una API, no solo una aplicación web, los componentes y bibliotecas centrales proporcionados por el proyecto Guacamole se pueden utilizar para añadir funciones de acceso remoto HTML5 a una aplicación existente. Además, Apache Guacamole es muy ligero y versátil, logra comprimir los archivos de imágenes que se generan y viajan por la red de punto a punto de manera cifrada gracias a HTTPS.

Aunque no es del todo cierto, la mini infraestructura que trae la imagen de Docker que se ha instalado tiene múltiples servicios:



Guacamole no es una aplicación web autónoma y está compuesta por muchas partes. La aplicación web está diseñada para ser simple y minimalista, con la mayoría del trabajo pesado realizado por componentes de nivel inferior.

Los administradores se conectarán al servidor Guacamole con un navegador web. El cliente de Guacamole, escrito en JavaScript, se sirve a los usuarios por un servidor web dentro del propio servidor Guacamole. Una vez cargado, este cliente se conecta de nuevo al servidor a través de HTTP usando el protocolo Guacamole.

La aplicación web desplegada en el servidor Guacamole lee el protocolo Guacamole y lo reenvía a guacd, el proxy nativo de Guacamole. Este proxy es el que realmente interpreta el contenido del protocolo Guacamole, conectándose a cualquier número de servidores de escritorio remoto con el nombre del usuario.

La combinación del protocolo Guacamole y guacd proporciona agnosticismo de protocolo: ni el cliente de Guacamole ni la aplicación web necesitan saber qué protocolo de escritorio remoto se está utilizando.

Protocolo Guacamole

La aplicación web no entiende ningún protocolo de escritorio remoto en absoluto. No contiene soporte para VNC, RDP u otro protocolo soportado por el stack de Guacamole. En realidad, solo entiende el protocolo Guacamole, que es un protocolo para renderizado remoto de pantallas y transporte de eventos. Aunque un protocolo con esas propiedades naturalmente tendría las mismas capacidades que un protocolo de escritorio remoto, los principios de diseño detrás de un protocolo de escritorio remoto y el protocolo Guacamole son diferentes: el protocolo Guacamole no está destinado a implementar las características de un entorno de escritorio específico, si no que implementa una gran variedad de protocolos de escritorio remoto.

Guacd

guacd es el corazón de Guacamole que carga dinámicamente el soporte para protocolos de escritorio remoto (llamados "plugins de cliente") y los conecta a escritorios remotos basándose en las instrucciones recibidas de la aplicación web.

guacd es un proceso daemon que se instala junto con Guacamole y se ejecuta en segundo plano, escuchando conexiones TCP desde la aplicación web. guacd tampoco entiende ningún protocolo de escritorio remoto específico, sino que implementa solo lo necesario del protocolo Guacamole para determinar qué soporte de protocolo debe cargarse y qué argumentos deben pasarse a este.

Aplicación Web

La parte de Guacamole con la que un usuario realmente interactúa es la aplicación web. La aplicación web, como se mencionó antes, no implementa ningún protocolo de escritorio remoto. Depende de guacd y no implementa nada más que una elegante interfaz web y una capa de autenticación.

Y por último, habrá un servidor Wazuh multi nodo, el cuál sirve para securizar la infraestructura mediante el despliegue de un agente por cada uno de los servidores desplegados. El objetivo, ya mencionado anteriormente, es la constante recolección de logs de los servidores para más tarde analizarlos y estudiar si existiese la posibilidad de un ataque que está siendo llevado a cabo para acceder al sistema o de lo contrario, ya está dentro el atacante.

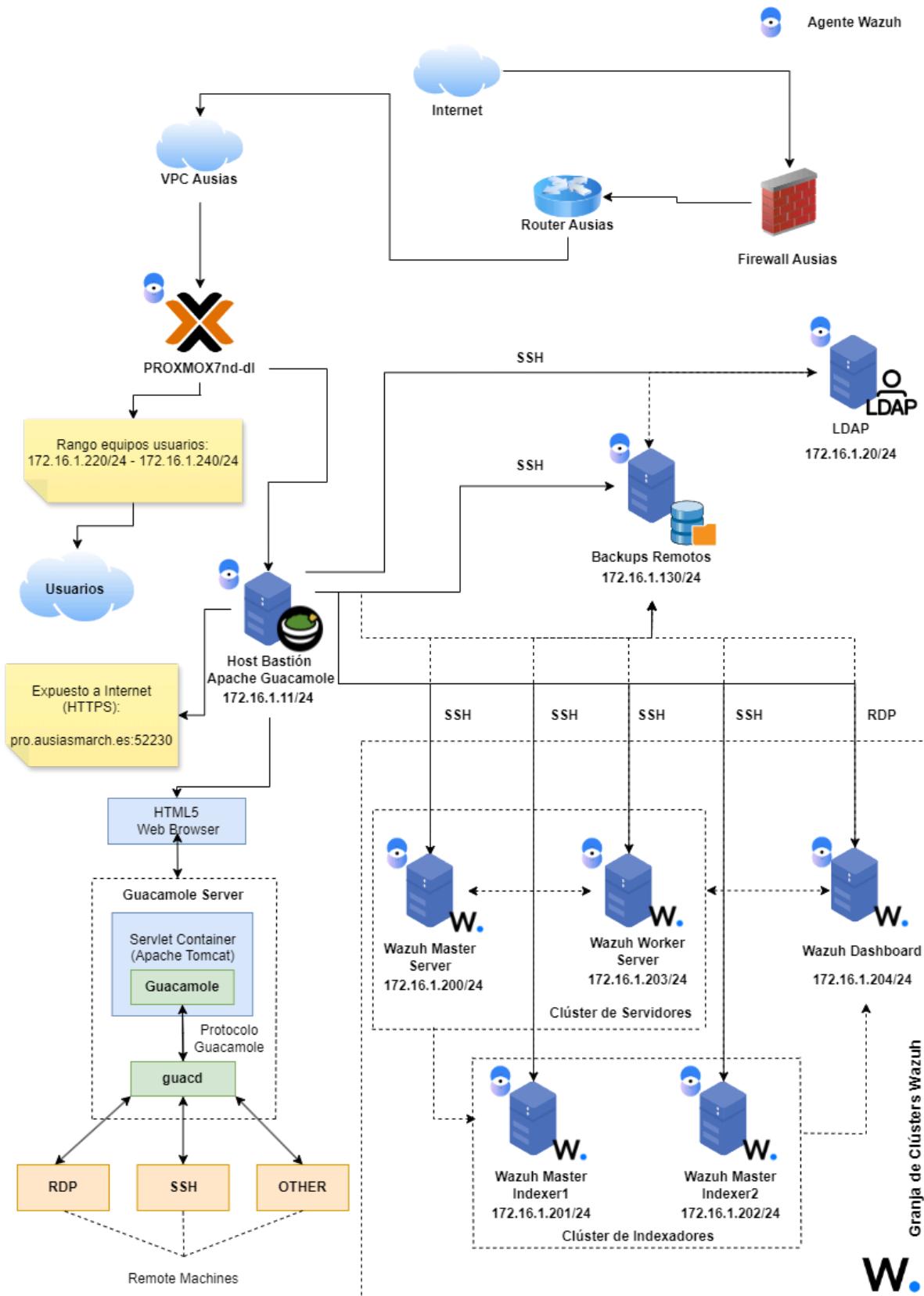
Wazuh da la posibilidad de realizar la instalación nativa de dos maneras diferentes: Single-node y Multi-node. La diferencia principal es que en Single-node todo el sistema estará centralizado en una única máquina host, y son varias las ventajas que este modo de operabilidad ofrece, algunas de ellas son el propio despliegue es más fácil de realizar, no hay que replicar certificados y contraseñas entre los diferentes nodo y una fácil gestión y acceso del sistema, mientras que Multi-node es bastante más complejo de desplegar, debido a que los componentes centrales de Wazuh están divididos entre los diferentes nodos hay que generar, autofirmar y transferir los certificados por cada uno de ellos, y posteriormente eliminarlos, almacenar cada contraseña cuidadosamente y siendo cada una de ellas diferente a la anterior, y lógicamente que exista conexión entre cada uno de ellos.

Sin embargo, la creación de un clúster de servidores Wazuh trae más ventajas que inconvenientes, ya que con el solo hecho de que se consigue la alta disponibilidad para conseguir la redundancia de operabilidad y de datos es factible. Además en el caso de atacar a uno de los nodos y que el atacante logre el dominio total del nodo no conseguiría dominio total sobre todo el servidor Wazuh, logrando así mayor seguridad.

Y por último, para presentar el servidor de Backups Remoto, la idea principal era crear copias de seguridad completas e incrementales de cada uno de los servidores para obtener la mayor cantidad de seguridad posible. Pero tras un largo tiempo de pensar sobre cómo podría optimizar el espacio usado para las copias de seguridad, porque estoy muy limitado, en cuanto a almacenamiento respecta, se me ocurrió crear los backups solamente de las rutas de ficheros más esenciales de cada servidor. De esta manera, mediante el comando rsync y configurando bien CRON para que no se solapen la ejecución de unos backups con otros, no solo se logra una reducción en la necesidad de espacio, si no que se optimiza enormemente el tráfico de red.

Los backups se realizan mediante la creación de un usuario desprivilegiado en la parte del servidor de backups remoto y compartiendo la clave pública de cada usuario de cada servidor que ejecutará rsync con los permisos cuidadosamente configurados.

3.2 Esquema de red IMPEL-DOWN



3.3 Funcionamiento

Cuando el administrador de la red se conecte a través del puerto expuesto por HTTPS entrará en el visor Web de Apache Guacamole. Este facilitará la administración de los demás nodos, como se puede apreciar en el esquema de red, Apache guacamole tiene conexión a todos los nodos excepto al propio Host Bastión y al nodo Proxmox7nd-dl porque considero un fallo de seguridad que el propio Guacamole pueda acceder a su propia máquina y cambiar los parámetros necesarios para entrar al resto de la infraestructura sin necesidad de aportar contraseña.

Se puede apreciar como el puerto SSH está abierto para 6/7 máquinas y en 1/7 es el protocolo xRDP. En esas 6 máquinas están corriendo sistemas operativos con entorno por consola, es decir, hay que apuntar al puerto SSH (22) que está abierto en ellas (más adelante lo explico), y en la única máquina donde está abierto el puerto RDP (3389) la conexión es, obviamente, gráfica, y es más tediosa de configurar (Más adelante lo explico).

3.4 Información de recursos de la red

Recursos en PROXMOX disponibles

vCPU	RAM	Almacenamiento en disco
32	28 GB	284 GB

Recursos asignados a cada máquina virtual o LXC

Máquina	vCPU	RAM	Almacenamiento en disco	S.O
Host Bastión + Guacamole	4 vCPU	5 GB	20 GB	Ubuntu Desktop 22.04
Wazuh-indexer1	4 vCPU	4 GB	20 GB	Ubuntu Server 22.04
Wazuh-indexer2	4 vCPU	4 GB	20 GB	Ubuntu Server 22.04
Wazuh-master	2 vCPU	2 GB	15 GB	LXC-Ubuntu Server 22.04
Wazuh-worker	2 vCPU	2 GB	10 GB	LXC-Ubuntu Server 22.04
Wazuh-dashboard	4 vCPU	4 GB	27 GB	Linux Mint 21.3
LDAP Server	2 vCPU	2 GB	15 GB	Ubuntu Server 22.04

Backup Server	4 vCPU	3 GB	80 GB	Ubuntu Server 22.04
TOTAL	26 vCPU	26 GB	207 GB	

3.5 Configuración Hardware de los servidores

Clúster de servidores Wazuh:

magellanws (Proxmox id: 200)

Memory	2.00 GiB
Swap	2.00 GiB
Cores	2
Root Disk	local:200/vm-200-disk-1.raw,size=15G

magellanww (Proxmox id: 203)

Add ▾	Edit	Remove	Volume Action ▾	Revert
Memory	2.00 GiB			
Swap	2.00 GiB			
Cores	2			
Root Disk	local:203/vm-203-disk-0.raw,size=10G			

Clúster de indexadores Wazuh:

MagellanW-I1 (Proxmox id: 201)

Memory	4.00 GiB
Processors	4 (1 sockets, 4 cores)
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
CD/DVD Drive (ide2)	none,media=cdrom
Hard Disk (scsi0)	local:201/vm-201-disk-0.qcow2,iothread=1,size=20G
Network Device (net0)	virtio=22:A6:17:E5:EA:C6,bridge=vmbr1,firewall=1

MagellanW-I2 (Proxmox id: 202)

Memory	4.00 GiB
Processors	4 (1 sockets, 4 cores)
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
CD/DVD Drive (ide2)	none,media=cdrom
Hard Disk (scsi0)	local:202/vm-202-disk-0.qcow2,iothread=1,size=20G
Network Device (net0)	virtio=8A:5A:9C:A3:43:56,bridge=vmbr1,firewall=1

Dashboard Wazuh:**MagellanW-D-Ashboard (Proxmox id: 204)**

Memory	4.00 GiB
Processors	4 (1 sockets, 4 cores)
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
CD/DVD Drive (ide2)	none,media=cdrom
Hard Disk (scsi0)	local:204/vm-204-disk-0.qcow2,iothread=1,size=27G
Network Device (net0)	virtio=9E:23:0D:A1:49:A0,bridge=vmbr1,firewall=1

Host bastión:**Kumacamole (Proxmox id: 110)**

Memory	5.00 GiB
Processors	4 (1 sockets, 4 cores)
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
CD/DVD Drive (ide2)	none,media=cdrom
Hard Disk (scsi0)	local:110/vm-110-disk-0.qcow2,iothread=1,size=20G
Network Device (net0)	virtio=26:D2:4E:1B:07:B4,bridge=vmbr1,firewall=1

Servidor LDAP:**CrimsonLDAP (Proxmox id: 120)**

Memory	2.00 GiB
Processors	2 (1 sockets, 2 cores)
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
CD/DVD Drive (ide2)	RepShare:iso/ubuntu-22.04.3-live-server-amd64.iso,media=cdrom,size=2083390K
Hard Disk (scsi0)	local:120/vm-120-disk-0.qcow2,iothread=1,size=15G
Network Device (net0)	virtio=6A:32:B4:06:45:7C,bridge=vmbr1,firewall=1

Servidor Remote Backups:**LogPose-RB (Proxmox id: 130)**

Memory	3.00 GiB
Processors	4 (1 sockets, 4 cores)
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
CD/DVD Drive (ide2)	RepShare:iso/ubuntu-22.04.3-live-server-amd64.iso,media=cdrom,size=2083390K
Hard Disk (scsi0)	local:130/vm-130-disk-0.qcow2,iothread=1,size=80G
Network Device (net0)	virtio=EA:00:29:5E:56:77,bridge=vmbr1,firewall=1

4. Arquitectura y funcionamiento de Wazuh

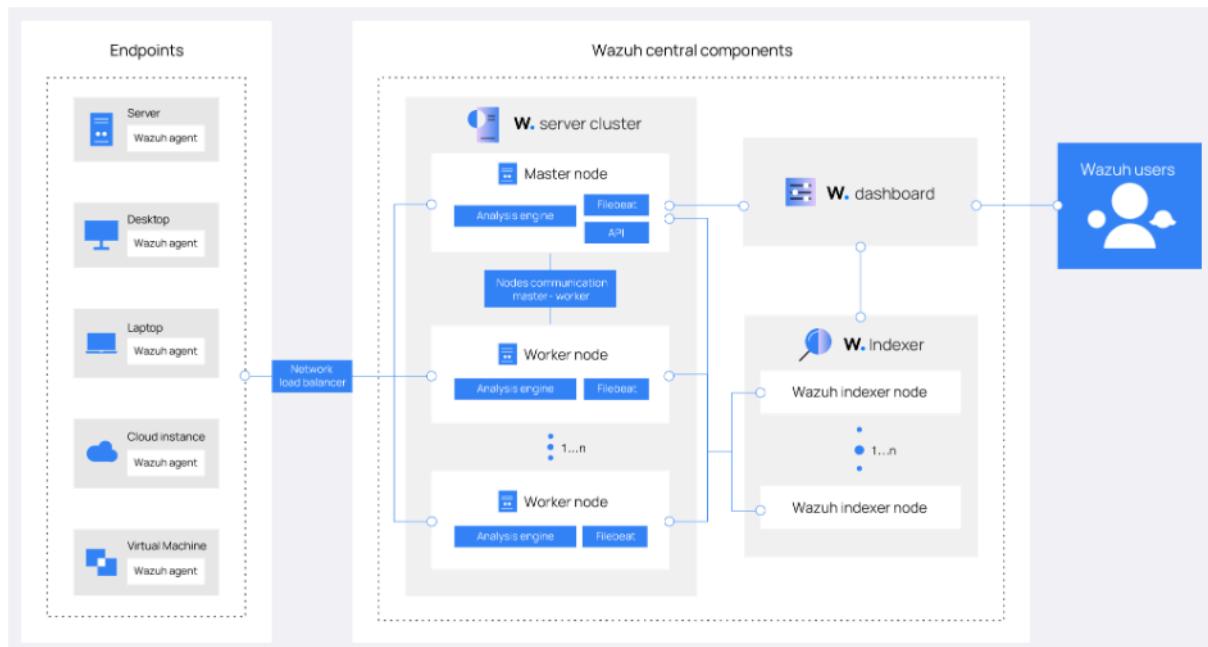
4.1 Arquitectura

La arquitectura de Wazuh se basa en agentes que envían datos de seguridad a un servidor central, el cual decodifica y analiza la información antes de pasar los resultados al indexador Wazuh para su almacenamiento. Los dispositivos sin agente, como cortafuegos y routers, pueden enviar registros a través de Syslog, SSH o API.

El clúster del indexador Wazuh, compuesto por uno o más nodos, maneja las operaciones de lectura y escritura en los índices. Un clúster de un solo nodo es adecuado para pequeñas implementaciones, mientras que los clústeres de múltiples nodos son recomendables para grandes volúmenes de datos o alta disponibilidad.

En entornos de producción, se sugiere separar el servidor Wazuh y el indexador en diferentes hosts, usando Filebeat para enviar de forma segura las alertas y eventos archivados al clúster del indexador con cifrado TLS.

El diagrama muestra la configuración de la arquitectura de Wazuh, destacando la configuración en clúster para balanceo de carga y alta disponibilidad.



Enumeración de los componentes principales de Wazuh

Componente 1: El servidor de Wazuh

El servidor Wazuh analiza los datos recibidos de los agentes Wazuh, generando alertas cuando se detectan amenazas o anomalías. También se utiliza para gestionar de forma remota la configuración de los agentes y monitorear su estado.

Componente 2: El indexador de Wazuh

El indexador Wazuh es un motor de búsqueda y análisis de texto completo altamente escalable. Este componente central de Wazuh indexa y almacena las alertas generadas por el servidor Wazuh y proporciona capacidades de búsqueda y análisis de datos en casi tiempo real.

Componente 3: La dashboard de Wazuh

Este componente central es una interfaz web flexible e intuitiva para explorar, analizar y visualizar datos de seguridad. Proporciona paneles de control preconfigurados, permitiéndote navegar sin problemas a través de la interfaz de usuario.

Con el panel de control de Wazuh, los usuarios pueden visualizar eventos de seguridad, vulnerabilidades detectadas, datos de monitoreo de integridad de archivos, resultados de evaluación de configuraciones, eventos de monitoreo de infraestructura en la nube y estándares de cumplimiento regulatorio.

4.2 Distribución de puertos

Se requieren varios servicios para la comunicación entre los componentes de Wazuh. Estos son los protocolos que son utilizados por los servicios:

Componente	Puerto	Protocolo	Propósito
Servidor Wazuh	1514	TCP (por defecto)	Servicio de conexión de agentes
	1514	UDP (opcional)	Servicio de conexión de agentes (deshabilitado por defecto)
	1515	TCP	Servicio de inscripción de agentes
	1516	TCP	Demonio del clúster de Wazuh
	514	UDP (por defecto)	Colector Syslog de Wazuh (deshabilitado por defecto)
	514	TCP (opcional)	Colector Syslog de Wazuh (deshabilitado por defecto)
	55000	TCP	API RESTful del servidor Wazuh
Indexador Wazuh	9200	TCP	API RESTful del indexador Wazuh
	9300 - 9400	TCP	Comunicación del clúster del indexador Wazuh
Dashboard Wazuh	443	TCP	Interfaz web de usuario de Wazuh

4.3 Comunicación agente - servidor

El agente Wazuh envía continuamente eventos al servidor Wazuh para su análisis y detección de amenazas. Para comenzar a enviar estos datos, el agente establece una conexión con el servicio del servidor para la conexión de agentes, que por defecto escucha en el puerto 1514 (esto es configurable).

El servidor Wazuh luego decodifica y verifica las reglas de los eventos recibidos, utilizando el motor de análisis. Los eventos que activan una regla se enriquecen con datos de alerta, como el ID de la regla y el nombre de la regla. Los eventos pueden almacenarse en uno o ambos de los siguientes archivos, dependiendo de si se activó o no una regla:

- El archivo `/var/ossec/logs/archives/archives.json` contiene todos los eventos, hayan activado o no una regla.
- El archivo `/var/ossec/logs/alerts/alerts.json` contiene sólo los eventos que activaron una regla con una prioridad suficientemente alta (el umbral es configurable).

El protocolo de mensajes de Wazuh utiliza cifrado AES por defecto, con 128 bits por bloque y claves de 256 bits. El cifrado Blowfish es opcional.

4.4 Comunicación servidor - indexador

El servidor Wazuh utiliza Filebeat para enviar datos de alertas y eventos al indexador Wazuh con cifrado TLS. Filebeat lee los datos de salida del servidor Wazuh y los envía al indexador Wazuh (que por defecto escucha en el puerto 9200/TCP). Una vez que los datos son indexados, se utiliza el dashboard de Wazuh para analizar y visualizar la información.

El dashboard de Wazuh consulta la API RESTful de Wazuh (que por defecto escucha en el puerto 55000/TCP en el servidor Wazuh) para mostrar información sobre la configuración y el estado del servidor Wazuh y sus agentes. También puede modificar la configuración de los agentes o del servidor a través de llamadas a la API. Esta comunicación está cifrada con TLS y autenticada con un nombre de usuario y una contraseña.

4.5 Casos de implementación Wazuh

Estos son algunas de las acciones que realiza Wazuh:

Seguridad de puntos finales	Inteligencia de amenazas	Operaciones de seguridad	Seguridad en la nube
Evaluación de configuración	Caza de amenazas	Respuesta a incidentes	Seguridad de contenedores
Detección de malware	Ánalisis de datos de registro	Cumplimiento normativo	Gestión de postura
Monitorizar integridad de archivos	Detección de vulnerabilidades	Higiene de TI	Protección de cargas de trabajo

Despliegue de IMPEL-DOWN

5. Desplegando Clúster de Indexadores

5.1 Descargar script y configuración de certificados

Primero hay que descargar el script que generará los certificados auto-firmados para encriptar las comunicaciones que se realizarán entre los componentes de Wazuh (Indexadores, servidores y dashboard), además se descargará el fichero de configuración de los nodos.

```
`# curl -sO https://packages.wazuh.com/4.7/wazuh-certs-tool.sh
# curl -sO https://packages.wazuh.com/4.7/config.yml`
```

5.2 Configuración de certificados

```
root@magellanwi1:/home/impeladmin# cat config.yml
nodes:
  # Wazuh indexer nodes
  indexer:
    - name: node-1
      ip: "172.16.1.201" # magellanwi1
    - name: node-2
      ip: "172.16.1.202" # magellanwi2
    #- 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: wazuh-1
      ip: "172.16.1.200" # magellanws
      node_type: master
    - name: wazuh-2
      ip: "172.16.1.203" # magellanww
      node_type: worker
    #- name: wazuh-3
    #  ip: "<wazuh-manager-ip>"
    #  node_type: worker

  # Wazuh dashboard nodes
  dashboard:
    - name: dashboard
      ip: "172.16.1.204" # magellanwdashboard
```

En el fichero de configuración con nombre “config.yml” hay que especificar cada uno de los nodos con su nombre seguido de su IP. En mi caso habrán **2 indexadores** (172.16.1.201 y 172.16.1.201), **2 servidores** (1 master “172.16.1.200” y 1 worker “172.16.1.203”) y **1 dashboard** (172.16.1.204).

5.3 Ejecutar script de certificados

```
root@magellanwi1:/home/impeladmin# bash ./wazuh-certs-tool.sh -A
25/04/2024 19:16:15 INFO: Admin certificates created.
25/04/2024 19:16:16 INFO: Wazuh indexer certificates created.
25/04/2024 19:16:17 INFO: Wazuh server certificates created.
25/04/2024 19:16:17 INFO: Wazuh dashboard certificates created.
```

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

Una vez tenemos configurado el fichero “**config.yml**” hay que ejecutar el script de generación de certificados descargado anteriormente, de esta manera recogerá la información del fichero de configuración y creará los nuevos certificados, los cuales habrá que ir aplicando a los componentes de Wazuh a medida que se avance en el despliegue.

Una creación exitosa de certificados se ve como en la imagen anterior.

5.4 Comprimir certificados

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

Ahora hay que comprimir la carpeta de certificados generados para copiarla en el siguiente paso al resto de los componentes.

5.5 Copiar certificados en remoto a los nodos

Cuando ya tengamos el fichero comprimido hay que replicarlo en el resto de componentes para que la comunicación encriptada sea entendida por todos los nodos. Para ello he empleado el comando “**scp**”, el cuál copia de manera segura (encriptada) y en remoto el fichero comprimido a los componentes de Wazuh.

```
`# scp wazuh-certificates.tar impeladmin@172.16.1.200:/home/impeladmin
# scp wazuh-certificates.tar impeladmin@172.16.1.202:/home/impeladmin
# scp wazuh-certificates.tar impeladmin@172.16.1.203:/home/impeladmin
# scp wazuh-certificates.tar impeladmin@172.16.1.204:/home/impeladmin`
```

Los componentes tienen el siguiente orden:

- “172.16.1.200” → magellanws
- “172.16.1.202” → magellanwi2
- “172.16.1.203” → magellanww
- “172.16.1.204” → magellanwdashboard

5.6 Instalación de dependencias

NOTA: A partir de este paso, hay que replicar los comandos en ambos indexadores en caso de que el despliegue sea multi-node.

Ejecutar el siguiente comando en caso de no tener los paquetes instalados:

```
# apt install debconf adduser procps
```

5.7 Añadir repositorios de Wazuh

Ahora añadir los repositorios de Wazuh en **todos los componentes**, aunque en mi caso solo lo haré en los indexadores por el momento, más adelante iré añadiendo los repositorios en el resto de componentes, ya que si no no se podrá instalar los componentes de Wazuh.

Instalar las herramientas necesarias:

```
# apt install gnupg apt-transport-https
```

Descargar la clave pública de GPG (GNU Privacy Guard) del repositorio de paquetes de Wazuh e importar la clave pública a un anillo de claves personalizado (wazuh.gpg) utilizando gpg, un programa que implementa el estándar OpenPGP para cifrado de datos y firmas digitales:

```
# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
```

Añadir los repositorios de Wazuh con la siguiente línea a un fichero nombrado “wazuh.list” dentro de la ruta de repositorios del sistema:

```
# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list
```

Por último actualizar los paquetes del sistema:

```
# apt update
```

5.8 Instalando wazuh-indexer

Instalar el paquete del **Indexador** de Wazuh:

```
# apt -y install wazuh-indexer
```

5.9 Configurando los indexadores

Ahora hay que editar el archivo de configuración de los indexadores: “`/etc/wazuh-indexer/opensearch.yml`”. Pero antes hago una copia del fichero original:

```
# cp /etc/wazuh-indexer/opensearch.yml /etc/wazuh-indexer/opensearch.original`
```

Los parámetros que hay que configurar son:

- **network.host**: Apunta a la dirección del mismo nodo (magellanwi1: “**172.16.1.201**” y magellanwi2: “**172.16.1.202**”) para el protocolo HTTP y tráfico de transporte. Este parámetro emplea la IP aportada como dirección pública para darse a conocer al resto de componentes.
- **node.name**: Nombre del Indexador de Wazuh (node-1 y node-2) cómo fue asignado en el fichero de configuración “**config.yml**”.
- **cluster.initial_master_nodes**: Lista de los nombres de los nodos elegibles para ser maestros. Estos nombres están definidos en el archivo “**config.yml**”. En mi caso, como tengo dos nodos tengo que especificar ambos. Si hubiese más nodos habría que especificarlos.
- **discovery.seed_hosts**: Lista de las direcciones de los nodos elegibles para ser maestros. En mi caso, como la arquitectura es multi-node debo descomentar la línea de “**node-2-ip**” y asignar la IP de magellanwi2 “172.16.1.202”.
- **plugins.security.nodes_dn**: Lista de los Nombres Distintivos (Distinguished Names) de los certificados de todos los nodos del clúster del índice de Wazuh. He descomentado la segunda línea, y aunque no sean los parámetro de España no hay problema, no afecta en el rendimiento de Wazuh.

Configuración MagellanWI1:

```

network.host: "172.16.1.201"
node.name: "node-1"
cluster.initial_master_nodes:
- "node-1"
- "node-2"
#- "node-3"
cluster.name: "wazuh-cluster"
discovery.seed_hosts:
- "172.16.1.201"
- "172.16.1.202"
# - "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=node-1,OU=Wazuh,O=Wazuh,L=California,C=US"
- "CN=node-2,OU=Wazuh,O=Wazuh,L=California,C=US"

```

Configuración MagellanWI2:

```

network.host: "172.16.1.202"
node.name: "node-2"
cluster.initial_master_nodes:
- "node-1"
- "node-2"
#- "node-3"
cluster.name: "wazuh-cluster"
discovery.seed_hosts:
- "172.16.1.201"
- "172.16.1.202"
# - "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=node-1,OU=Wazuh,O=Wazuh,L=California,C=US"
- "CN=node-2,OU=Wazuh,O=Wazuh,L=California,C=US"

```

5.10 Desplegando certificados

A continuación, declaro una variable como **node-1**:

```
# NODE_NAME=node-1
```

```
root@magellanwi1:/home/impeladmin# NODE_NAME=node-1
root@magellanwi1:/home/impeladmin# echo $NODE_NAME
node-1
```

Y despu s ejecuto los siguientes comandos para desplegar los certificados en mi indexador 1.

1. Crea un directorio llamado 'certs' dentro de '/etc/wazuh-indexer':

```
# mkdir /etc/wazuh-indexer/certs
```

2. Extrae archivos del archivo 'wazuh-certificates.tar' a '/etc/wazuh-indexer/certs/' y selecciona archivos espec ficos:

```
# 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
```

3. Renombra el archivo '\$NODE_NAME.pem' como 'indexer.pem' en el directorio '/etc/wazuh-indexer/certs/':

```
# mv -n /etc/wazuh-indexer/certs/${NODE_NAME}.pem
/etc/wazuh-indexer/certs/indexer.pem
```

4. Renombra el archivo '\$NODE_NAME-key.pem' como 'indexer-key.pem' en el directorio '/etc/wazuh-indexer/certs/'.

```
# mv -n /etc/wazuh-indexer/certs/${NODE_NAME}-key.pem
/etc/wazuh-indexer/certs/indexer-key.pem
```

5. Establece permisos de ejecuci n para el directorio '/etc/wazuh-indexer/certs'.

```
# chmod 500 /etc/wazuh-indexer/certs
```

6. Establece permisos de solo lectura para todos los archivos dentro de '/etc/wazuh-indexer/certs/'.

```
# chmod 400 /etc/wazuh-indexer/certs/*
```

7. Cambia el propietario y grupo del directorio '/etc/wazuh-indexer/certs' a 'wazuh-indexer'.

```
# chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
```

Ahora realizo los mismos pasos pero en magellanwi2 “**Indexador 2**” con la variable definida como **node-2**:

```
# NODE_NAME=node-2
```

```
root@magellanwi2:/home/impeleadmin# NODE_NAME=node-2
root@magellanwi2:/home/impeleadmin# echo $NODE_NAME
node-2
```

Y después ejecuto los siguientes comandos para desplegar los certificados en mi indexador 2.

1. Crea un directorio llamado 'certs' dentro de '/etc/wazuh-indexer':

```
# mkdir /etc/wazuh-indexer/certs
```

2. Extrae archivos del archivo 'wazuh-certificates.tar' a '/etc/wazuh-indexer/certs/' y selecciona archivos específicos:

```
# 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
```

3. Renombra el archivo '\$NODE_NAME.pem' como 'indexer.pem' en el directorio '/etc/wazuh-indexer/certs/':

```
# mv -n /etc/wazuh-indexer/certs/${NODE_NAME}.pem
/etc/wazuh-indexer/certs/indexer.pem
```

4. Renombra el archivo '\$NODE_NAME-key.pem' como 'indexer-key.pem' en el directorio '/etc/wazuh-indexer/certs/'.

```
# mv -n /etc/wazuh-indexer/certs/${NODE_NAME}-key.pem
/etc/wazuh-indexer/certs/indexer-key.pem
```

5. Establece permisos de ejecución para el directorio '/etc/wazuh-indexer/certs'.

```
# chmod 500 /etc/wazuh-indexer/certs
```

6. Establece permisos de solo lectura para todos los archivos dentro de '/etc/wazuh-indexer/certs/'.

```
# chmod 400 /etc/wazuh-indexer/certs/*
```

7. Cambia el propietario y grupo del directorio '/etc/wazuh-indexer/certs' a 'wazuh-indexer'.

```
# chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
```

5.11 Iniciar servicios

Para reiniciar el servicio **wazuh-indexer**:

```
# systemctl daemon-reload
# systemctl enable wazuh-indexer
# systemctl start wazuh-indexer`
```

Repetir en cada nodo.

5.12 Inicializar el Clúster de indexadores

Correr el siguiente script en cualquier nodo de Indexadores para cargar los nuevos certificados y arrancar el clúster multi-node:

```
# /usr/share/wazuh-indexer/bin/indexer-security-init.sh`
```

5.13 Verificar el funcionamiento del Clúster

Para comprobar que los indexadores están funcionando correctamente:

- **Instalación exitosa:**

```
root@magellanwi1:/home/impeladmin# curl -k -u admin:admin https://172.16.1.201:9200
{
  "name" : "node-1",
  "cluster_name" : "wazuh-cluster",
  "cluster_uuid" : "wAMyp06kRmSYjoSF0UVyHw",
  "version" : {
    "number" : "7.10.2",
    "build_type" : "rpm",
    "build_hash" : "db90a415ff2fd428b4f7b3f800a51dc229287cb4",
    "build_date" : "2023-06-03T06:24:25.112415503Z",
    "build_snapshot" : false,
    "lucene_version" : "9.6.0",
    "minimum_wire_compatibility_version" : "7.10.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "The OpenSearch Project: https://opensearch.org/"
}
```

```
root@magellanwi1:/home/impeladmin# curl -k -u admin:admin https://172.16.1.202:9200
{
  "name" : "node-2",
  "cluster_name" : "wazuh-cluster",
  "cluster_uuid" : "wAMyp06kRmSYjoSF0UVyHw",
  "version" : {
    "number" : "7.10.2",
    "build_type" : "rpm",
    "build_hash" : "db90a415ff2fd428b4f7b3f800a51dc229287cb4",
    "build_date" : "2023-06-03T06:24:25.112415503Z",
    "build_snapshot" : false,
    "lucene_version" : "9.6.0",
    "minimum_wire_compatibility_version" : "7.10.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "The OpenSearch Project: https://opensearch.org/"
}
```

```
# curl -k -u admin:admin https://172.16.1.201:9200
# curl -k -u admin:admin https://172.16.1.202:9200`
```

- Clúster funcionando correctamente:

```
root@magellanwi1:/home/impeladmin# curl -k -u admin:admin https://172.16.1.201:9200/_cat/nodes?v
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role node.roles           cluster_manager name
172.16.1.201    45        95     1   0.00    0.02    0.00 dimr      cluster_manager,data,ingest,remote_cluster_client -       node-1
172.16.1.202    36        97     1   0.00    0.06    0.07 dimr      cluster_manager,data,ingest,remote_cluster_client *       node-2
```

Para este paso no importa la IP del nodo del clúster que indiquemos:

```
`# curl -k -u admin:admin https://172.16.1.201:9200/_cat/nodes?v`
```

6. Desplegando Clúster de Servidores

6.1 Añadir repositorios de Wazuh

Como ahora estamos en los LXC (magellanws y magellanww) donde correrá wazuh-manager se debe añadir los repositorios de Wazuh nuevamente a ambos LXC:

```
`# apt-get install gnupg apt-transport-https`  
  
`# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg  
--no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import &&  
chmod 644 /usr/share/keyrings/wazuh.gpg`  
  
`# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]  
https://packages.wazuh.com/4.x/apt/ stable main" | tee -a  
/etc/apt/sources.list.d/wazuh.list`  
  
`# apt-get update`
```

La explicación de la acción que realiza cada comando se puede consultar en el apartado “Añadir repositorios de Wazuh” de los indexadores.

6.2 Instalando wazuh-manager

Para instalar el paquete de wazuh-manager:

```
`# apt-get -y install wazuh-manager`
```

6.3 Iniciar servicios

Habilitar e iniciar Wazuh-manager:

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

Después comprobar el estado de Wazuh-manager:

```
`# systemctl status wazuh-manager`
```

Magellanws funcionando correctamente:

```
impeleadmin@magellanws:~$ sudo systemctl status wazuh-manager
[sudo] password for impeleadmin:
* wazuh-manager.service - Wazuh manager
  Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2024-04-28 22:03:34 UTC; 19h ago
    Process: 6065 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
   Tasks: 125 (limit: 33549)
  Memory: 290.6M
     CPU: 11min 50.533s
    CGroup: /system.slice/wazuh-manager.service
              |-6121 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              |-6122 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              |-6125 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              |-6128 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              |-6169 /var/ossec/bin/wazuh-authd
              |-6186 /var/ossec/bin/wazuh-db
              |-6210 /var/ossec/bin/wazuh-execd
              |-6224 /var/ossec/bin/wazuh-analysisd
              |-6238 /var/ossec/bin/wazuh-syscheckd
              |-6284 /var/ossec/bin/wazuh-remoted
              |-6317 /var/ossec/bin/wazuh-logcollector
              |-6338 /var/ossec/bin/wazuh-monitord
              |-6380 /var/ossec/bin/wazuh-modulesd
              |-6794 /var/ossec/framework/python/bin/python3 /var/ossec/framework/scripts/wazuh_clusterd.py
              |-6864 /var/ossec/framework/python/bin/python3 /var/ossec/framework/scripts/wazuh_clusterd.py
              |-6865 /var/ossec/framework/python/bin/python3 /var/ossec/framework/scripts/wazuh_clusterd.py

Apr 28 22:03:24 magellanws env[6065]: Started wazuh-execd...
Apr 28 22:03:26 magellanws env[6065]: Started wazuh-analysisd...
Apr 28 22:03:27 magellanws env[6065]: Started wazuh-syscheckd...
Apr 28 22:03:28 magellanws env[6065]: Started wazuh-remoted...
Apr 28 22:03:29 magellanws env[6065]: Started wazuh-logcollector...
Apr 28 22:03:30 magellanws env[6065]: Started wazuh-monitord...
Apr 28 22:03:31 magellanws env[6065]: Started wazuh-modulesd...
Apr 28 22:03:32 magellanws env[6065]: Started wazuh-clusterd...
Apr 28 22:03:34 magellanws env[6065]: Completed.
Apr 28 22:03:34 magellanws systemd[1]: Started Wazuh manager.
```

Magellanww funcionando correctamente:

```
impeleadmin@magellanww:~$ sudo systemctl status wazuh-manager
[sudo] password for impeleadmin:
* wazuh-manager.service - Wazuh manager
  Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2024-04-28 22:04:10 UTC; 19h ago
    Process: 2564 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
   Tasks: 121 (limit: 33549)
  Memory: 296.4M
     CPU: 12min 7.668s
    CGroup: /system.slice/wazuh-manager.service
              |-2620 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              |-2621 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              |-2624 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              |-2627 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              |-2669 /var/ossec/bin/wazuh-authd
              |-2684 /var/ossec/bin/wazuh-db
              |-2708 /var/ossec/bin/wazuh-execd
              |-2722 /var/ossec/bin/wazuh-analysisd
              |-2735 /var/ossec/bin/wazuh-syscheckd
              |-2782 /var/ossec/bin/wazuh-remoted
              |-2814 /var/ossec/bin/wazuh-logcollector
              |-2836 /var/ossec/bin/wazuh-monitord
              |-2877 /var/ossec/bin/wazuh-modulesd
              |-3303 /var/ossec/framework/python/bin/python3 /var/ossec/framework/scripts/wazuh_clusterd.py
              |-3509 /var/ossec/framework/python/bin/python3 /var/ossec/framework/scripts/wazuh_clusterd.py

Apr 28 22:04:00 magellanww env[2564]: Started wazuh-execd...
Apr 28 22:04:01 magellanww env[2564]: Started wazuh-analysisd...
Apr 28 22:04:03 magellanww env[2564]: Started wazuh-syscheckd...
Apr 28 22:04:04 magellanww env[2564]: Started wazuh-remoted...
Apr 28 22:04:05 magellanww env[2564]: Started wazuh-logcollector...
Apr 28 22:04:06 magellanww env[2564]: Started wazuh-monitord...
Apr 28 22:04:07 magellanww env[2564]: Started wazuh-modulesd...
Apr 28 22:04:08 magellanww env[2564]: Started wazuh-clusterd...
Apr 28 22:04:10 magellanww env[2564]: Completed.
Apr 28 22:04:10 magellanww systemd[1]: Started Wazuh manager.
```

6.4 Solución de error en PROXMOX: Wazuh-analysisd

Al ver el estado del wazuh-manager.service se podía apreciar que había un error que no permitía establecer el límite de archivos abiertos por los permisos, porque son insuficientes. Y este error no ha sido por una mala configuración por mi parte, si no que después de haber investigado por Internet encontré una gran variedad de artículos que apuntaban a que el error se materializa al instalar Wazuh sobre un LXC de Proxmox.

```
root@magellanws:/home/impeladmin# systemctl status wazuh-manager
* wazuh-manager.service - Wazuh manager
  Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
    Active: active (running) since Sat 2024-04-27 21:01:55 UTC; 2s ago
      Process: 1462 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
     Tasks: 118 (limit: 33549)
    Memory: 188.8M
       CPU: 23.419s
      CGroup: /system.slice/wazuh-manager.service
              ├─1518 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              ├─1519 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              ├─1522 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              ├─1525 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
              ├─1526 /var/ossec/bin/wazuh-cthd
              ├─1593 /var/ossec/bin/wazuh-db
              ├─1607 /var/ossec/bin/wazuh-execd
              ├─1621 /var/ossec/bin/wazuh-analysisd
              ├─1635 /var/ossec/bin/wazuh-syscheckd
              ├─1681 /var/ossec/bin/wazuh-remoted
              ├─1714 /var/ossec/bin/wazuh-logcollector
              ├─1735 /var/ossec/bin/wazuh-monitord
              ├─1785 /var/ossec/bin/wazuh-modulesd
Apr 27 21:01:46 magellanws env[1462]: Started wazuh-execd...
Apr 27 21:01:46 magellanws env[1618]: 2024/04/27 21:01:46 wazuh-analysisd: ERROR: Could not set resource limit for file descriptors to 458752: Operation not permitted (1)
Apr 27 21:01:47 magellanws env[1462]: Started wazuh-analysisd...
Apr 27 21:01:48 magellanws env[1462]: Started wazuh-syscheckd...
Apr 27 21:01:49 magellanws env[1462]: Started wazuh-remoted...
Apr 27 21:01:50 magellanws env[1462]: Started wazuh-logcollector...
Apr 27 21:01:52 magellanws env[1462]: Started wazuh-monitord...
Apr 27 21:01:53 magellanws env[1462]: Started wazuh-modulesd...
Apr 27 21:01:55 magellanws env[1462]: Completed.
Apr 27 21:01:55 magellanws systemd[1]: Started Wazuh manager.
root@magellanws:/home/impeladmin# nano /etc/security/limits.conf
```

Tras un largo “research” encuentro la solución, debía cambiar dentro de la configuración del servicio (con privilegios de sudo) el límite “LimitNOFILE=x” a la cantidad que me indicaba el error, en este caso 458752:

```
root@magellanws:/home/impeladmin# cat /lib/systemd/system/wazuh-manager.service
[Unit]
Description=Wazuh manager
Wants=network-online.target
After=network.target network-online.target

[Service]
Type=forking
LimitNOFILE=458752 ←

ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start
ExecStop=/usr/bin/env /var/ossec/bin/wazuh-control stop
ExecReload=/usr/bin/env /var/ossec/bin/wazuh-control reload

KillMode=process
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

6.5 Instalar Filebeat

Para instalar el paquete de Filebeat ejecutar el siguiente comando:

```
# apt-get -y install filebeat
```

6.6 Configurar Filebeat

Primero hay que descargar el fichero de configuración predeterminada de Filebeat

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

Después hay que editar el fichero de configuración “/etc/filebeat/filebeat.yml” y reemplazar lo siguiente:

- **hosts:** La lista de indexadores de Wazuh a los que conectar. Aquí hay que proporcionar la IP de los indexadores de Wazuh que tengamos activos, en mi caso como tengo dos pondré ambos separados de la siguiente manera: **hosts: ["172.16.1.201:9200", "172.16.1.202:9200"]**.

```
impeladmin@magellanws:~$ sudo cat /etc/filebeat/filebeat.yml
# Wazuh - Filebeat configuration file
output.elasticsearch:
  hosts: ["172.16.1.201:9200", "172.16.1.202:9200"]
  protocol: https
  username: ${username}
  password: ${password}
  ssl.certificateAuthorities:
    - /etc/filebeat/certs/root-ca.pem
  ssl.certificate: "/etc/filebeat/certs/filebeat.pem"
  ssl.key: "/etc/filebeat/certs/filebeat-key.pem"
setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false
```

6.7 Crear almacén de claves seguro

Este paso sirve para almacenar las claves que se van a generar a continuación de manera segura:

```
# filebeat keystore create
```

6.8 Configurar credenciales predeterminadas

Añadir las entradas con el usuario y contraseña por defecto, por el momento, admin:admin. Soy consciente del riesgo de seguridad que esto implica pero esto el despliegue, el cambio de credenciales del sistema de Wazuh se realizará al final.

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

Keystore magellanws:

```
root@magellanws:/home/impeladmin# echo admin | filebeat keystore add username --stdin --force
Successfully updated the keystore
root@magellanws:/home/impeladmin# echo admin | filebeat keystore add password --stdin --force
Successfully updated the keystore
```

Keystore magellanww:

```
root@magellanww:/home/impeladmin# echo admin | filebeat keystore add username --stdin --force
Successfully updated the keystore
root@magellanww:/home/impeladmin# echo admin | filebeat keystore add password --stdin --force
Successfully updated the keystore
```

6.9 Descargar plantilla de alertas

Hay que descargar la plantilla de alertas para el indexador de Wazuh y establecer permisos de lectura para grupo y otros:

```
`# 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`
```

6.10 Descargar el módulo de Wazuh para Filebeat

Este módulo permite a Filebeat enviar datos de logs a Wazuh para su análisis y correlación con otros eventos de seguridad:

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

magellanws:

```
root@magellanws:/home/impeladmin# curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.3.tar.gz | tar -xvz -C /usr/share/filebeat/module
wazuh/
wazuh/archives/
wazuh/archives/ingest/
wazuh/archives/ingest/pipeline.json
wazuh/archives/config/
wazuh/archives/config/archives.yml
wazuh/archives/manifest.yml
wazuh/_meta/
wazuh/_meta/config.yml
wazuh/_meta/docs.asciidoc
wazuh/_meta/fields.yml
wazuh/alerts/
wazuh/alerts/ingest/
wazuh/alerts/ingest/pipeline.json
wazuh/alerts/config/
wazuh/alerts/config/alerts.yml
wazuh/alerts/manifest.yml
wazuh/module.yml
```

magellanww:

```
root@magellanww:/home/impeladmin# curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.3.tar.gz | tar -xvz -C /usr/share/filebeat/module
wazuh/
wazuh/archives/
wazuh/archives/ingest/
wazuh/archives/ingest/pipeline.json
wazuh/archives/config/
wazuh/archives/config/archives.yml
wazuh/archives/manifest.yml
wazuh/_meta/
wazuh/_meta/config.yml
wazuh/_meta/docs.asciidoc
wazuh/_meta/fields.yml
wazuh/alerts/
wazuh/alerts/ingest/
wazuh/alerts/ingest/pipeline.json
wazuh/alerts/config/
wazuh/alerts/config/alerts.yml
wazuh/alerts/manifest.yml
wazuh/module.yml
```

6.11 Desplegando certificados

A continuación, declaro una variable como **wazuh-1**:

```
`# NODE_NAME=wazuh-1`
```

```
root@magellanws:/home/impeladmin# NODE_NAME=wazuh-1
root@magellanws:/home/impeladmin# echo $NODE_NAME
wazuh-1
```

Y después ejecuto los siguientes comandos para desplegar los certificados en mi servidor 1.

```
`# 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`
```

Ahora realizo los mismos pasos pero en magellanww “**Servidor 2**” con la variable definida como **wazuh-2**:

```
# NODE_NAME=wazuh-2

root@magellanww:/home/impeladmin# NODE_NAME=wazuh-2
root@magellanww:/home/impeladmin# echo $NODE_NAME
wazuh-2

# 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`
```

NOTA: Para ver la acción que realiza cada comando en caso de duda está explicado en el despliegue de certificados “Desplegando certificados” del clúster de indexadores.

6.12 Iniciar el servicio de Filebeat

Para habilitar e iniciar el servicio de Filebeat:

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

6.13 Verificar el estado de Filebeat

Para verificar que Filebeat está correctamente instalado y funcionando ejecutar el siguiente comando:

```
# filebeat test output`
```

magellanws:

```
root@magellanws:/home/impeladmin# filebeat test output
elasticsearch: https://172.16.1.201:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 172.16.1.201
    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
elasticsearch: https://172.16.1.202:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 172.16.1.202
    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
```

magellanww:

```
root@magellanww:/home/impeladmin# filebeat test output
elasticsearch: https://172.16.1.201:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 172.16.1.201
    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
elasticsearch: https://172.16.1.202:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 172.16.1.202
    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
```

6.14 Inicializar el Clúster de servidores

Después de haber completado la instalación del servidor de Wazuh en cada nodo necesito configurar un nodo como rol “master” y otro como rol “worker”.

Nodo Master:

Editar la ruta “/var/ossec/etc/ossec.conf”:

```
<cluster>
<name>wazuh</name>
<node_name>master-node</node_name>
<node_type>master</node_type>
<key>86600b10f2876bc5879c87be9236822f</key>
<port>1516</port>
<bind_addr>0.0.0.0</bind_addr>
<nodes>
    <node>172.16.1.200</node>
</nodes>
<hidden>no</hidden>
<disabled>no</disabled>
</cluster>
```

```
<cluster>
<name>wazuh</name>
<node_name>master-node</node_name>
<node_type>master</node_type>
<key>86600b10f2876bc5879c87be9236822f</key>
<port>1516</port>
<bind_addr>0.0.0.0</bind_addr>
<nodes>
    <node>172.16.1.200</node>
</nodes>
<hidden>no</hidden>
<disabled>no</disabled>
</cluster>
```

- **name:** Indica el nombre del clúster.
- **node_name:** Indica el nombre del nodo actual.
- **node_type:** Especifica el rol del nodo. Debe configurarse como maestro.
- **key:** Clave que se utiliza para cifrar la comunicación entre los nodos del clúster. La clave debe tener 32 caracteres de longitud y ser la misma para todos los nodos. Para generar la clave: openssl rand -hex 16.
- **port:** Indica el puerto de destino para la comunicación del clúster.
- **bind_addr:** Es la dirección IP a la que el nodo está configurado para escuchar solicitudes entrantes (0.0.0.0 para cualquier IP).
- **nodes:** Es la dirección del nodo maestro y puede ser una IP o un DNS. Este parámetro debe especificarse en todos los nodos, incluido el maestro mismo.
- **hidden:** Muestra u oculta la información del clúster en las alertas generadas.
- **disabled:** Indica si el nodo está habilitado o deshabilitado en el clúster.

Nodo Worker:

Editar la ruta “/var/ossec/etc/ossec.conf”:

```
<cluster>
  <name>wazuh</name>
  <node_name>worker-node</node_name>
  <node_type>worker</node_type>
  <key>86600b10f2876bc5879c87be9236822f</key>
  <port>1516</port>
  <bind_addr>0.0.0.0</bind_addr>
  <nodes>
    <node>172.16.1.200</node>
  </nodes>
  <hidden>no</hidden>
  <disabled>no</disabled>
</cluster>
```

```
<cluster>
  <name>wazuh</name>
  <node_name>worker-node</node_name>
  <node_type>worker</node_type>
  <key>86600b10f2876bc5879c87be9236822f</key>
  <port>1516</port>
  <bind_addr>0.0.0.0</bind_addr>
  <nodes>
    <node>172.16.1.200</node>
  </nodes>
  <hidden>no</hidden>
  <disabled>no</disabled>
</cluster>
```

- **name:** Indica el nombre del clúster.
- **node_name:** Indica el nombre del nodo actual. Cada nodo del clúster debe tener un nombre único.
- **node_type:** Especifica el rol del nodo. Debe configurarse como trabajador.
- **key:** La clave creada previamente para el nodo maestro. Debe ser la misma para todos los nodos.
- **nodes:** Debe contener la dirección del nodo maestro y puede ser una IP o un DNS.
- **disabled:** Indica si el nodo está habilitado o deshabilitado en el clúster. Debe establecerse en no.

Después de haber editado los ficheros de los dos nodos ejecutar en cada uno:

```
`# systemctl restart wazuh-manager`
```

6.15 Verificar el funcionamiento del Clúster

Para verificar que el clúster está habilitado y todos los nodos están conectados ejecutar el comando:

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

Un buen estado del clúster sería como el siguiente:

```
root@magellanws :/home/impeladmin# /var/ossec/bin/cluster_control -l
NAME      TYPE    VERSION   ADDRESS
master-node  master  4.7.3    172.16.1.200
worker-node  worker  4.7.3    172.16.1.203
```

7. Desplegando la Dashboard de Wazuh

7.1 Instalación de dependencias

Instalar las siguientes dependencias en caso de faltar:

```
# apt-get install debhelper tar curl libcap2-bin` #debhelper version 9 or later
```

7.2 Añadir repositorios de Wazuh

Al estar ahora en una nueva máquina (“wazuhdashboard”) hay que añadir los repositorios de Wazuh al nuevo sistema.

```
# apt-get install gnupg apt-transport-https
```

```
# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring  
--keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644  
/usr/share/keyrings/wazuh.gpg
```

```
# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]  
https://packages.wazuh.com/4.x/apt/ stable main" | tee -a  
/etc/apt/sources.list.d/wazuh.list
```

```
# apt-get update`
```

7.3 Instalando wazuh-dashboard

Para instalar el paquete con la dashboard de Wazuh:

```
# apt-get -y install wazuh-dashboard`
```

7.4 Configurando wazuh-dashboard

Para configurar el archivo con los parámetros necesarios para que el servicio de la dashboard esté en escucha editar “ /etc/wazuh-dashboard/opensearch_dashboards.yml”.

```
root@magellanwashboard:/home/impeladmin# cat /etc/wazuh-dashboard/opensearch_dashboards.yml
server.host: 172.16.1.204
server.port: 443
opensearch.hosts: ["https://172.16.1.201:9200", "https://172.16.1.202:9200"]
opensearch.ssl.verificationMode: certificate
#opensearch.username:
#opensearch.password:
opensearch.requestHeadersAllowlist: ["securitytenant", "Authorization"]
opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ["kibana_read_only"]
server.ssl.enabled: true
server.ssl.key: "/etc/wazuh-dashboard/certs/dashboard-key.pem"
server.ssl.certificate: "/etc/wazuh-dashboard/certs/dashboard.pem"
opensearch.ssl.certificateAuthorities: ["/etc/wazuh-dashboard/certs/root-ca.pem"]
uiSettings.overrides.defaultRoute: /app/wazuh
```

- **server.host:** Esta configuración especifica el host del servidor de la interfaz de usuario de Wazuh. Para permitir que los usuarios remotos se conecten, establece el valor en la dirección IP o nombre DNS del servidor de la interfaz de usuario de Wazuh. El valor 0.0.0.0 aceptará todas las direcciones IP disponibles del host.
- **opensearch.hosts:** Las URL de las instancias del indexador de Wazuh que se utilizarán para todas las consultas. Las direcciones de los nodos pueden separarse por comas. Por ejemplo, ["https://172.16.1.201:9200", "https://172.16.1.202:9200"].

7.5 Desplegando certificados

```
`# NODE_NAME=dashboard`  
  
root@magellanwashboard:/home/impeladmin# NODE_NAME=dashboard  
root@magellanwashboard:/home/impeladmin# echo $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`
```

7.6 Iniciar servicios

Para habilitar e iniciar el servicio de wazuh-dashboard:

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

7.7 Apuntar al Master Node

Editar el archivo “/usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml” y especificar la IP del nodo Maestro:

```
hosts:
  - default:
      url: https://172.16.1.200
      port: 55000
      username: wazuh-wui
      password: wazuh-wui
      run_as: false
```

En mi caso es la 172.16.1.200.

7.8 Accediendo a la interfaz web de Wazuh

Para acceder a la web de Wazuh una vez completado el despliegue multi-node:

URL: https://<wazuh-dashboard-ip>

Username: admin

Password: admin

7.9 Cambiar contraseña web de admin

Para cambiar la contraseña del panel de login web de Wazuh introducir el siguiente comando desde **ambos** indexadores de Wazuh:

```
# /usr/share/wazuh-indexer/plugins/opensearch-security/tools/wazuh-passwords-tool.sh -u admin -p [Contraseña_nueva]`
```

Y ejecutar este comando en ambos nodos wazuh-manager del Clúster:

```
# echo <Contraseña_nueva> | filebeat keystore add password --stdin --force`
```

Por último ejecutar también el siguiente comando en ambos nodos wazuh-manager:

```
# systemctl restart filebeat`
```

8. Desplegando agente de Wazuh

8.1 Añadir agente

The screenshot shows the Wazuh dashboard interface. At the top, it displays the number of agents: Total agents (0), Active agents (0), Disconnected agents (0), Pending agents (0), and Never connected agents (0). A prominent message says "No agents were added to this manager. Add agent". Below this, there are two main sections: "SECURITY INFORMATION MANAGEMENT" and "AUDITING AND POLICY MONITORING". Each section contains four sub-modules:

- SECURITY INFORMATION MANAGEMENT:**
 - Security events: Browse through your security alerts, identifying issues and threats in your environment.
 - Integrity monitoring: Alerts related to file changes, including permissions, content, ownership and attributes.
- AUDITING AND POLICY MONITORING:**
 - Policy monitoring: Verify that your systems are configured according to your security policies baseline.
 - System auditing: Audit users behavior, monitoring command execution and alerting on access to critical files.
 - Security configuration assessment: Scan your assets as part of a configuration assessment audit.

Lo primero será dirigirse al apartado de “Add agent”. Donde nos redirigirá a una página donde comenzaré con la configuración del agente a desplegar en la máquina Kumacamole. He elegido este servidor primero porque es el más crítico de la infraestructura, ya que se ha de pasar por este nodo para conectarse al resto de la infraestructura.

8.2 Desplegar agente

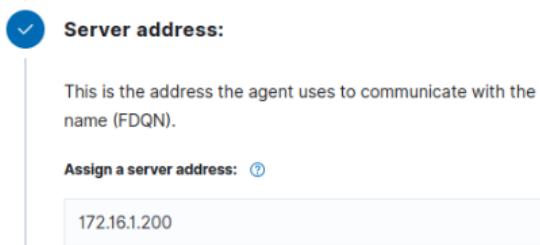
Como Kumacamole es un ubuntu 22.04 amd64 elegiré la opción “DEB amd64”.

The screenshot shows the "Deploy new agent" page. It starts with a dropdown menu with the instruction "Select the package to download and install on your system:". Below this, there are three main sections corresponding to different operating systems:

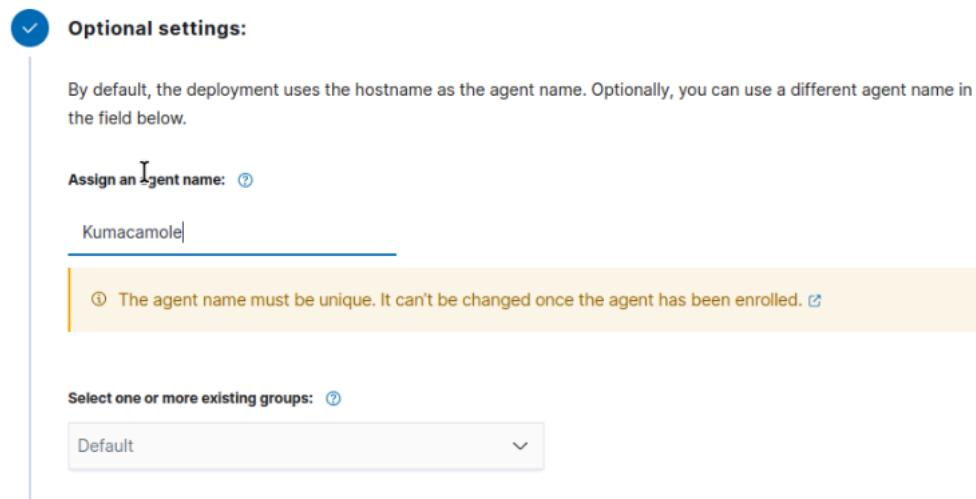
- LINUX:** Contains options for "RPM amd64", "RPM aarch64", "DEB amd64" (which is selected), and "DEB aarch64".
- WINDOWS:** Contains the option "MSI 32/64 bits".
- macOS:** Contains options for "Intel" and "Apple silicon".

At the bottom of the page, there is a note: "For additional systems and architectures, please check our documentation."

El siguiente paso será apuntar a la dirección IP de mi servidor Maestro de Wazuh, la “172.16.1.200”.



A continuación, hay que nombrar el agente con un nombre único y que no se cambiará más adelante. En este caso lo nombraré con un nombre identificador y simple como “Kumacamole”. El grupo será el de por defecto, ya que no hay grupos creados todavía.



El cuarto paso es ejecutar el siguiente script en la consola de Kumacamole como root. El cual descargará el paquete del agente con los parámetros aportados anteriormente y ejecutará el paquete.

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

```
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.4-1_amd64.deb &&
sudo WAZUH_MANAGER='172.16.1.200' WAZUH_AGENT_NAME='Kumacamole' dpkg -i ./wazuh-
agent_4.7.4-1_amd64.deb
```

① Requirements

- You will need administrator privileges to perform this installation.
- Shell Bash is required.

Keep in mind you need to run this command in a Shell Bash terminal.

Y por último se deberá reiniciar los “demonios” del sistema y habilitar e iniciar el agente.

5 Start the agent:

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

Y esta es la vista del “Home” de Wazuh al finalizar el despliegue del agente. Como se puede apreciar en la imagen se cambia a “1” el número total de agentes disponibles y activos.

The screenshot shows the Wazuh Home dashboard with the following statistics:

- Total agents: 1
- Active agents: 1
- Disconnected agents: 0
- Pending agents: 0
- Never connected agents: 0

The dashboard is divided into several sections:

- SECURITY INFORMATION MANAGEMENT:**
 - Security events: Browse through your security alerts, identifying issues and threats in your environment.
 - Integrity monitoring: Alerts related to file changes, including permissions, content, ownership and attributes.
- AUDITING AND POLICY MONITORING:**
 - Policy monitoring: Verify that your systems are configured according to your security policies baseline.
 - System auditing: Audit users behavior, monitoring command execution and alerting on access to critical files.
- THREAT DETECTION AND RESPONSE:**
- REGULATORY COMPLIANCE:**

Y esta sería la vista de la pestaña de monitorización del agente:

The screenshot shows the Wazuh Agents dashboard for agent ID 001, named Kumacamole, running on Ubuntu 22.04.4 LTS (master-node).

Agent details:

ID	Status	IP address	Version	Groups	Operating system	Cluster node
001	●	172.16.1.11	Wazuh v4.7.4	default	Ubuntu 22.04.4 LTS	master-node

Registration date: May 6, 2024 @ 21:37:23.000

Last keep alive: May 6, 2024 @ 22:10:20.000

MITRE Top Tactics:

- Credential Access: 3
- Lateral Movement: 2
- Defense Evasion: 1

Compliance (Last 24 hours):

PCI DSS
2.2 (183)
10.6.1 (7)
10.2.4 (3)
10.2.5 (3)
10.2.6 (2)

9. Generación de alertas via Gmail de nivel 12

9.1 Instalación de dependencias

Debido a que Wazuh no incorpora autenticación SMTP de manera nativa hará falta instalar ciertos paquetes necesarios para hacer un “server relay” de Postfix:

```
`# apt-get update && apt-get install postfix mailutils libsasl2-2 ca-certificates
libsasl2-modules`
```

En este paso si aparece una ventana para seleccionar el tipo de configuración hay que marcar “No configuration”.

9.2 Configuración de “main.cf”

A continuación, editar o crear si hace falta el siguiente fichero de configuración “/etc/postfix/main.cf” con el siguiente código:

```
relayhost = [smtp.gmail.com]:587
mynetworks = 127.0.0.0/8
inet_interfaces = loopback-only
smtp_use_tls = yes
smtp_always_send_ehlo = yes
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_sasl_tls_security_options = noanonymous
smtp_tls_security_level = encrypt
smtp_generic_maps = hash:/etc/postfix/generic
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtpd_relay_restrictions = permit_mynetworks, permit_sasl_authenticated,
defer_unauth_destination
```

9.3 Crear los ficheros de autenticación y configurar permisos

Ahora hay que crear el fichero “/etc/postfix/sasl_passwd” con la cuenta de gmail (en mi caso) y la contraseña APP PASSWORD generada, NO la contraseña de la cuenta.

Según Google una App Password es una contraseña de 16 dígitos que le otorga permiso a un dispositivo o app menos seguros para acceder a la cuenta de Google. Las contraseñas de aplicaciones solo se pueden usar con cuentas que tengan la Verificación en 2 pasos activada.

Para este proyecto me he tomado la libertad de crear una cuenta “remitente y receptor” (reconmagellanw@gmail.com) donde tengo activada la verificación en 2 pasos y además me configuré una App Password.

Este debe ser el contenido de “/etc/postfix/sasl_passwd”:

```
`# [smtp.gmail.com]:587 reconmagellanw@gmail.com:<AppPasswordHere>`
```

Ahora hay que configurar correctamente los permisos:

```
`# sudo chown root:root /etc/postfix/sasl_passwd
# sudo chmod 0600 /etc/postfix/sasl_passwd
# sudo postmap /etc/postfix/sasl_passwd`
```

El comando "postmap" se utiliza para convertir archivos de mapa de texto plano en un formato más eficiente para su uso por parte de Postfix.

Después crear el siguiente fichero genérico “/etc/postfix/generic” con el contenido:

```
root@localdomain reconmagellanw@gmail.com
@localdomain reconmagellanw@gmail.com
```

Donde “reconmagellanw@gmail.com” es mi correo electrónico. Y de la misma manera configurar los permisos correspondientes:

```
`# sudo chown root:root /etc/postfix/generic
# sudo chmod 0600 /etc/postfix/generic
# sudo postmap /etc/postfix/generic`
```

Y reiniciar el servicio de Postfix para realizar los cambios:

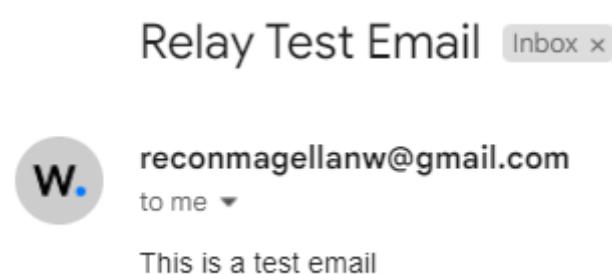
```
`# systemctl restart postfix`
```

9.4 Verificar el funcionamiento de Postfix

Para comprobar la comunicación del servidor “relay” de correo que hemos instalado en el nodo ejecutar este comando:

```
`# echo "This is a test email" | mail -s "Relay Test Email" reconmagellanw@gmail.com
-a "FROM:reconmagellanw@gmail.com"`
```

Debería llegar la notificación si todo ha salido correctamente.



9.5 Configurar ossec.conf

Si todo ha ido exitosamente editar en el fichero “/var/ossec/etc/ossec.conf” las siguientes líneas:

```
<email_notification>yes</email_notification>
<smtp_server>localhost</smtp_server>
<email_from>reconmagellanw@gmail.com</email_from>
<email_to>reconmagellanw@gmail.com</email_to>
<email_maxperhour>12</email_maxperhour>
```

- **email_notification:** Habilitarla a “yes”, lo que indicará que el sistema enviará notificaciones por correo electrónico.
- **smtp_server:** El servidor SMTP que se utilizará para enviar los correos electrónicos es "localhost", lo que significa que el sistema enviará correos electrónicos a través del servidor SMTP que se ejecuta en la misma máquina.
- **email_from:** La dirección de correo electrónico que se utilizará como remitente en los correos electrónicos es "reconmagellanw@gmail.com".
- **email_to:** La dirección de correo electrónico a la que se enviarán las notificaciones es "reconmagellanw@gmail.com".
- **email_maxperhour:** Se limita el número máximo de correos electrónicos que se pueden enviar por hora a 12. Esto puede ser útil para evitar el envío excesivo de correos electrónicos y posibles problemas de spam.

Y por último reiniciar el proceso de Wazuh-manager:

```
`# systemctl restart wazuh-manager`
```

NOTA: Si se quieren añadir más recipientes hay que añadir más etiquetas de <email_to>.

10. Reglas customizadas

10.1 Introducción y rutas de las reglas

Hay dos rutas a tener en cuenta a la hora de crear reglas:

- 1º `/var/ossec/ruleset/rules/*`
- 2º `/var/ossec/etc/rules/local_rules.xml`

La primera ruta es donde se encuentran las plantillas de las reglas preconfiguradas por Wazuh. Y la segunda es donde se pueden customizar las reglas en base al “output” que devuelve el servidor, por ejemplo, si se hace un login por SSH a magellanws se dispararán dos reglas: 5715 (sshd) y 5501 (PAM: login). Al estar ya configuradas no hay que hacer nada, tan solo se copiará el id de la regla que quieras y en la primera ruta ejecutar `# grep -rl "id_regla"` para encontrar en qué archivo está y entender cómo funciona y de qué regla padre cuelga.

Para entender la estructura de reglas padre y reglas hijo, las reglas padre contienen una agrupación de instrucciones que definen el propio grupo al que pertenecen mediante decoders que apuntan a los diferentes logs del sistema, por ejemplo, el grupo sshd.

Una vez se obtiene el id de la regla padre y ejecutado el comando `# grep -rl "id_regla"` en `"/var/ossec/ruleset/rules/"` se puede obtener el archivo donde esta la regla, visualizarlo y buscar donde está el bloque de esa regla, lo copiamos y lo introducimos en el archivo `"/var/ossec/etc/rules/local_rules.xml"` modificando el campo “rule id=x” donde “x” será el nuevo “rule_id” para no alterar la anterior y subiendo el nivel a mínimo 12 nos notificará por email (suponiendo que el mínimo está establecido en 12 en el archivo `"/var/ossec/etc/ossec.conf"`).

NOTA: Las reglas que he configurado están documentadas en el propio fichero `"local_rules.xml"`.

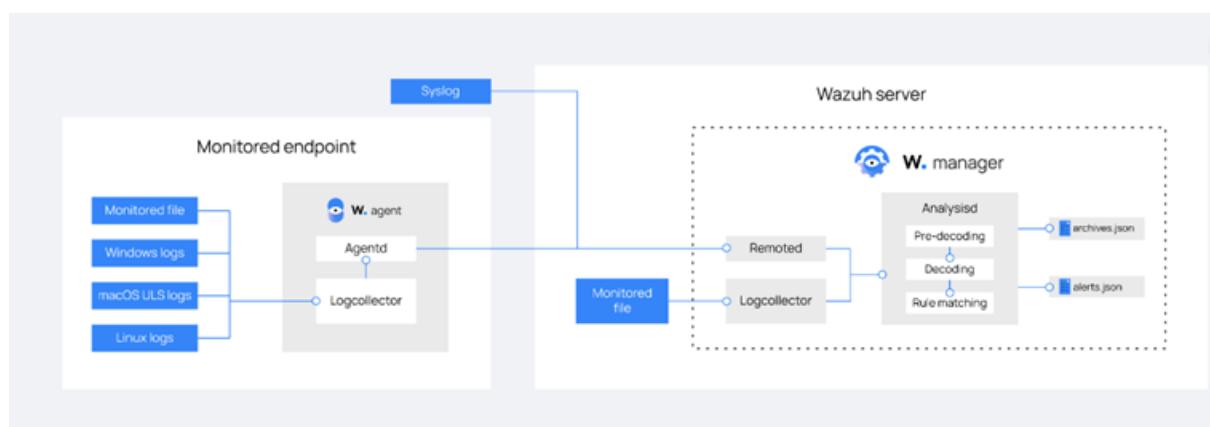
11. Eliminación de logs antiguos con CRON

11.1 Introducción

Por defecto Wazuh almacena logs y no los elimina automáticamente. Sin embargo, existen maneras para borrarlos de forma automática o manual, en este caso para no estar pendientes y gastar tiempo en borrarlos a mano he configurado una tarea programada con CRON por consola para eliminar ficheros automáticamente más antiguos de 90 días.

El módulo de Analysisd en el Servidor de Wazuh evalúa los logs decodificados de contra las reglas y los almacena en los archivos “`/var/ossec/logs/alerts/alerts.log`” y “`/var/ossec/logs/alerts/alerts.json`”.

Además de los logs de alerta, Wazuh almacena todos los logs recogidos en archivos de log dedicados, concretamente en “`/var/ossec/logs/archives/archives.log`” y “`/var/ossec/logs/archives/archives.json`”. Estos archivos de log capturan todos los logs, incluidos aquellos que no disparan ninguna alerta para un futuro análisis más detallado con referencias.



11.2 Sintaxis de CRON



11.3 Tarea programada

El comando para editar la tarea programada es:

```
# crontab -e
```

Y el contenido que hay que añadir al final del archivo es:

```
0 0 1 * * find /var/ossec/logs/alerts/ -name "*.gz" -type f -mtime +90 -exec rm -f {} \;
0 0 1 * * find /var/ossec/logs/archives/ -name "*.sum" -type f -mtime +90 -exec rm -f {} \;
```

La explicación del contenido anterior es que se ejecutarán dos tareas programadas a las 00:00 am, el día 1 de cada mes, y ejecutarán un “find” para buscar archivos en las rutas “/var/ossec/logs/alerts” y “/var/ossec/logs/archives” con las extensiones “.gz” y “.sum”, que sean de tipo archivo, con un tiempo superior a 90 días y que ejecute un borrado forzado con “rm -f” de la cadena de archivos encontrada “{}”.

```
root@magellanws:/home/impeladmin# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#
0 0 1 * * find /var/ossec/logs/alerts/ -name "*.gz" -type f -mtime +90 -exec rm -f {} \;
0 0 1 * * find /var/ossec/logs/archives/ -name "*.sum" -type f -mtime +90 -exec rm -f {} \;
```

El mínimo de espacio de disco dependerá del número de equipos conectados al servidor Wazuh:

- Disk space requirements

The amount of data depends on the generated alerts per second (APS). This table details the estimated disk space needed per agent to store 90 days of alerts on a Wazuh server, depending on the type of monitored endpoints.

Monitored endpoints	APS	Storage in Wazuh Server (GB/90 days)
Servers	0.25	0.1
Workstations	0.1	0.04
Network devices	0.5	0.2

For example, for an environment with 80 workstations, 10 servers, and 10 network devices, the storage needed on the Wazuh server for 90 days of alerts is 6 GB.

12. Breve introducción a Host Bastión

El propósito por el que existe esta máquina es para proteger el interior de la infraestructura interponiendo esta máquina entre la intranet y el exterior. De esta manera se deberá primero acceder al host bastión haciendo de “portón” para tener “acceso” con los permisos adecuados al resto de la infraestructura. La gracia de esta máquina es que no se accede directamente por reenvío de puertos como tal a la propia máquina, si no que se instalará y administrará el servicio de Apache Guacamole en Docker, para una administración más sencilla y remota de todas las conexiones existentes vía web. Existiendo para acceder a este una contraseña y un factor de autenticación en dos pasos.

12.1 Introduciendo Apache Guacamole

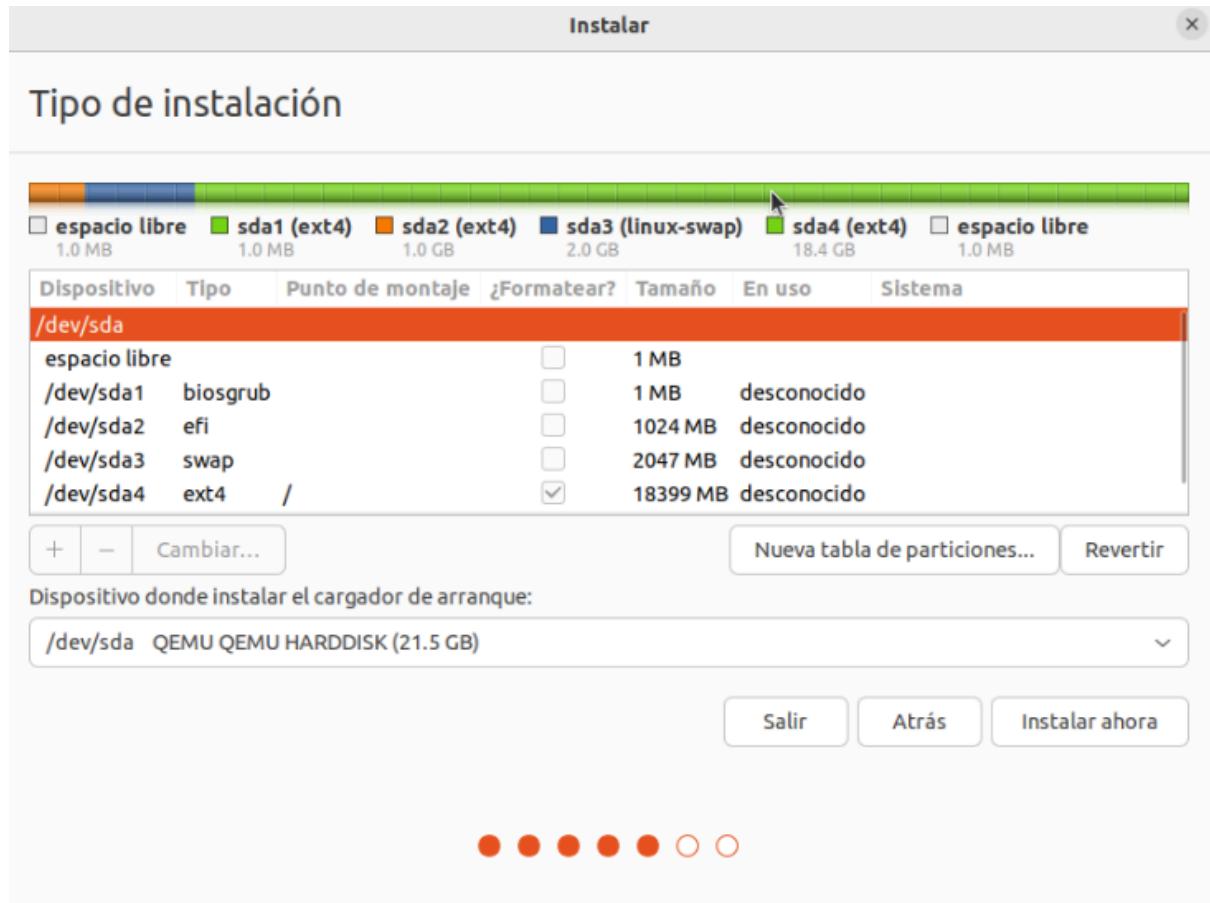
Guacamole es una aplicación web HTML5 que proporciona acceso a entornos de escritorio utilizando protocolos de escritorio remoto (como VNC o RDP). Guacamole también es el proyecto que produce esta aplicación web y proporciona una API que la impulsa. Esta API puede ser utilizada para alimentar otras aplicaciones o servicios similares.

"Guacamole" se usa comúnmente para referirse a la aplicación web producida por el proyecto Guacamole utilizando su API. Esta aplicación web es parte de una pila que proporciona una puerta de enlace de escritorio remoto independiente del protocolo. Escrita en JavaScript y utilizando solo HTML5 y otros estándares, la parte cliente de Guacamole no requiere nada más que un navegador web moderno o un dispositivo habilitado para la web cuando se accede a cualquiera de los escritorios servidos.

Guacamole permite acceder a uno o más escritorios desde cualquier lugar de forma remota, sin tener que instalar un cliente, especialmente cuando instalar un cliente no es posible. Al configurar un servidor Guacamole, puedes proporcionar acceso a cualquier otra computadora en la red desde prácticamente cualquier otra computadora en Internet, en cualquier parte del mundo. Incluso se pueden usar teléfonos móviles o tabletas, sin tener que instalar nada.

13. Configuración de Kumacamole

13.1 Configuración de disco de Kumacamole



Estas son las asignaciones de las particiones del disco duro del host bastión, de ahora en adelante llamado (kumacamole), habrán 4:

- **Bios Grub -> 1 MB**
- **efi -> 1024 MB**
- **swap -> 2047 MB**
- **ext4 -> 18399 MB**

Con esta configuración se logra que en caso de que se llene la memoria RAM siempre hayan disponibles 2 gb de espacio de disco para que no se congele el servicio y se puedan salvaguardar posibles cambios en archivos además de los 18 gb libres que dispone para guardar datos.

13.2 Configuración de red de Kumacamole



La configuración de IPv4 con IP “172.16.1.11”, máscara de red “/24” y puerta de enlace apuntando la interfaz de red del proxmox “172.16.1.254” en DMZ y DNS de Google, “8.8.8.8”.

```
impeladmin@kumacamole:~/Escritorio$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 26:d2:4e:1b:07:b4 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 172.16.1.11/24 brd 172.16.1.255 scope global noprefixroute ens18
        valid_lft forever preferred_lft forever
    inet6 fe80::3151:7240:fd4b:df54/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Breve comprobación de la salida del comando “ip a” para comprobar que se ha aplicado la configuración.

13.3 Configuración de “/etc/hosts”

```
Impeladmin@kumacamole:~/Escritorio$ sudo cat /etc/hosts
[sudo] contraseña para impeladmin:
127.0.0.1      localhost
127.0.1.1      kumacamole.impeldown.com kumacamole
172.16.1.11    kumacamole.impeldown.com kumacamole

# Impel-Down
172.16.1.20    crimsonldap
172.16.1.130   logposerb
172.16.1.200   magellanws
172.16.1.201   magellanwi1
172.16.1.202   magellanwi2
172.16.1.203   magellanww
172.16.1.204   magellanwdashboard
```

En la captura de pantalla ya sale configurado la unión de dominio y todos los demás servidores con sus nombres resueltos porque esta captura es posterior a la creación de la infraestructura. Como adelanto se puede configurar “/etc/hosts” para tenerlo ya preparado.

14. Configuración previa de Apache Guacamole

14.1 Instalación de Docker en Ubuntu

Link oficial: [Install Docker Engine on Ubuntu | Docker Docs](#)

Desinstalar versiones antiguas:

```
`# for pkg in docker.io docker-doc docker-compose docker-compose-v2
podman-docker containerd runc; do sudo apt-get remove $pkg; done`
```

Establecer los repositorios apt:

```
`# sudo apt-get update
# sudo apt-get install ca-certificates curl
# sudo install -m 0755 -d /etc/apt/keyrings
# sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
# sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
# echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
# sudo apt-get update`
```

Instalar los paquetes de Docker:

```
'# sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
docker-compose-plugin'
```

15. Desplegando Apache Guacamole

15.1 Instalación y configuración de Guacamole

Link Github: [GitHub - boschkundendienst/guacamole-docker-compose: Guacamole with docker-compose using PostgreSQL, nginx with SSL \(self-signed\)](https://github.com/boschkundendienst/guacamole-docker-compose)

Inicio rápido

Clona el repositorio GIT y arranca Guacamole:

```
'# git clone "https://github.com/boschkundendienst/guacamole-docker-compose.git"  
  
# cd guacamole-docker-compose  
  
# ./prepare.sh  
  
# docker-compose up -d'
```

El servidor de Guacamole debería estar disponible en https://ip_servidor:8443/. El nombre de usuario por defecto es guacadmin con la contraseña guacadmin.

Detalles:

Redes

La siguiente parte de docker-compose.yml creará una red con el nombre guacnetwork_compose en modo puente (bridged).

```
# crear una red 'guacnetwork_compose' en modo 'bridged'  
networks:  
  guacnetwork_compose:  
    driver: bridge  
Servicios  
guacd
```

La siguiente parte de docker-compose.yml creará el servicio guacd. guacd es el corazón de Guacamole que carga dinámicamente el soporte para protocolos de escritorio remoto (llamados "plugins de cliente") y los conecta a escritorios remotos basándose en las instrucciones recibidas de la aplicación web. El contenedor se llamará guacd_compose basado en la imagen de Docker guacamole/guacd conectada a nuestra red creada previamente guacnetwork_compose. Además, mapeamos las dos carpetas locales ./drive y ./record en el contenedor. Podemos usarlas más tarde para mapear unidades de usuario y almacenar grabaciones de sesiones.

```
services:
# guacd
guacd:
  container_name: guacd_compose
  image: guacamole/guacd
  networks:
    guacnetwork_compose:
  restart: always
  volumes:
    - ./drive:/drive:rw
    - ./record:/record:rw
```

PostgreSQL

La siguiente parte de docker-compose.yml creará una instancia de PostgreSQL usando la imagen oficial de Docker. Esta imagen es altamente configurable usando variables de entorno. Por ejemplo, se inicializará una base de datos si se encuentra un script de inicialización en la carpeta /docker-entrypoint-initdb.d dentro de la imagen. Dado que mapeamos la carpeta local ./init dentro del contenedor como docker-entrypoint-initdb.d, podemos inicializar la base de datos para Guacamole usando nuestro propio script (./init/initdb.sql).

```
postgres:
  container_name: postgres_guacamole_compose
  environment:
    PGDATA: /var/lib/postgresql/data/guacamole
    POSTGRES_DB: guacamole_db
    POSTGRES_PASSWORD: ChooseYourOwnPasswordHere1234
    POSTGRES_USER: guacamole_user
  image: postgres
  networks:
    guacnetwork_compose:
  restart: always
  volumes:
    - ./init:/docker-entrypoint-initdb.d:ro
    - ./data:/var/lib/postgresql/data:rw
```

Guacamole

La siguiente parte de docker-compose.yml creará una instancia de Guacamole usando la imagen de Docker guacamole del Docker Hub. También es altamente configurable usando variables de entorno. En esta configuración, está configurado para conectarse a la instancia de postgres creada anteriormente usando un nombre de usuario y contraseña y la base de datos guacamole_db. ¡El puerto 8080 solo se expone localmente! Adjuntamos una instancia de nginx para su exposición pública en el siguiente paso.

```
guacamole:
  container_name: guacamole_compose
  depends_on:
    - guacd
    - postgres
  environment:
    GUACD_HOSTNAME: guacd
    POSTGRES_DATABASE: guacamole_db
    POSTGRES_HOSTNAME: postgres
    POSTGRES_PASSWORD: ChooseYourOwnPasswordHere1234
    POSTGRES_USER: guacamole_user
  image: guacamole/guacamole
  links:
    - guacd
  networks:
    guacnetwork_compose:
  ports:
    - 8080/tcp
  restart: always
```

nginx

La siguiente parte de docker-compose.yml creará una instancia de nginx que mapea el puerto público 8443 al puerto interno 443. El puerto interno 443 luego se mapea a Guacamole usando el archivo ./nginx/templates/guacamole.conf.template. El contenedor usará el certificado autofirmado generado previamente (prepare.sh) en ./nginx/ssl/ con ./nginx/ssl/self-ssl.key y ./nginx/ssl/self.cert.

```
# nginx
nginx:
  container_name: nginx_guacamole_compose
  restart: always
  image: nginx
  volumes:
    - ./nginx/templates:/etc/nginx/templates:ro
    - ./nginx/ssl/self.cert:/etc/nginx/ssl/self.cert:ro
    - ./nginx/ssl/self-ssl.key:/etc/nginx/ssl/self-ssl.key:ro
  ports:
    - 8443:443
```

links:

- guacamole

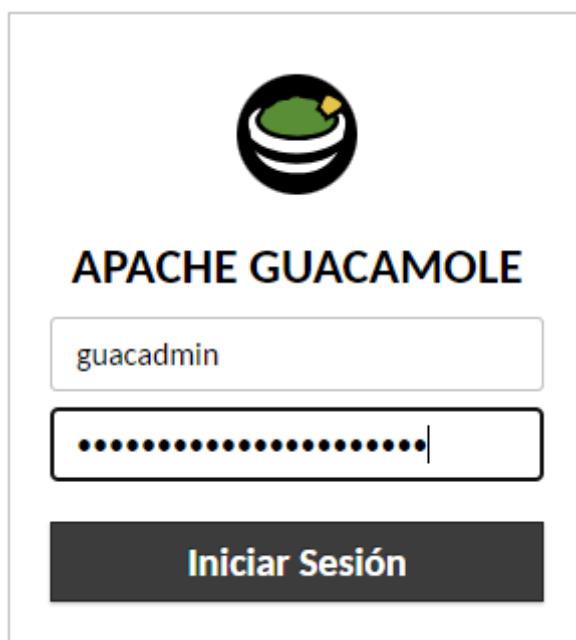
networks:

guacnetwork_compose:

NOTA: Tener mucho cuidado con los espacios, no tabulaciones, y todos los servicios deben estar perfectamente alineados, es un archivo “yaml”.

15.2 Iniciar sesión en Apache Guacamole

Login Guacamole: <https://pro.ausiasmarch.es:52230/>



He creado un reenvío de puertos del 8443 de la máquina de Kumacamole al 52230 del NAT de PfSense de mi Proxmox para poder acceder desde el navegador y así resulte más cómodo.

Please enter your authentication code to verify your identity.

567116

Continuar

Además, para añadir más “seguridad” he integrado la verificación en dos pasos con TOTP. La cual está ligada a mi Google Authenticator y se renueva el código cada 20 sec.

Sin conexiones recientes.

TODAS LAS CONEXIONES

Filtros

- IMPEL-DOWN
- LDAP
 - CrimsonLDAP
- RDP
 - LogPose-RB
- Wazuh-multi-node
 - MagellanW-D-Ashboard
 - MagellanW-I1
 - MagellanW-I2
 - magellanws
 - magellanww

Así se vería la configuración final de las conexiones en el “HOME” de Guacamole. Esta captura es posterior a la finalización del proyecto.

15.3 Administración de Apache Guacamole

CONFIGURACIONES

Sesiones Activas Historial Usuarios Grupos de Usuarios Conexiones Preferencias

Haga clic o toque un grupo de la lista inferior para gestionar ese grupo. Dependiendo de su nivel de acceso, podrá agregar/borrar grupos y cambiar los miembros y grupos del mismo.

Nuevo Grupo Filtros

Nombre de Grupo ▾

Administrators
Users

Creación de grupos “Administrators” y “Users”, el primero es privilegiado y el segundo no tiene privilegios. Con los grupos es más fácil administrar porque solo asignas los usuarios al grupo correspondiente, así no los estableces por cada usuario.

Además de configurar las conexiones que tendrán.

CONFIGURACIONES

Sesiones Activas Historial Usuarios Grupos de Usuarios Conexiones Preferencias

Haga Clic o toque un usuario de la lista inferior para gestionar ese usuario. Dependiendo de su nivel de acceso, podrá añadir/borrar usuarios y cambiar sus contraseñas.

Nuevo Usuario Filtros

Usuario ▾	Organización	Nombre completo	última conexión
guacadmin			27-05-2024 14:42:53
iermab			15-05-2024 16:01:08

Y aquí los usuarios con sus últimas conexiones para monitorizarlos.

CONFIGURACIONES

Sesiones Activas Historial Usuarios Grupos de Usuarios **Conexiones** Preferencias

Haga clic o toque en una de las conexiones de abajo para gestionar esa conexión. Dependiendo de su niv

Nueva Conexión **Nuevo Grupo** **Filtros**

- IMPEL-DOWN
 - LDAP
 - + CrimsonLDAP
 - Nueva Conexión
 - Nuevo Grupo
 - RB
 - + LogPose-RB
 - Nueva Conexión
 - Nuevo Grupo
 - Wazuh_multi-node
 - + MagellanW-D-Ashboard
 - + MagellanW-I1
 - + MagellanW-I2
 - + magellanws
 - + magellanww
 - Nueva Conexión
 - Nuevo Grupo

Aquí las conexiones que tengo configuradas, de las cuales una es gráfica, MagellanW-D-Ashboard, la máquina que contiene el navegador para entrar a la dashboard de Wazuh.

En los siguientes pasos enseñaré como configuro una conexión SSH (Porque hay varias, y solo hay que cambiar a donde apuntan “IP”) y una conexión XRDP, con todos los parámetros empleados para optimizar la transferencia de gráficos por la red.

Conexión SSH:

Nombre: magellanws
Ubicación: Wazuh_multi-node
Protocolo: SSH

Para este ejemplo muestro la configuración de magellanws (aunque también hay que crear la de magellanww, MagellanW-I1, MagellanW-I2, CrimsonLDAP, LogPose-RB). La única diferencia en esta primera captura es el nombre y la ubicación, que en la captura anterior se muestra donde deben estar.

PARÁMETROS**Red**

Nombre de Host: 172.16.1.200
Puerto: 22
Clave pública host (Base64):

Ahora hay que configurar la IP de cada servidor por el puerto donde está habilitado SSH (22).

Conexión XRDP:

Nombre: MagellanW-D-Ashboard
Ubicación: Wazuh_multi-node
Protocolo: RDP

Primero definir el Nombre de la conexión, la única conexión gráfica que tendré es MagellanW-D-Ashboard. Y el protocolo RDP.

PARÁMETROS

Red

Nombre de Host:
Puerto:

A continuación, establecer la IP y el puerto apropiado, el puerto de XRDp es el 3389, y la IP es la 172.16.1.204.

Visualización

Ancho:
Altura:

Y aquí viene uno de los mayores problemas que he tenido con la sesión gráfica, si me salgo de esta configuración la sesión gráfica tendrá una tasa de refresco muy lenta, es decir, irá con mucho retraso y parecerá que la conexión es lenta. Para solucionar este problema hay que bajar la resolución o instalar una tarjeta gráfica dedicada al servidor donde se está abriendo la sesión.

Rendimiento

- Activar Fondo de pantalla:
- Activar temas:
- Activar suavizado de fuente (ClearType):
- Activar arrastre de ventana completa:
- Activar composición de escritorio (Aero):
- Activar animaciones de menú:
- Desactivar caché bitmap:
- Desactivar caché off-screen:
- Desactivar caché glyph:

Y por temas de optimización deshabilitar caché bitmap, caché off-screen y caché glyph.

15.4 Configuración del servicio XRDП

A diferencia del protocolo SSH, en XRDП hay que ajustar unos parámetros en los ficheros de configuración para habilitar y que funcione de manera optimizada la sesión gráfica, por supuesto después de instalar el servicio xrdp.

/etc/xrdp/xrdp.ini:

```
root@magellanwashboard:/home/impeladmin# cat /etc/xrdp/xrdp.ini
[Globals]
; xrdp.ini file version number
ini_version=1

; fork a new process for each incoming connection
fork=true

; ports to listen on, number alone means listen on all interfaces
; 0.0.0.0 or :: if ipv6 is configured
; space between multiple occurrences
; ALL specified interfaces must be UP when xrdp starts, otherwise xrdp will fail to start
;

; Examples:
; port=3389
; port=unix:///tmp/xrdp.socket
; port=tcp://.:3389           127.0.0.1:3389
; port=tcp://:3389            *:3389
; port=tcp://<any ipv4 format addr>:3389   192.168.1.1:3389
; port=tcp6://.:3389          ::1:3389
; port=tcp6://:3389           *:3389
; port=tcp6://<any ipv6 format addr>:3389   {FC00:0:0:0:0:0:0:1}:3389
; port=vsock://<cid>:<port>
port=udp://0.0.0.0:3389 ←
```

Se añade la línea `port=udp://0.0.0.0:3389` por defecto viene configurado el protocolo tcp, pero yo lo cambio a udp para que genere menos tráfico de red y sea más ligero, y “0.0.0.0” para que reciba peticiones desde cualquier dirección, por el puerto rdp (3389).

```
root@magellanwashboard:/home/impeladmin# systemctl enable xrdp
Synchronizing state of xrdp.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable xrdp
root@magellanwashboard:/home/impeladmin# systemctl start xrdp
root@magellanwashboard:/home/impeladmin# systemctl status xrdp
● xrdp.service - xrdp daemon
   Loaded: loaded /lib/systemd/system/xrdp.service; enabled; vendor preset: enabled
   Active: active (running) since Mon 2024-05-27 09:02:51 CEST; 9h ago
     Docs: man:xrdp(8)
           man:xrdp.ini(5)
     Main PID: 789 (xrdp)
        Tasks: 1 (limit: 4534)
      Memory: 1.4M
        CPU: 60ms
       CGroup: /system.slice/xrdp.service
               └─789 /usr/sbin/xrdp

may 27 09:02:52 magellanwashboard xrdp[789]: [INFO ] send buffer set to 425984 bytes
may 27 09:02:52 magellanwashboard xrdp[789]: [INFO ] address [0.0.0.0] port [0] mode 1
may 27 09:02:52 magellanwashboard xrdp[789]: [INFO ] listening to port 0 on 0.0.0.0
may 27 09:02:52 magellanwashboard xrdp[789]: [INFO ] setting send buffer to 8388608 bytes
may 27 09:02:52 magellanwashboard xrdp[789]: [INFO ] send buffer set to 425984 bytes
may 27 09:02:52 magellanwashboard xrdp[789]: [INFO ] address [0.0.0.0] port [3389] mode 1
may 27 09:02:52 magellanwashboard xrdp[789]: [INFO ] listening to port 3389 on 0.0.0.0
may 27 09:02:52 magellanwashboard xrdp[789]: [INFO ] setting send buffer to 8388608 bytes
may 27 09:02:52 magellanwashboard xrdp[789]: [INFO ] send buffer set to 425984 bytes
may 27 09:02:52 magellanwashboard xrdp[789]: [INFO ] xrdp_listen_pp done
```

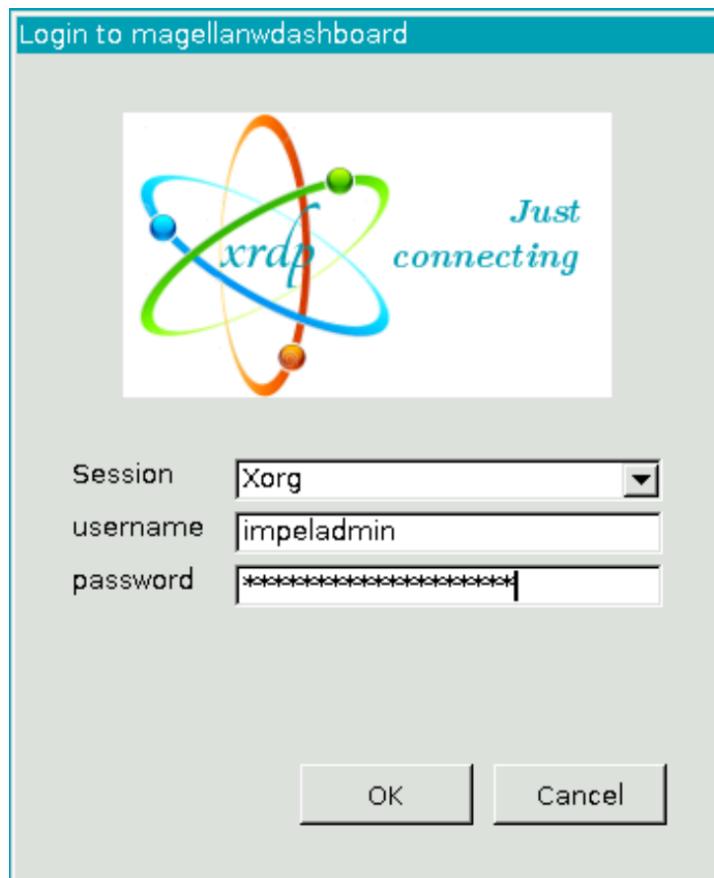
Por último habilitar, iniciar y comprobar el servicio xrdp, y está todo correcto.

16. Acceder remotamente a MagellanW-D-Ashboard

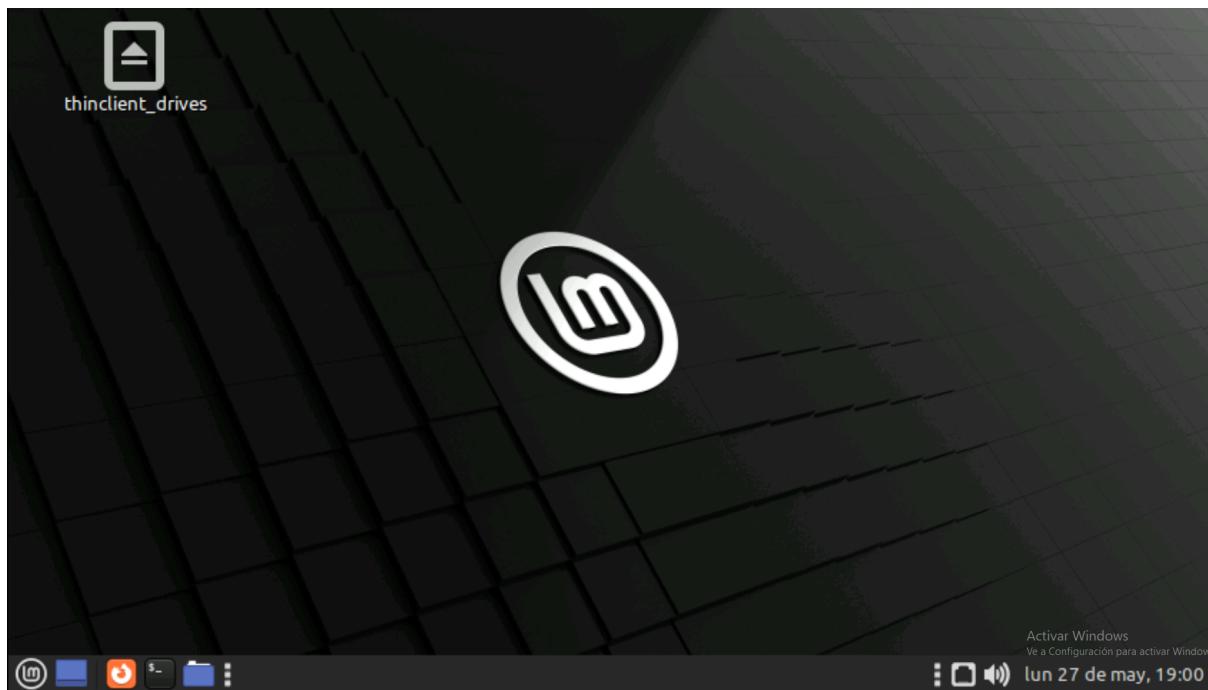
16.1 Iniciar sesión xRDP



Seleccionar la sesión de MagellanW-D-Ashboard.



Ahora introducir las credenciales de un usuario existente en la máquina objetivo.



Y ya estaremos administrando remotamente.

16.2 Monitorizar inicios de sesión

CONFIGURACIONES						
Sesiones Activas	Historial	Usuarios	Grupos de Usuarios	Conexiones	Preferencias	
Aqui se detalla el historial de las últimas conexiones y se pueden ordenar haciendo clic en los encabezados de la columna. Para buscar un registro específico, introduzca la cadena de texto a filtrar y haga clic en "Buscar". Solo se listarán los registros que coincidan con el filtro introducido.						
<input type="text"/> Filtros Buscar Descargar						
Usuario	Activo Desde ▾	Duración	Nombre de conexión	Host Remoto	Logs	
guacadmin	27-05-2024 18:59:05	4.3 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	27-05-2024 12:32:56	5.7 minutes	LogPose-RB	172.19.0.3		
guacadmin	27-05-2024 08:39:33	1.7 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	26-05-2024 22:42:59	6.3 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	25-05-2024 21:13:38	25.2 seconds	LogPose-RB	172.19.0.3		
guacadmin	25-05-2024 21:13:16	17.1 seconds	LogPose-RB	172.19.0.3		
guacadmin	25-05-2024 21:02:25	10.7 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	25-05-2024 20:59:52	2.5 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	25-05-2024 20:57:21	2.5 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	25-05-2024 17:55:56	22.9 seconds	LogPose-RB	172.19.0.3		
guacadmin	25-05-2024 17:48:47	7.1 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	25-05-2024 15:31:08	1.8 hours	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	25-05-2024 15:24:13	3 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	25-05-2024 15:23:51	18.2 seconds	LogPose-RB	172.19.0.3		
guacadmin	24-05-2024 13:36:36	1.2 hours	magellanws	172.19.0.3		
guacadmin	24-05-2024 11:30:44	20.4 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	24-05-2024 11:29:10	1.5 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	24-05-2024 11:27:57	1.2 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	24-05-2024 11:26:34	1.4 minutes	MagellanW-D-Ashboard	172.19.0.3		
guacadmin	24-05-2024 11:19:51	5.7 minutes	MagellanW-D-Ashboard	172.19.0.3	Activar Windows Ve a Configuración para activar Windows	
guacadmin	24-05-2024 11:02:55	16.8 minutes	magellanws	172.19.0.3		

Este apartado es muy interesante y recomiendo entrar varias veces al día para saber quién ha iniciado sesión.

17. Breve introducción a LDAP

LDAP es un protocolo de acceso a directorios que se utiliza para mantener y acceder a información distribuida en una red. Es comúnmente empleado para autenticación y autorización de usuarios, almacenamiento de información de directorio y para integrar servicios en una red. Se utiliza para gestionar credenciales de usuario, controlar el acceso a recursos, almacenar información de contacto y configuraciones de red, y centralizar la administración de identidades y políticas de seguridad.

LDAP se basa en un modelo cliente-servidor, donde los clientes LDAP realizan solicitudes al servidor LDAP para acceder y manipular la información del directorio. Utiliza un enfoque ligero y eficiente para acceder a la información del directorio, lo que lo hace adecuado para su implementación en una amplia variedad de entornos de red, desde pequeñas empresas hasta grandes organizaciones y proveedores de servicios de Internet. Además, LDAP está respaldado por una amplia gama de herramientas y bibliotecas de software que facilitan su implementación y administración.

18. Desplegando dominio de LDAP

18.1 Configuración de red

```
root@crimsonldap:/home/impeladmin# cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens18:
      dhcp4: false
      addresses: [172.16.1.20/24]
      gateway4: 172.16.1.254
  version: 2
```

Para empezar hay que configurar la red de la máquina para que tenga comunicación con los demás equipos de la red. Como es un Ubuntu server sin entorno gráfico el configurador de la red es **netplan**, y por consiguiente hay que editar el fichero de configuración “/etc/netplan/00-installer-config.yaml”. En mi caso le asigné la dirección de red “172.16.1.20/24”, desactivando el descubrimiento de servidores DHCP porque quiero que sea una IP fija.

```
root@crimsonldap:/home/impeladmin# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
  link/loopback brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
  link/ether 6a:32:b4:06:45:7c brd ff:ff:ff:ff:ff:ff
  altname enp0s18
  inet 172.16.1.20/24 brd 172.16.1.255 scope global ens18
    valid_lft forever preferred_lft forever
  inet6 fe80::6832:b4ff:fe06:457c/64 scope link
    valid_lft forever preferred_lft forever
```

Después de guardar los cambios y ejecutar el comando “netplan try” para aplicar la configuración realicé un “ip a” para comprobar que se cambió correctamente.

18.2 Configuración de hostname

```
root@crimsonldap:/home/impeladmin# hostname -f
crimsonldap.impeldown.com
root@crimsonldap:/home/impeladmin# cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 crimsonldap.impeldown.com crimsonldap
172.16.1.20 crimsonldap.impeldown.com crimsonldap
```

Es obligatorio tener configurado el nombre de la máquina correctamente. Ya que más tarde cuando configuremos el dominio y unamos los hosts a él tendremos que resolver al nombre que configuremos.

Con “hostname -f” compruebo que el nombre del host es el que yo he asignado (crimsonldap) y en el fichero “/etc/hosts” estoy apuntando a mi propio host de manera local y con la IP asignada (3^a línea).

18.3 Instalación de dependencias

```
root@crimsonldap:/home/impeladmin# apt install slapd ldap-utils -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ldap-utils ya está en su versión más reciente (2.5.17+dfsg-0ubuntu0.22.04.1).
slapd ya está en su versión más reciente (2.5.17+dfsg-0ubuntu0.22.04.1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
```

Una vez configurado todo lo anterior, para comenzar con el despliegue del dominio LDAP, hay que descargar las utilidades que se van a necesitar para la creación y configuración del dominio:

`# apt install slapd ldap-utils -y`

La instalación se realiza desde los repositorios de Ubuntu, empleando **apt** se instalará **slapd** y **ldap-utils**.

```
root@crimsonldap:/home/impeladmin# echo "comandos del servicio slapd"; dpkg -L slapd | grep bin/ ; echo "Comandos de las utilidades de ldap"; dpkg -L ldap-util
s | grep bin/
comandos del servicio slapd
/usr/sbin/slappac
/usr/sbin/slappadd
/usr/sbin/slappauth
/usr/sbin/slappcat
/usr/sbin/slapp
/usr/sbin/slappdn
/usr/sbin/slappindex
/usr/sbin/slappmodify
/usr/sbin/slappasswd
/usr/sbin/slapschema
/usr/sbin/slapptest
Comandos de las utilidades de ldap
/usr/bin/dapcompare
/usr/bin/dapdelete
/usr/bin/dapexop
/usr/bin/dapmodify
/usr/bin/dapmodrdn
/usr/bin/dappasswd
/usr/bin/dapsearch
/usr/bin/dapuri
/usr/bin/dapuhobjami
/usr/bin/dapadd
```

Para verificar la instalación ejecutar el comando:

```
`# echo "comandos del servicio slapd"; dpkg -L slapd | grep bin/ ; echo "comandos de las utilidades de ldap" ; dpkg -L ldap-utils | grep bin/`
```

18.4 Verificación del dominio

```
root@crimsonldap:/home/impeleadmin# sudo slapcat
dn: dc=impeardown,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: impeardown.com
dc: impeardown
structuralObjectClass: organization
entryUUID: 5ddaa0392-9ca2-103e-9df1-15655e1d3f25
creatorsName: cn=admin,dc=impeardown,dc=com
createTimestamp: 202405020736002
entryCSN: 20240502073600.426386Z#000000#000#000000
modifiersName: cn=admin,dc=impeardown,dc=com
modifyTimestamp: 202405020736002
```

Ejecutando **slapcat** se puede comprobar la configuración inicial del dominio. La salida del comando es la recolección de los datos almacenados en el fichero “/etc/ldap/slapd.d/cn=config.ldif”.

```
root@crimsonldap:/home/impeleadmin# systemctl status slapd
● slapd.service - LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol)
   Loaded: loaded (/etc/init.d/slapd; generated)
   Drop-In: /usr/lib/systemd/system/slapd.service.d
             └─slapd-remain-after-exit.conf
     Active: active (running) since Thu 2024-05-02 07:36:01 UTC; 6min ago
       Docs: man:systemd-sysv-generator(8)
   Process: 1431 ExecStart=/etc/init.d/slapd start (code=exited, status=0/SUCCESS)
   Tasks: 3 (limit: 2220)
  Memory: 3.3M
    CPU: 43ms
   CGroup: /system.slice/slapd.service
           └─1438 /usr/sbin/slapd -h "ldap:/// ldapi://" -g openldap -u openldap -F /etc/ldap/slapd.d

may 02 07:36:01 crimsonldap systemd[1]: Starting LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol).
may 02 07:36:01 crimsonldap slapd[1431]: * Starting OpenLDAP slapd
may 02 07:36:01 crimsonldap slapd[1431]: @(#) $OpenLDAP: slapd 2.5.17+dfsg-0ubuntu0.22.04.1 (Feb 9 2024 20:12:07) $
                                         Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
may 02 07:36:01 crimsonldap slapd[1438]: slapd starting
may 02 07:36:01 crimsonldap slapd[1431]: ...done.
may 02 07:36:01 crimsonldap systemd[1]: Started LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol).
```

También se puede consultar el estado del servicio de **slapd.service** con el comando:

```
`# systemctl status slapd`
```

18.5 Historia y estructura de LDAP

Antiguamente la configuración residía en el fichero “/etc/slapd.conf” y era estática. Si quisiéramos cambiarla en aquel momento debíamos reiniciar el servicio para que los cambios tuvieran efecto. Y en la mayoría de los casos no se podían realizar reinicios tan agresivos.

A partir de la versión 2.3 OpenLDAP permite la configuración dinámica del demonio slapd. Esta configuración RTC (Run Time Configuration) se almacena en un directorio especial con un DIT (Directory Information Tree) predefinido con raíz cn=config. Ahí se guarda la información global como la definición de los esquemas, backends y bases de datos, entre otros.

En máquinas como Ubuntu o derivados de Debian, se almacena en el directorio “/etc/ldap/slapd.d”

```
root@crimsonldap:/home/impeladmin# tree /etc/ldap/slapd.d/
/etc/ldap/slapd.d/
└── cn=config
    ├── cn=module{0}.ldif
    ├── cn=schema
    │   ├── cn={0}core.ldif
    │   ├── cn={1}cosine.ldif
    │   ├── cn={2}nis.ldif
    │   └── cn={3}inetorgperson.ldif
    ├── cn=schema.ldif
    ├── olcDatabase={0}config.ldif
    ├── olcDatabase={-1}frontend.ldif
    └── olcDatabase={1}mdb.ldif
    └── cn=config.ldif

2 directories, 10 files
```

18.6 Configuración del servicio de directorio

Para comenzar con la configuración inicial ejecutar el comando:

dpkg-reconfigure slapd

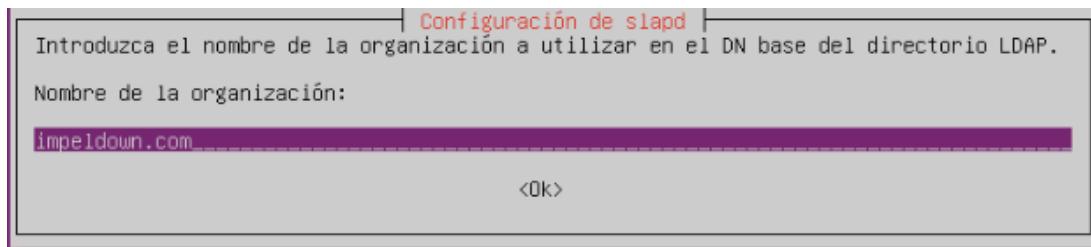
NOTA: Este comando se puede hacer las veces que haga falta.

DNS:



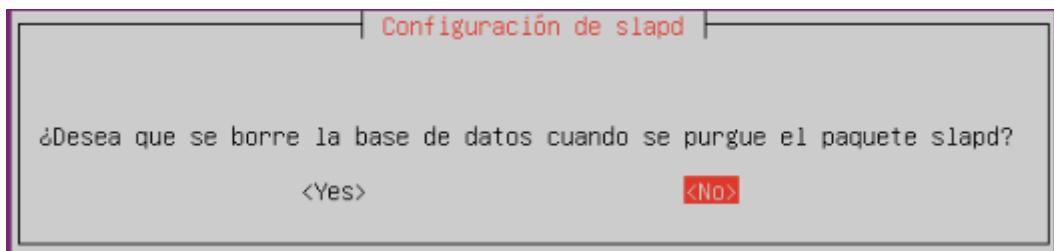
Este será el DNS (Domain Name System) “impeldown.com”, es decir, el nombre de dominio al cual resolverán los hosts.

Organización:



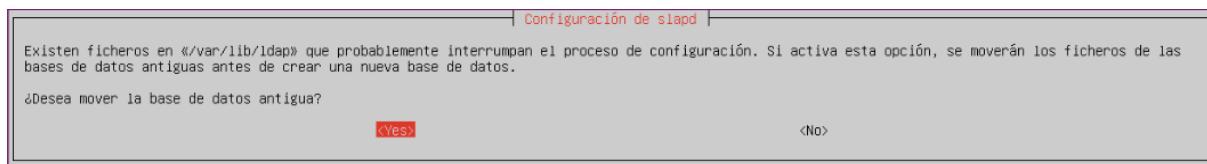
El nombre de la organización correspondiente, en mi caso lo mismo.

Borrar BD:



En este paso yo he elegido que no quiero que se borre la base de datos porque en caso de que quiera eliminar el paquete de slapd me gustaría guardar la estructura de mi dominio. Pero este paso puede depender de las políticas de cada organización.

Mover BD antigua:



He elegido que se muevan los ficheros de bases de datos antiguas cuando se cree una nueva base de datos.

18.7 Generando la estructura de dominio

Para generar la estructura se emplearán Scripts en **Idif** que es el método por el cual se crean UOs, usuarios, grupos, etc ... Los cuales se almacenarán en la ruta "/etc/ldap/ldifs" para establecer un orden.

UOs:

```
root@crimsonldap:/etc/ldap/ldifs# cat UnidadesOrganizativas.ldif
#OU People
dn: ou=People,dc=impeldorf,dc=com
objectClass: organizationalUnit
ou: People

#OU Group
dn: ou=Group,dc=impeldorf,dc=com
objectClass: organizationalUnit
ou: Group
```

El primer paso será crear las UOs (Unidades Organizativas) "People" y "Group".

Las Unidades Organizativas (OU, por sus siglas en inglés de "Organizational Units") son contenedores lógicos dentro de un directorio LDAP que se utilizan para organizar y estructurar la información de manera jerárquica. Son similares a carpetas en un sistema de archivos, donde se pueden agrupar objetos relacionados.

```
root@crimsonldap:/etc/ldap/ldifs# ldapadd -x -D cn=admin,dc=impeldorf,dc=com -W -f UnidadesOrganizativas.ldif
Enter LDAP Password:
adding new entry "ou=People,dc=impeldorf,dc=com"
adding new entry "ou=Group,dc=impeldorf,dc=com"
```

El comando utilizado para ejecutar el ldif creado anteriormente es:

```
# ldapadd -x -D cn=admin,dc=impeldorf,dc=com -W -f UnidadesOrganizativas.ldif
```

- **ldapadd:** Es el comando que se utiliza para agregar datos al directorio LDAP.
- **-x:** Especifica que se utilizará una autenticación simple. Esto significa que se utilizará una autenticación simple en lugar de una autenticación SASL (Simple Authentication and Security Layer).
- **-D cn=admin,dc=impeldorf,dc=com:** Especifica la identidad (DN, Distinguished Name) que se utilizará para autenticarse ante el servidor LDAP. En este caso, se está utilizando el DN del usuario administrador "admin" en el dominio "dc=impeldorf,dc=com".
- **-W:** Sigue al comando para solicitar la contraseña de forma interactiva. Despues de ingresar el comando, el usuario deberá proporcionar la contraseña del usuario administrativo.
- **-f UnidadesOrganizativas.ldif:** Especifica el archivo LDIF (LDAP Data Interchange Format) que contiene los datos que se van a agregar al directorio LDAP. En este caso, el archivo se llama "UnidadesOrganizativas.ldif".

Breve comprobación de que se han creado exitosamente las OUs:

```
root@crimsonldap:/etc/ldap/ldifs# slapcat
dn: dc=impeldown,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: impeldown.com
dc: impeldown
structuralObjectClass: organization
entryUUID: 9c624bb2-9ca5-103e-8102-c5db629ff0b6
creatorsName: cn=admin,dc=impeldown,dc=com
createTimestamp: 20240502075913Z
entryCSN: 20240502075913.828429Z#000000#000#000000
modifiersName: cn=admin,dc=impeldown,dc=com
modifyTimestamp: 20240502075913Z
dn: ou=People,dc=impeldown,dc=com 1
objectClass: organizationalUnit
ou: People
structuralObjectClass: organizationalUnit
entryUUID: 38ee3248-9cab-103e-8590-0b17957384f1
creatorsName: cn=admin,dc=impeldown,dc=com
createTimestamp: 20240502083923Z
entryCSN: 20240502083923.953537Z#000000#000#000000
modifiersName: cn=admin,dc=impeldown,dc=com
modifyTimestamp: 20240502083923Z
dn: ou=Group,dc=impeldown,dc=com 2
objectClass: organizationalUnit
ou: Group
structuralObjectClass: organizationalUnit
entryUUID: 38efe46c-9cab-103e-8591-0b17957384f1
creatorsName: cn=admin,dc=impeldown,dc=com
createTimestamp: 20240502083923Z
entryCSN: 20240502083923.964694Z#000000#000#000000
modifiersName: cn=admin,dc=impeldown,dc=com
modifyTimestamp: 20240502083923Z
```

Grupos:

```
root@crimsonldap:/etc/ldap/ldifs# cat Grupos.ldif
#Grupo de admins
dn: cn=admins,ou=Group,dc=impeldown,dc=com
objectClass: posixGroup
cn: admins
gidNumber: 5001

#Grupo de users
dn: cn=users,ou=Group,dc=impeldown,dc=com
objectClass: posixGroup
cn: users
gidNumber: 5002
```

El siguiente paso ha sido crear otro Script Ldif llamado “Grupos.ldif” con dos grupos en la Unidad Organizativa “Group”, “admins” y “users”. En el de admins se pondrá, de momento, un usuario administrador de dominio y en el de usuarios habrán más adelante usuarios de dominio desprivilegiados, pero durante el despliegue de dominio sólo habrá uno, el administrador de dominio.

El grupo de admins tendrá el gidNumber en 5001 y el grupo de users tendrá el gidNumber 5002.

```
root@crimsonldap:/etc/ldap/ldifs# ldapadd -x -D cn=admin,dc=impeldown,dc=com -W -f Grupos.ldif
Enter LDAP Password:
adding new entry "cn=admins,ou=Group,dc=impeldown,dc=com"
adding new entry "cn=users,ou=Group,dc=impeldown,dc=com"
```

El comando utilizado para ejecutar el ldif creado anteriormente es:

```
`# ldapadd -x -D cn=admin,dc=impeldown,dc=com -W -f Grupos.ldif`
```

El cual tiene las mismas características que el anterior para generar las OUs con la diferencia de que ejecutará el nuevo Script.

NOTA: En el siguiente paso cuando se agregue el usuario de dominio se agregará al gidNumber de admins, no habrá ningún campo de “cn=admins”.

Usuario:

```
root@crimsonldap:/home/impeladmin# cat /etc/ldap/ldifs/Usuarios.ldif
#Usuario administrador de dominio
dn: uid=emperor,ou=People,dc=impeldown,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: emperor
sn: op
givenName: emperor
cn: emperor
displayName: emperor
uidNumber: 10001
gidNumber: 5001
userPassword: ##_Y0nko99_$$
gecos: emperor
loginShell: /bin/bash
homeDirectory: /home/admins/emperor
```

En este paso estoy creando al nuevo usuario de dominio que estará en el grupo de administradores y he nombrado como “emperor”.

En la imagen se puede apreciar que están definidas características muy importantes como su uid, el cn, el uidNumber, gidNumber, contraseña, tipo de shell y ruta de su “home”.

```
root@crimsonldap:/etc/ldap/ldifs# ldapadd -x -D cn=admin,dc=impeldown,dc=com -W -f Usuarios.ldif
Enter LDAP Password:
adding new entry "uid=emperor,ou=People,dc=impeldown,dc=com"
adding new entry "uid=buggy,ou=People,dc=impeldown,dc=com"
```

De nuevo, para ejecutar el Script ldif con el nuevo usuario es necesario correr el mismo comando que las veces anteriores con la diferencia del nombre de ldif a ejecutar:

```
`# ldapadd -x -D cn=admin,dc=impeldown,dc=com -W -f Usuarios.ldif`
```

Y se puede ver en la salida del comando como han sido realizados los cambios exitosamente.

Como ya dije en la creación de los Grupos, más adelante crearé usuarios de dominio desprivilegiados. De momento con este usuario será suficiente para probar.

18.8 Instalar administrador gráfico de LDAP (LAM)

```
root@crimsonldap:/home/impeladmin# apt install ldap-account-manager
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ldap-account-manager ya está en su versión más reciente (7.7-1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
```

Debido a la incomodidad que puede resultar la administración por consola, he decidido instalar una herramienta gráfica para mayor comodidad, de cara a consultas, modificaciones o creaciones de elementos. Además, esta herramienta está en formato HTML, lo que la hace más versátil porque se puede usar desde cualquier ubicación.

Para poder instalarla utilizar el comando:

```
'# apt install ldap-account-manager'
```

Después se instalarán todos los paquetes necesarios. Cuando termine la instalación se podrá acceder desde la URL “<http://172.16.1.20/lam>”.

19. Unión de Kumacamole al dominio

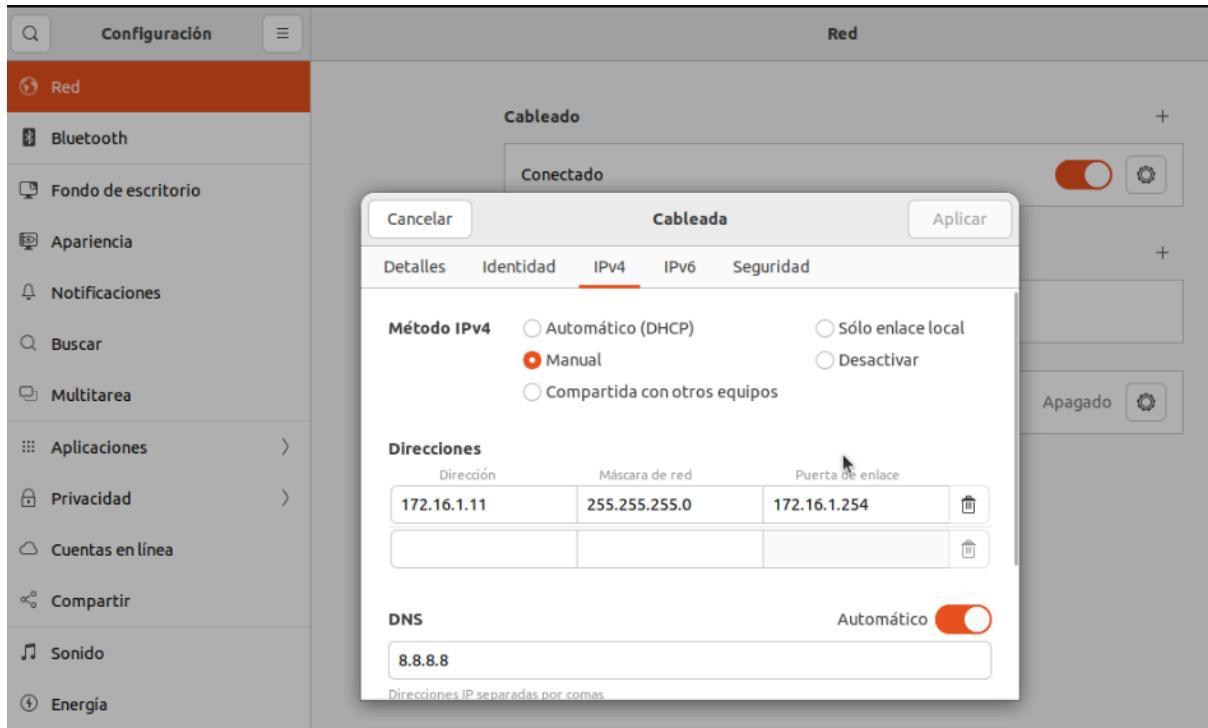
19.1 Configuración de hostname

Una de las tareas más importantes después de crear el servidor LDAP es unir los clientes al dominio. Para ello primero que nada uniré a Kumacamole. La razón principal es porque al ser por entorno gráfico se puede administrar el servidor LDAP de manera más cómoda desde el LAM instalado.

```
impeladmin@kumacamole:~/Escritorio$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      kumacamole.impeldown.com kumacamole
172.16.1.11     kumacamole.impeldown.com kumacamole
```

Lo que se añade son las líneas de configuración del hostname para que resuelva la máquina al dominio. Donde aparecen dos direcciones, una local y otra la propia IP, y las líneas son “**Nombre_de_maquina.servidor.ldap Nombre_de_maquina**”.

19.2 Configuración de red

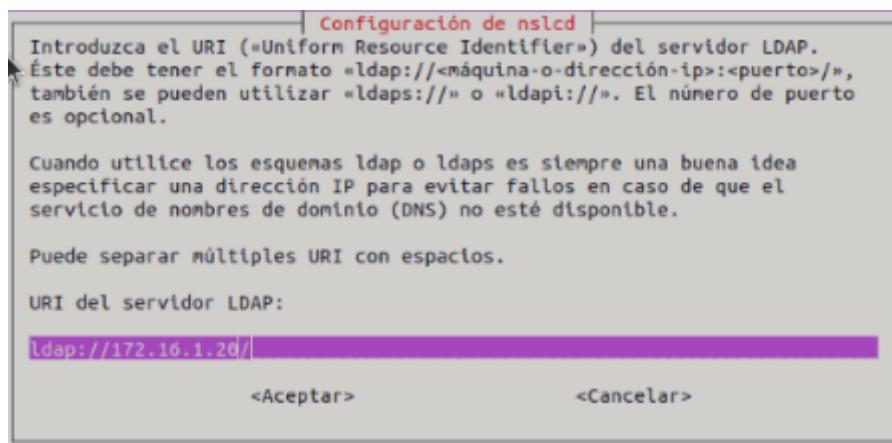


La configuración de red se mantiene igual que se configuró anteriormente, porque está en la misma dirección de red que el servidor LDAP.

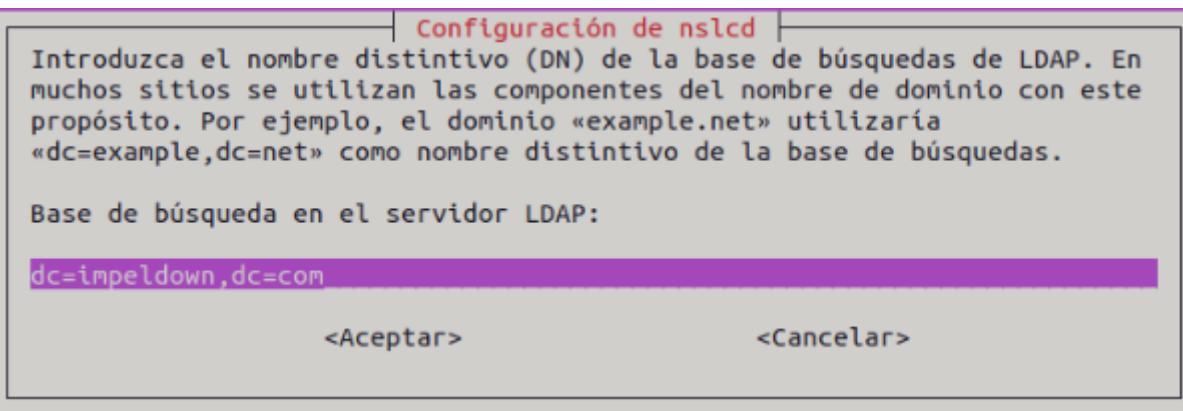
19.3 Instalación del software

El comando que se utilizará para comenzar la configuración de la unión del cliente al servidor LDAP es:

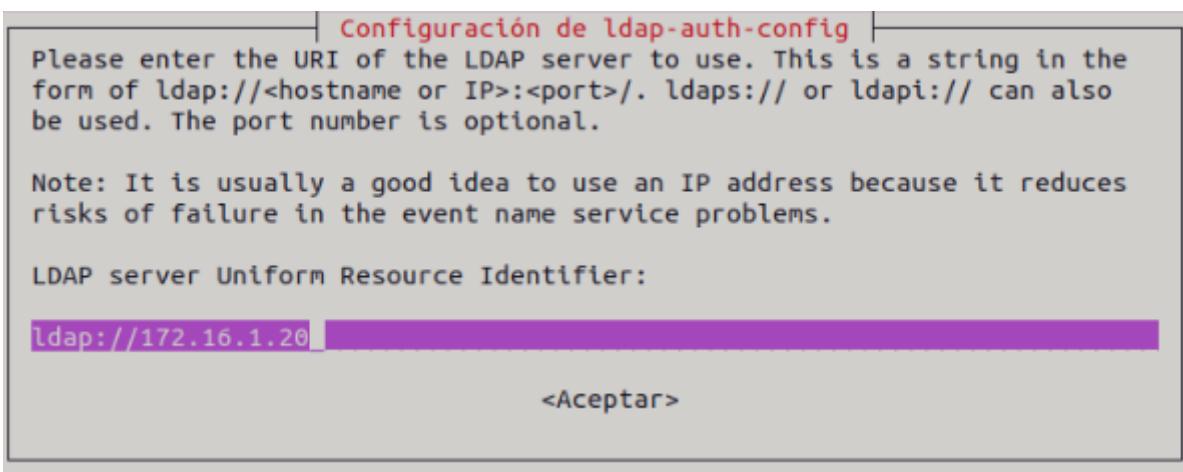
```
# apt install libnss-ldap nslcd
```



De este modo se abre el primer paso del instalador, donde habrá que apuntar al servidor LDAP.



A continuación hay que establecer cuál será la base de búsquedas de LDAP. En mi caso es “dc=impeldown,dc=com”.



Ahora hay que apuntar a la dirección IP de mi servidor LDAP la cual es “172.16.1.20”.

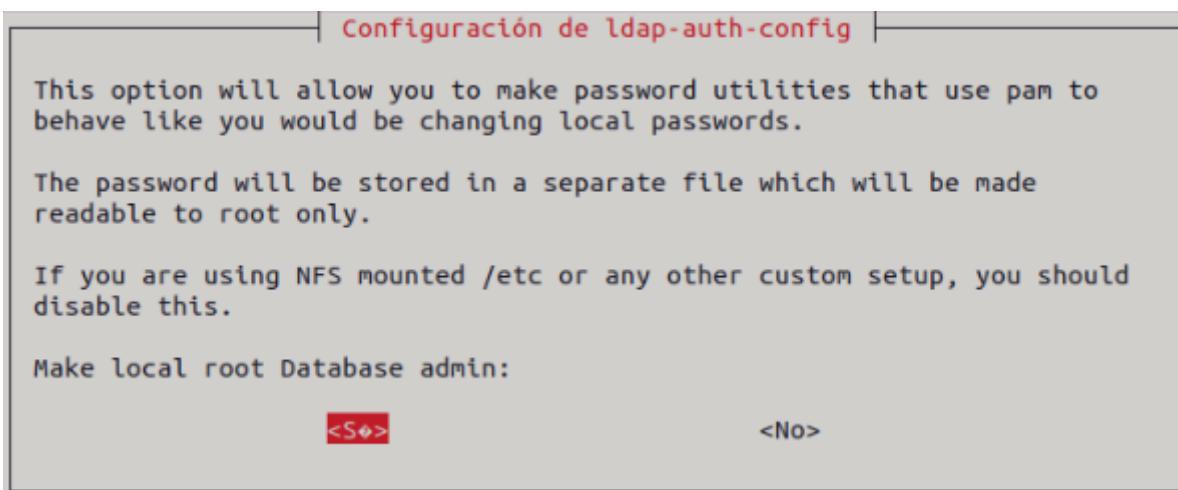
OJO: Por defecto viene como “`ldapi://IP_server/`” cambiar a “`ldap://IP_server`”, de lo contrario no funcionará.



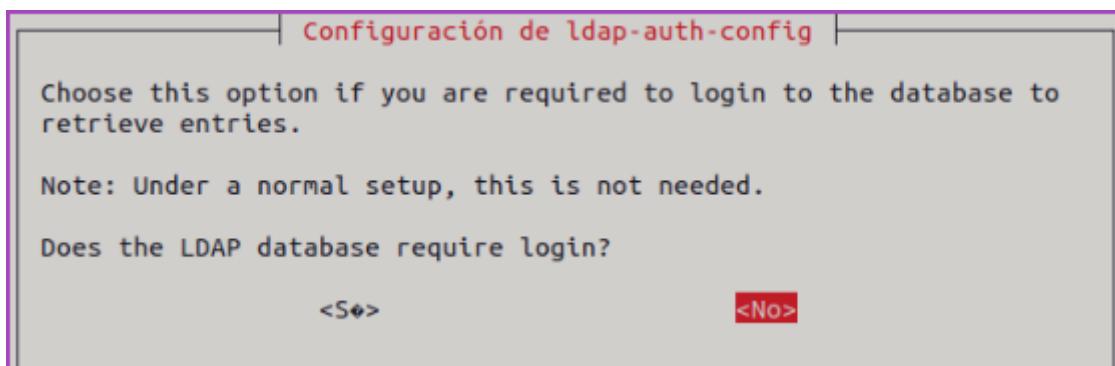
Y ahora volver a introducir el dominio al que resolver para el paquete `ldap-auth-config` “dc=impeldown, dc=com”.



La versión del paquete será la 3 que es la más nueva y con la que será compatible nuestro sistema.



En este paso se pregunta si se quiere convertir la base de datos raíz en admin. Es decir, si queremos cambiar la contraseña será posible con root. Esta será guardada en otra ruta distinta a la del esquema de nuestra organización.



Ahora pregunta si es necesario hacer login para recibir entradas de la base de datos. Yo he marcado que no quiero.

20. Revisión de la configuración

20.1 Fichero “/etc/ldap.conf”

El siguiente paso es comprobar el fichero de “/etc/ldap.conf” el cual estará correctamente excepto la “pam_password” que la cambio a crypt.

```
# The distinguished name of the search base.
base dc=impeldown,dc=com
```

El nombre distintivo de la base de búsqueda, el mismo configurado anteriormente “dc=impeldown,dc=com”.

```
# Another way to specify your LDAP server is to provide an
uri ldap://172.16.1.20/
```

La uri del servidor LDAP “ldap://172.16.1.20/”.

```
# The LDAP version to use (defaults to 3
# if supported by client library)
ldap_version 3
```

La versión de LDAP, la 3.

```
# The distinguished name to bind to the server with
# if the effective user ID is root. Password is
# stored in /etc/ldap.secret (mode 600)
rootbinddn cn=admin,dc=impeldown,dc=com
```

La contraseña se guardará en el fichero “/etc/ldap.secret” y el usuario root será admin.

```
# HEADS UP: the pam_crypt, pam_nds_passwd,
# and pam_ad_passwd options are no
# longer supported.
#
# Do not hash the password at all; presume
# the directory server will do it, if
# necessary. This is the default.
pam_password crypt
```

Y el tipo de cifrado lo cambio de MD5 a crypt, ya que MD5 es muy inseguro y crypt es compatible con el sistema y es bastante fiable al mismo tiempo.

20.2 Fichero “/etc/nsswitch.conf”

Hay que configurar el fichero de “/etc/nsswitch.conf” para que nuestra máquina tenga acceso a los usuarios del servidor LDAP. Para ello se añade la palabra “ldap” detrás de files en las primeras 3 líneas. De esta manera, se logra que en primer lugar priorice los usuarios locales y después los de dominio.

```
root@kumacamole:/home/impeladmin/Escritorio# cat /etc/nsswitch.conf
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'glibc-doc-reference' and 'info' packages installed, try:
# 'info libc "Name Service Switch"' for information about this file.

passwd:      files ldap systemd
group:       files ldap systemd
shadow:      files ldap
gshadow:     files

hosts:        files mdns4_minimal [NOTFOUND=return] dns
networks:    files

protocols:   db files
services:    db files
ethers:      db files
rpc:         db files
```

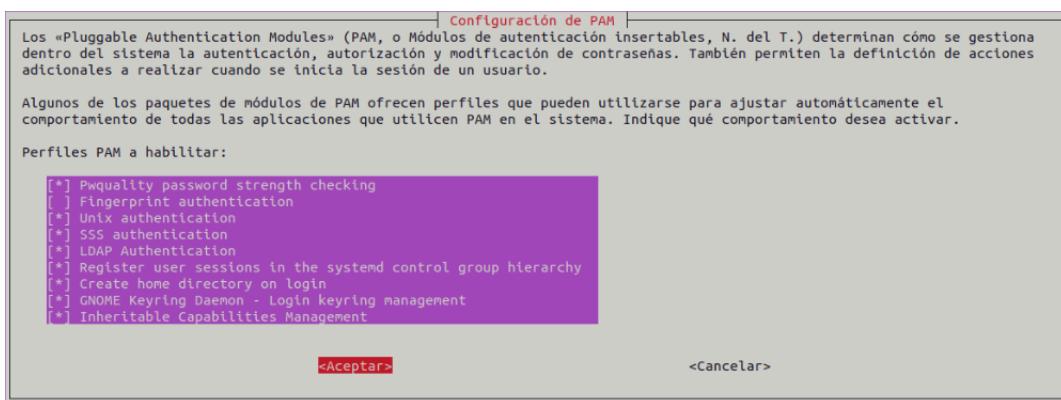
Así es como queda el fichero después de editarse.

20.3 Comando “pam-auth-update”

Ejecutar el comando “pam-auth-update” para indicar los servicios que va a usar el sistema para autenticar a los usuarios:

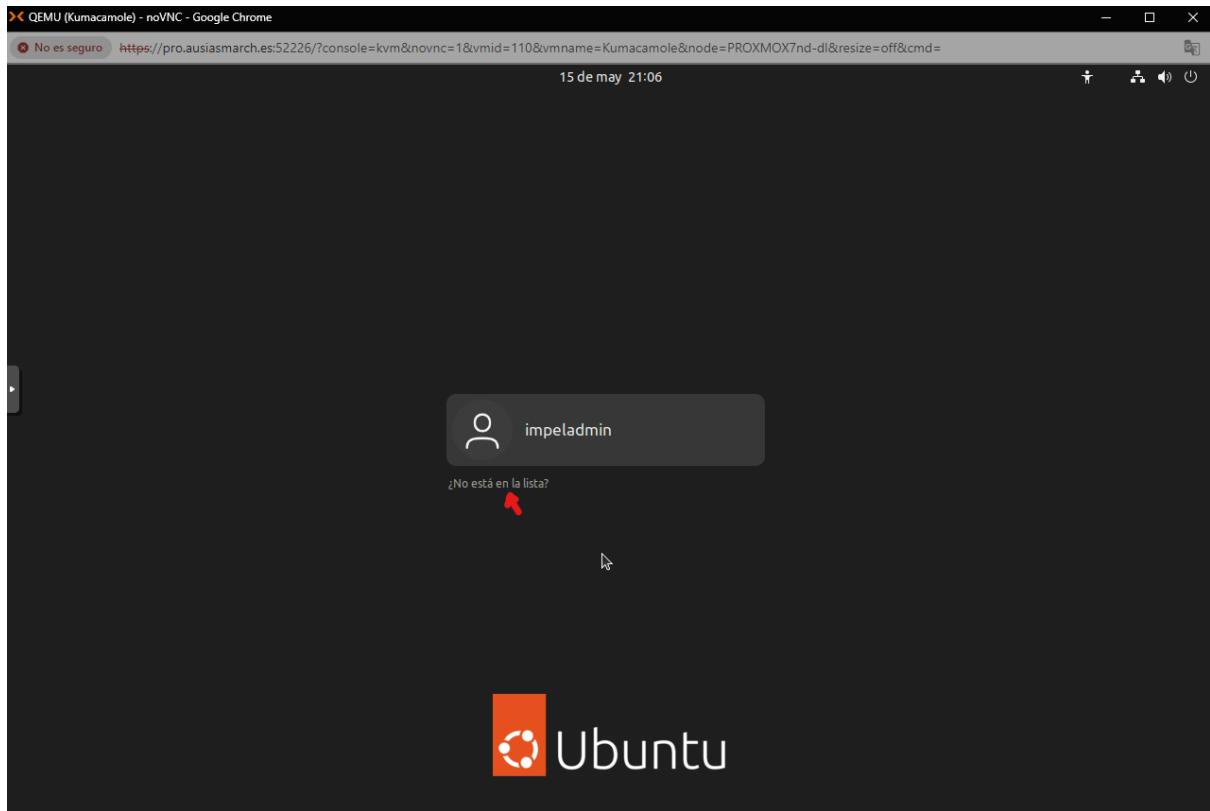
`# pam-auth-update`

Cuando se abra la ventana con los parámetros de instalación elegir marcar “Create home directory on login” para que al iniciar sesión el usuario automáticamente cree su “home”.

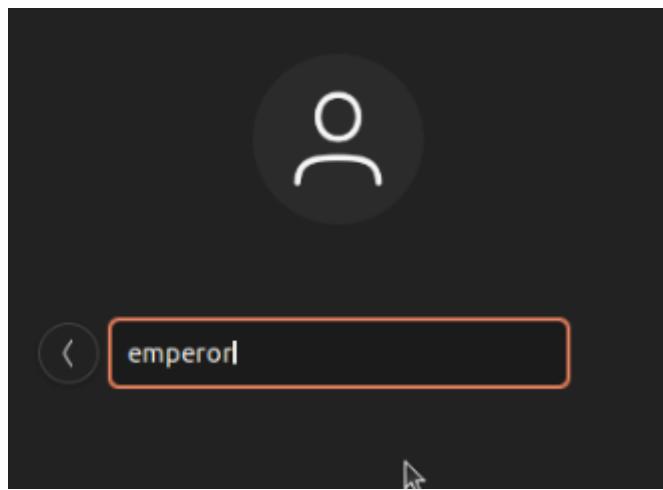


20.4 Iniciar sesión gráfica

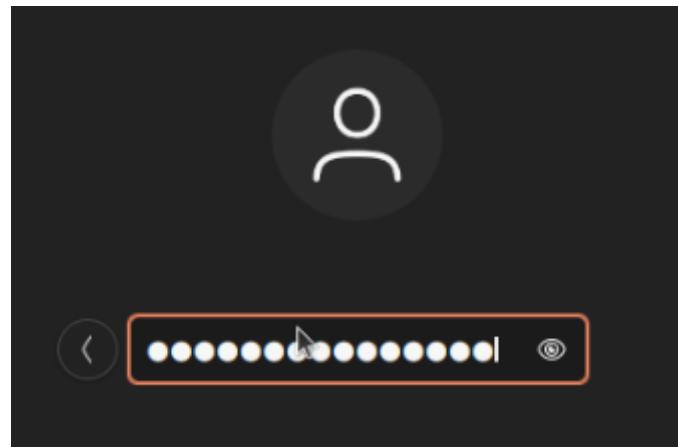
La prueba de que realmente lo hemos unido al dominio. Pero antes de intentar hacer login como usuario de dominio lo primero será **reiniciar** el equipo.



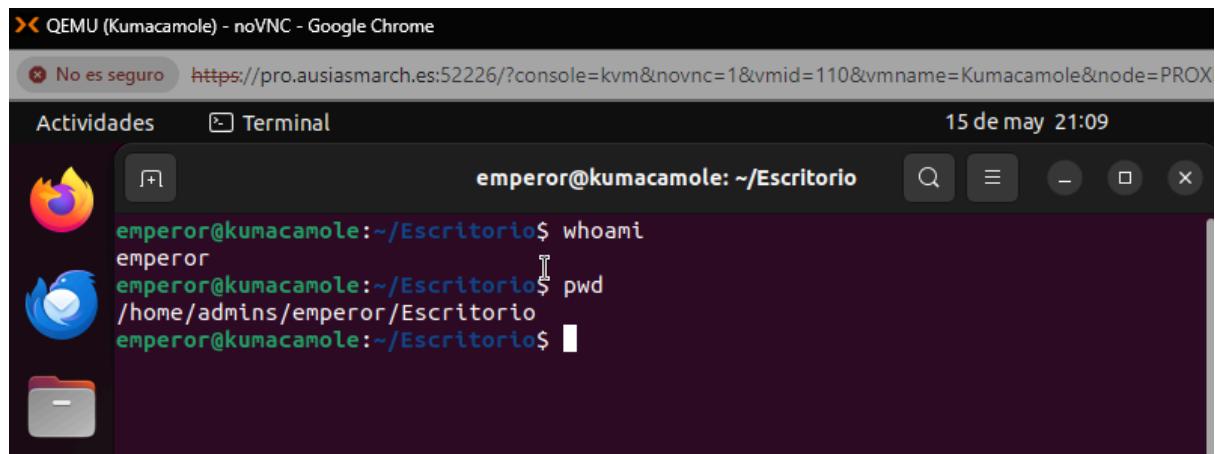
El primer paso será hacer clic donde dice “No está en la lista?” porque es un usuario que no ha iniciado sesión anteriormente.



Y a continuación proporcionar el nombre de usuario.



Y después la contraseña.



Y ya está, una breve comprobación de que soy el usuario de dominio “emperor” y la ruta de mi carpeta personal se ha creado automáticamente: “/home/admins/emperor”.

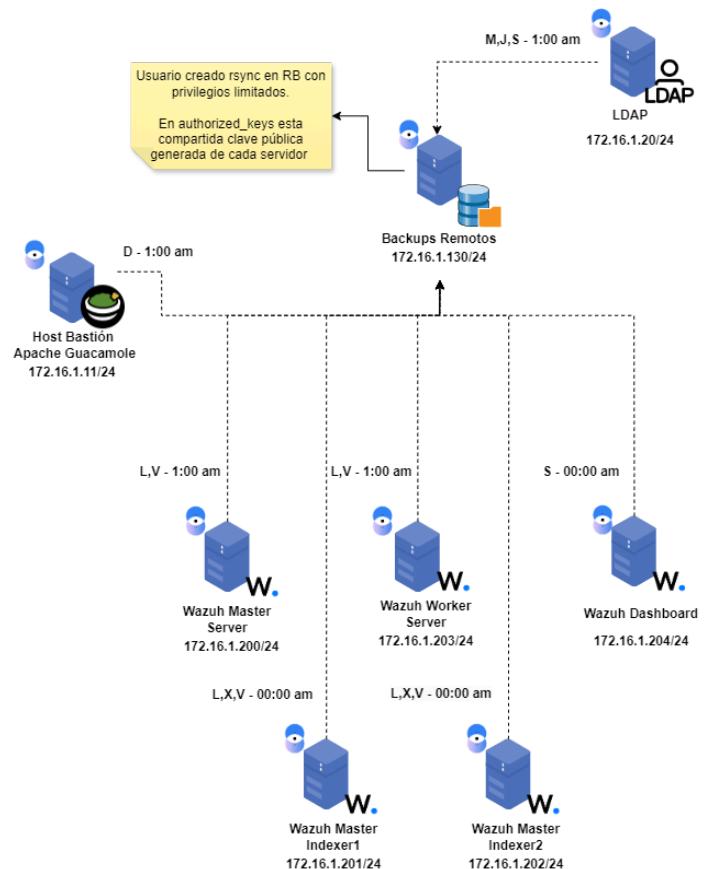
21. Breve introducción a Remote Backups Server

Tener un servidor de backups remoto en la infraestructura es crucial por varias razones que benefician la seguridad, disponibilidad e integridad de los datos. Primero, protege contra la pérdida de información en caso de fallos de hardware, errores humanos, ataques de ransomware o desastres naturales, asegurando que los datos esenciales puedan ser restaurados rápidamente. Además, proporciona redundancia y alta disponibilidad, permitiendo recuperar datos desde el servidor de backup remoto si la infraestructura principal falla, minimizando así el tiempo de inactividad. También ayuda a cumplir con normativas y regulaciones sobre la protección y retención de datos, garantizando que los datos estén almacenados de manera segura y puedan ser recuperados cuando sea necesario.

Desde una perspectiva económica, utilizar un servidor remoto puede ser más rentable a largo plazo, ya que te permite escalar el almacenamiento según sea necesario sin tener que invertir constantemente en hardware local. Finalmente, los servidores de backup remoto suelen estar optimizados para recuperaciones rápidas y eficientes, reduciendo el impacto en las operaciones comerciales.

21.1 Funcionamiento de Remote Backups Server

Este es el esquema (extraído del esquema de red) para observar con mayor claridad las conexiones realizadas para las copias de seguridad en remoto.

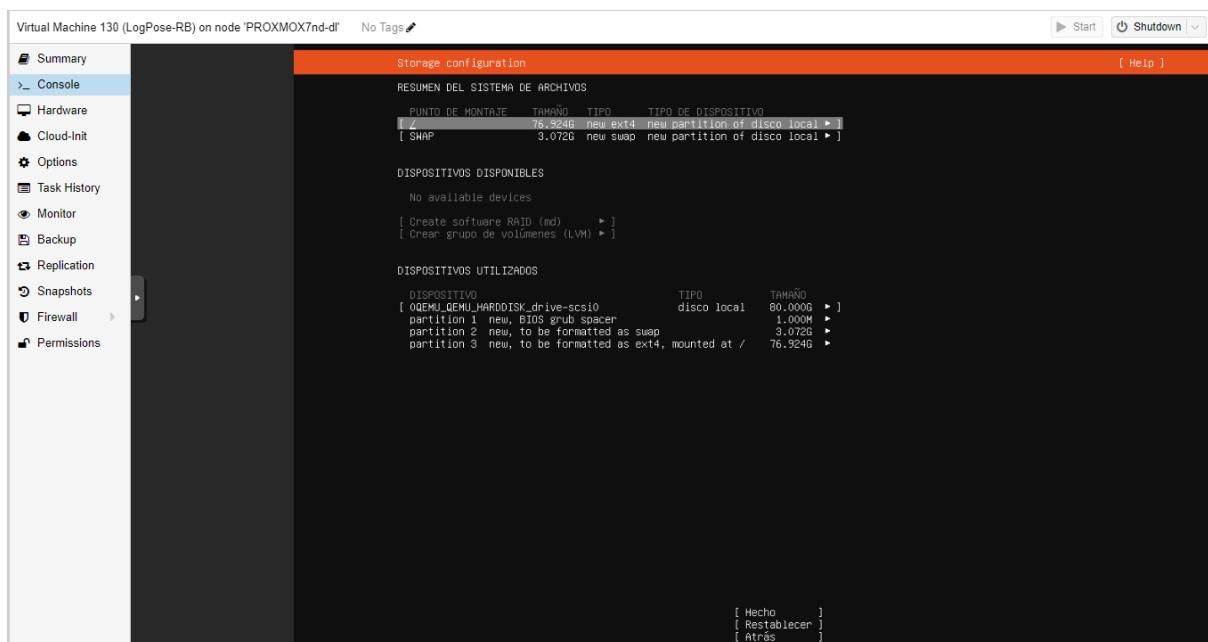


En el esquema se pueden apreciar momentos muy concretos durante la semana que se estarán realizando los “jobs”. Si se fija bien están todos perfectamente configurados para que no se solapen unos con otros. De esta manera, se obtiene un sistema de Copias de Seguridad en red que genera un tráfico mínimo. Además, se irá apreciando a lo largo del despliegue del servidor de Backups Remotos que nos son copias completas de los servidores de la infraestructura, si no que se han elegido cuidadosamente las rutas de ficheros más importantes de cada servidor (Configuraciones, logs y datos varios) para una mayor optimización a nivel de transferencia de archivos por la red.

Todo este proceso también implica una correcta configuración de permisos mínimamente configurados mediante acls, por lo que si alguien comprometiese mi servidor de backups remotos no accederán completamente a toda la información, añadiendo una barrera más de seguridad.

22. Desplegando Remote Backups Server

22.1 Configuración de disco



He decidido crear una partición de Swap por la misma razón que en los demás servidores, para que no se quede congelada la máquina en caso de memoria RAM insuficiente. Dejando libres 77 GB que se utilizarán, formateados en ext4 desde la raíz, para almacenar las copias de seguridad.

Se puede apreciar que este servidor es el que más almacenamiento tiene ya que ahora mismo no, pero a largo plazo será necesario mucho más almacenamiento dependiendo del tamaño final de la infraestructura de la organización donde se vaya a instalar esta infraestructura. En cuyo caso se realizará un estudio para planificar la viabilidad de la instalación de NASs para aumentar el tamaño de espacio disponible en la red de una manera sencilla.

22.2 Configuración de red

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens18:
      dhcp4: no
      addresses: [172.16.1.130/24]
      gateway4: 172.16.1.254
      nameservers:
        addresses: [8.8.8.8]
version: 2
```

Mediante el archivo de configuración de red de Netplan se edita con los parámetros mostrados en la imagen. La IP será la “172.16.1.130”, la máscara “/24”, el gateway la interfaz de PROXMOX con dirección “172.16.1.254” y el DNS será Google para tener resolución de nombres de dominio.

Después ejecutar el siguiente comando para aplicar cambios:

```
# netplan try
```

```
root@logposerb:/home/impeladmin# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
      valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
      valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
  link/ether ea:00:29:5e:56:77 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 172.16.1.130/24 brd 172.16.1.255 scope global ens18
      valid_lft forever preferred_lft forever
    inet6 fe80::e800:29ff:fe5e:5677/64 scope link
      valid_lft forever preferred_lft forever
root@logposerb:/home/impeladmin# hostname -f
logposerb
```

Breve comprobación de que efectivamente se ha aplicado la configuración con “ip a”.

```
root@logposerb:/home/impeladmin# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2024-05-24 10:55:10 UTC; 44min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 16061 (sshd)
    Tasks: 1 (limit: 3425)
   Memory: 1.7M
      CPU: 104ms
     CGroup: /system.slice/ssh.service
             └─16061 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

may 24 10:55:10 logposerb systemd[1]: Starting OpenBSD Secure Shell server...
may 24 10:55:10 logposerb sshd[16061]: Server listening on 0.0.0.0 port 22.
may 24 10:55:10 logposerb sshd[16061]: Server listening on :: port 22.
may 24 10:55:10 logposerb systemd[1]: Started OpenBSD Secure Shell server.
may 24 11:37:57 logposerb sshd[16249]: Accepted password for impeladmin from 172.16.1.200 port 58406 ssh2
may 24 11:37:57 logposerb sshd[16249]: pam_unix(sshd:session): session opened for user impeladmin(uid=1000) by (uid=0)
may 24 11:38:00 logposerb sshd[16249]: pam_unix(sshd:session): session closed for user impeladmin
```

Y por último una comprobación que en la instalación del SO se aplicó el servidor SSH.

23. Configurando accesos usuario local rsync

23.1 Generación de clave rsa

```
impeladmin@magellanws:~/.ssh$ ssh-keygen -t rsa -b 4096 -f ~/.ssh/rsync.key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/impeladmin/.ssh/rsync.key
Your public key has been saved in /home/impeladmin/.ssh/rsync.key.pub
The key fingerprint is:
SHA256:6K46GGMy5jE2wNrvEy+htpT3f2c4bFvx+dQG6Z4KH6g impeladmin@magellanws
The key's randomart image is:
+---[RSA 4096]---+
| |
| .
| .. . S   o |
| *B..o. .... ..|
| *=*o.+.  ooo....+|
| .o+.+o. .Booo++|
| .o=o+o.Eo +ooo...|
+---[SHA256]---+
```

Esta clave es importantísima para acceder de manera segura al servidor remoto. Esta clave se crea en el equipo local desde donde se hará la copia de seguridad al servidor de Backups Remoto.

Ejecutar el siguiente comando en todos los servidores desde donde se quiera realizar las copias de seguridad (en mi caso magellanws, magellanww, magellanwi1, magellanwi2, magellanwdashboard, crimsonldap, kumacamole):

```
`$ ssh-keygen -t rsa -b 4096 -f ~/.ssh/rsync.key`
```

Esto creará nuestras claves encriptadas en rsa, con un bloque de 4096 bits para crear una longitud de clave bastante fuerte, en el directorio home del usuario en la carpeta oculta ssh con el nombre de rsync.key añadiendo la pública con la extensión “.pub” al final del nombre de la privada.

La clave pública es la que se utilizará más adelante para copiarla en el archivo de claves autorizadas del servidor de Backups Remoto para que a la hora de acceder hayan menos privilegios ya que accederán como usuario rsync.

23.2 Copiar clave pública al servidor remoto

```
impeladmin@logposerb:~/.ssh$ sudo su rsync
$ rsync -e ssh -v $HOME/.ssh
$ echo "rsa AAAAB3NzaC1yc2EAAQDHR9R2Rqq0TLT0nk0wvXmut09ZRxUppsVuXyxr3MqmRg0NB04lJcrrjN7qW7GCFx9/Nk4QKAzk1jBgg0L0Sh1k0lfey7YU8u0uWvuxHSECLdyf7j/heCzuqx0tgsS6Ufw1VdC889orC4w8C4PwU809frYxu20Wxabzoevb1fy0t0LJw22+29n7du8xb1Vhf1YSLU19fJ141e2lxXh7nct//ybwVpFD7G9zyFCOPNIcldM92YrVrFf1tp5d4luL592peEsEg+tqjvzNRikh1f7Nsho3LuxSpK7kx07B02SP/mgzt1ebFKan1Upub2sMQ3YSHJUFjjrrCwfbdEl2H29dwb+bxtBa9ns2fc/pgKJX82KxPdJEZndapqZxHZD01gh10KNRH0m/befu)wChw1130n20z52Ykwa0xtsur/Jhr05d5AWFteh1l7UAx18U0z3PXjaYFcuh1krEhwlcMuYBnvCrnpdxZS57hAnfjjbxjKdwbl+0NTPE0pa2obrBatt08yfxpN7x0AMERKEMQ1Pw12e2b6v1172d0Uc1k3qHsay1ldap6dxeAwqT19exghVxtb)0/5002s1xZmq+d73850oc0vdj5TTMD3o9CC7qtw= impeladmin@magellanws" > $HOME/.ssh/authorized_keys
$ chmod 644 $HOME/.ssh/authorized_keys
```

El siguiente paso es, como ya aclaré en el paso anterior, acceder como usuario rsync al servidor remoto donde se almacenarán las copias de seguridad y copiar la clave “.pub” anteriormente generada en authorized_keys.

Hay que tener en cuenta que este directorio también está con los permisos acordemente limitados por cuestiones de seguridad, además del archivo authorized_keys.

NOTA: Este paso también se realiza para los demás servidores que se quieran realizar copias de seguridad (en mi caso copiaría los archivos “.pub” generados en magellanws, magellanww, magellanwi1, magellanwi2, magellanwdashboard, crimsonldap, kumacamole).

23.3 Configurar auto-conexiones al servidor remoto

```
impeladmin@magellanws:~/.ssh$ cat $HOME/.ssh/config
host logposerb
  hostname 172.16.1.130
  user rsync
  IdentityFile ~/.ssh/rsync.key
```

Con este archivo creado en la carpeta oculta “.ssh” se consigue indicar el usuario por defecto que accederá al servidor remoto y con qué llave privada para que la realización del comando rsync más adelante sea exitosa.

```
'host logposerb
  hostname 172.16.1.130
  user rsync
  IdentityFile ~/.ssh/rsync.key'
```

23.4 Configurar acl para usuario impeladmin

```
impeladmin@magellanws:/var$ sudo setfacl -R -m u:impeladmin:rx /var/ossec
impeladmin@magellanws:/var$ sudo find /var/ossec -type f -exec setfacl -d -m u:impeladmin:r {} \;
```

Como el usuario que realizará localmente las conexiones en remoto mediante el comando de rsync será impeladmin se ha realizado una investigación de cuál es la manera más adecuada de acceder a los ficheros sin dar permisos root al usuario desprivilegiado.

Con estos comandos lo que se consigue es primeramente asignar unos permisos recursivos de lectura y escritura sobre la ruta que nos interesa, en este caso el servidor es magellanws (nodo master de Wazuh) y lo que me interesa es la ruta de configuraciones y logs que se generen “/var/ossec”. Y después encontrar todos los ficheros y asignar solo permisos de lectura. Así se quedaría en: Directarios (rx) y Ficheros (r) para el usuario impeladmin.

Rutas seleccionadas para realizar los Backups

magellanws

```
impeladmin@magellanws:/var$ sudo setfacl -R -m u:impeladmin:rx /var/ossec
impeladmin@magellanws:/var$ sudo find /var/ossec -type f -exec setfacl -d -m u:impeladmin:r {} \;
```

magellanww

```
impeladmin@magellanww:~/.ssh$ sudo setfacl -R -m u:impeladmin:rx /var/ossec
impeladmin@magellanww:~/.ssh$ sudo find /var/ossec -type f -exec setfacl -d -m u:impeladmin:r {} \;
```

magellanwi1

```
impeladmin@magellanwi1:~$ sudo setfacl -R -m u:impeladmin:rx /etc/wazuh-indexer
impeladmin@magellanwi1:~$ sudo find /etc/wazuh-indexer -type f -exec setfacl -d -m u:impeladmin:r {} \;
```

magellanwi2

```
impeladmin@magellanwi2:~$ sudo setfacl -R -m u:impeladmin:rx /etc/wazuh-indexer
impeladmin@magellanwi2:~$ sudo find /etc/wazuh-indexer -type f -exec setfacl -d -m u:impeladmin:r {} \;
```

magellanwdashboard

```
impeladmin@magellanwdashboard:~/.ssh$ sudo setfacl -R -m u:impeladmin:rx /etc/wazuh-dashboard
impeladmin@magellanwdashboard:~/.ssh$ sudo find /etc/wazuh-dashboard -type f -exec setfacl -d -m u:impeladmin:r {} \;
```

crimsonldap

```
root@crimsonldap:~# sudo setfacl -R -m u:impeladmin:rx /etc/ldap
root@crimsonldap:~# sudo find /etc/ldap -type f -exec setfacl -d -m u:impeladmin:r {} \;
```

kumacamole

```
impeladmin@kumacamole:~$ sudo setfacl -R -m u:impeladmin:rx /opt/guacamole-docker-compose
impeladmin@kumacamole:~$ sudo find /opt/guacamole-docker-compose -type f -exec setfacl -d -m u:impeladmin:r {} \;
```

23.5 Automatización de Jobs con tareas CRON

Cron es un gestor de tareas en segundo plano que facilita la programación de trabajos para que se ejecuten en el sistema en intervalos regulares o en momentos específicos del día o de la semana. En sistemas operativos Unix, los procesos que deben ejecutarse y su horario se definen en el archivo crontab. Cron se ejecuta cada minuto revisando los archivos en /var/spool/cron o /etc/crontab.

Tareas CRON de cada servidor para realizar los Backups en Remoto

magellanws

```
impeladmin@magellanws:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#
0 1 * * 1,5 rsync -avz --delete /var/ossec rsync@logposerb:/etc/backups/magellanws
```

Se ejecutará el comando rsync para realizar copias de seguridad completas y comprimidas de manera incremental desde “/var/ossec” a la carpeta creada destino “/etc/backups/magellanws” como usuario rsync a la 1:00 am los Lunes y los Viernes.

magellanww

```
impeladmin@magellanww:~/ssh$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#
0 1 * * 1,5 rsync -avz --delete /var/ossec rsync@logposerb:/etc/backups/magellanww
```

Se ejecutará el comando rsync para realizar copias de seguridad completas y comprimidas de manera incremental desde “/var/ossec” a la carpeta creada destino “/etc/backups/magellanww” como usuario rsync a la 1:00 am los Lunes y los Viernes.

magellanwi1

```
impeladmin@magellanwi1:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#
0 0 * * 1,3,5 rsync -avz --delete /etc/wazuh-indexer rsync@logposerb:/etc/backups/magellanwi1
```

Se ejecutará el comando rsync para realizar copias de seguridad completas y comprimidas de manera incremental desde “/etc/wazuh-indexer” a la carpeta creada destino “/etc/backups/magellanwi1” como usuario rsync a las 00:00 am los Lunes, Miércoles y Viernes.

magellanwi2

```
impeladmin@magellanwi2:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#
0 0 * * 1,3,5 rsync -avz --delete /etc/wazuh-indexer rsync@logposerb:/etc/backups/magellanwi2
```

Se ejecutará el comando rsync para realizar copias de seguridad completas y comprimidas de manera incremental desde “/etc/wazuh-indexer” a la carpeta creada destino “/etc/backups/magellanwi2” como usuario rsync a las 00:00 am los Lunes, Miércoles y Viernes.

magellanwdashboard

```
impeadmin@magellanwdashboard:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#
0 0 * * 6 rsync -avz --delete /etc/wazuh-dashboard rsync@logposerb:/etc/backups/magellanwdashboard
```

Se ejecutará el comando rsync para realizar copias de seguridad completas y comprimidas de manera incremental desde “/etc/wazuh-dashboard” a la carpeta creada destino “/etc/backups/magellanwdashboard” como usuario rsync a las 00:00 am los Sábados.

crimsonldap

```
impeadmin@crimsonldap:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#
0 1 * * 2,4,6 rsync -avz --delete /etc/ldap rsync@logposerb:/etc/backups/crimsonldap
```

Se ejecutará el comando rsync para realizar copias de seguridad completas y comprimidas de manera incremental desde “/etc/ldap” a la carpeta creada destino “/etc/backups/crimsonldap” como usuario rsync a la 1:00 am los Martes, Jueves y Sábados.

kumacamole

```
impeladmin@kumacamole:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#
0 1 * * 0 rsync -avz --delete /opt/guacamole-docker-compose rsync@logposerb:/etc/backups/guacamole
```

Se ejecutará el comando rsync para realizar copias de seguridad completas y comprimidas de manera incremental desde “/opt/guacamole-docker-compose” a la carpeta creada destino “/etc/backups/guacamole” como usuario rsync a la 1:00 am los Domingos.

logposerb

```
root@logposerb:/etc/backups# tree -d -L 2
.
└── crimsonldap
    └── ldap
── guacamole
── magellanwdashboard
── magellanwi1
    └── wazuh-indexer
── magellanwi2
    └── wazuh-indexer
── magellanws
    └── ossec
── magellanww
    └── ossec
```

Esta es la estructura de carpetas del segundo nivel que quedaría en el servidor de Backups remoto.

NOTA: magellanwdashboard y guacamole todavía no han realizado copias de seguridad porque esta captura ha sido tomada 5 días después del primer domingo donde configuré los Jobs en CRON.

24. Encriptar backups

24.1 Solución de seguridad

Uno de los problemas existentes en mi configuración backups remotos, es que por mucho que viajen cifradas las copias de seguridad cuando se depositan en el servidor remoto se quedarán en texto plano. De esta manera si un atacante me compromete el servidor de Backups Remotos podrá ver algunos de los archivos más importantes que tiene mi infraestructura, como archivos de configuración, logs, registros de usuarios con sus contraseñas, entre otros. Es por eso que como solución de seguridad se ha propuesto comprimir y encriptar los datos de la ruta “/etc/backups” del servidor remoto con GPG, que es una herramienta de cifrado y firmas digitales que es software libre licenciado bajo la GPL. De esta manera, se necesita una contraseña para descifrar las copias de seguridad y poder leerlas.

24.2 Script de cifrado

```
#!/bin/bash

# Directorio a cifrar
DIR_TO_ENCRYPT="/etc/backups"

# Directorio donde se guardará el archivo cifrado
ENCRYPTED_DIR="/etc/backups"

# Nombre del archivo comprimido y cifrado
TIMESTAMP=$(date +"%Y%m%d%H%M%S")
TAR_FILE="$DIR_TO_ENCRYPT/backups_$TIMESTAMP.tar"
ENCRYPTED_FILE="$ENCRYPTED_DIR/backups_$TIMESTAMP.tar.gpg"

# Contraseña para cifrado
PASSPHRASE="_Gear5th##1420"

# Comprimir el directorio
echo "Comprimiendo el directorio $DIR_TO_ENCRYPT..."
tar -cvf $TAR_FILE -C $DIR_TO_ENCRYPT .

# Verificar si el archivo tar se creó correctamente
if [ $? -ne 0 ]; then
    echo "Error al comprimir el directorio $DIR_TO_ENCRYPT."
    exit 1
fi

# Cifrar el archivo comprimido
echo "Cifrando el archivo $TAR_FILE..."
echo $PASSPHRASE | gpg --batch --yes --passphrase-fd 0 -c $TAR_FILE

# Verificar si el archivo se cifró correctamente
if [ $? -ne 0 ]; then
    echo "Error al cifrar el archivo $TAR_FILE."
    exit 1
fi

# Mover el archivo cifrado al directorio seguro
mv $TAR_FILE.gpg $ENCRYPTED_FILE

# Eliminar el archivo comprimido no cifrado
echo "Eliminando el archivo comprimido no cifrado $TAR_FILE..."
rm $TAR_FILE

# Eliminar el contenido del directorio original sin eliminar el directorio en sí
echo "Eliminando el contenido del directorio original $DIR_TO_ENCRYPT..."
rm -rf $DIR_TO_ENCRYPT/**/

echo "El archivo cifrado se encuentra en: $ENCRYPTED_FILE"
```

El comando “**gpg --batch --yes --passphrase-fd 0 -c \$TAR_FILE**” se utiliza para cifrar un archivo de manera automatizada con GnuPG. Funciona en modo batch, sin interacción con el usuario, y asume respuestas afirmativas a todas las preguntas.

La opción “**--passphrase-fd 0**” permite que la frase de contraseña se lea desde la entrada estándar (stdin), asegurando que la contraseña se pueda proporcionar de manera segura. El archivo que se va a cifrar se especifica mediante la variable \$TAR_FILE.

24.3 Script de descifrado

```
#!/bin/bash

# Directorio de backups
BACKUP_DIR="/etc/backups"

# Contraseña para descifrado
PASSPHRASE=' _Gear5th_ ##1420'

# Buscar el archivo cifrado más reciente
ENCRYPTED_FILE=$(ls -t $BACKUP_DIR/backups_*.tar.gpg | head -n 1)

# Verificar si se encontró un archivo
if [ -z "$ENCRYPTED_FILE" ]; then
    echo "No se encontró ningún archivo de backup cifrado en $BACKUP_DIR."
    exit 1
fi

# Archivo descriptado
DECRYPTED_FILE="${ENCRYPTED_FILE%.gpg}"

# Descriptar el archivo
echo "Descriptando el archivo $ENCRYPTED_FILE..."
echo "$PASSPHRASE" | gpg --batch --yes --passphrase-fd 0 -o "$DECRYPTED_FILE" -d "$ENCRYPTED_FILE"

# Verificar si el archivo se descriptó correctamente
if [ $? -ne 0 ]; then
    echo "Error al descriptar el archivo $ENCRYPTED_FILE."
    exit 1
fi

# Extraer el archivo tar
echo "Extrayendo el archivo $DECRYPTED_FILE..."
tar -xvf "$DECRYPTED_FILE" -C $BACKUP_DIR

# Verificar si el archivo se extrajo correctamente
if [ $? -ne 0 ]; then
    echo "Error al extraer el archivo $DECRYPTED_FILE."
    exit 1
fi

# Eliminar el archivo tar descriptado
echo "Eliminando el archivo tar descriptado $DECRYPTED_FILE..."
rm "$DECRYPTED_FILE"

echo "El archivo ha sido descriptado y extraído en $BACKUP_DIR"
```

Este script, a diferencia del anterior, solamente se ejecutará de manera manual cuando se requiera acceder a una copia de seguridad para consultar o restaurar datos, y esta acción será permitida única y exclusivamente por el usuario root de la máquina de Backups Remotos.

La principal diferencia que tiene este script es que en la línea donde se ejecutará la orden gpg en vez de cifrar renombrará el archivo que se define en la variable “\$DECRYPTED_FILE”, es decir, el nombre del archivo quitando la extensión “.gpg” y descifrará con el parámetro “-d” el archivo definido en la variable “\$ENCRYPTED_FILE” con la misma contraseña con la que se cifró.

24.4 Automatización de encriptado

```
root@logposerb:/etc/backups# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
#30 1 * * 0,1,2,3,4,5,6 /usr/local/bin/cifrar_backups.sh
```

Para automatizar el cifrado de las copias de seguridad se ha optado por ejecutar el script de cifrado de backups, alojado en la ruta “**/usr/local/bin/cifrar_backups.sh**”, durante toda la semana (0 - 6) a la 1:30 am. De esta manera, se proporciona tiempo suficiente para ejecutar las copias de seguridad diarias y así nos aseguramos de que acabará a tiempo y acto seguido se realizará la copia de seguridad y se borrarán los datos en texto plano automáticamente.

25. Conclusiones

Después de el despliegue en su totalidad y la documentación que he llevado a cabo durante todos los días de este proyecto, investigando una nueva tecnología, como es Wazuh, securizando a sí mismo y a los demás servidores, en cuanto a recolección de logs, realizando copias de seguridad de rutas de ficheros muy importantes del sistema optimizando la velocidad de transferencia por la red cada dia o cada semana, el despliegue de un servidor de dominio para centralizar la futura creación de usuarios en el sistema y el despliegue de un servicio web en docker (“aislado”) para centralizar conexiones en remoto y de esta manera no dar acceso directo al hipervisor de Proxmox, interponiendo además un proxy con Nginx entre las máquinas administradas destino y el servidor guacamole, me he dado cuenta que ni aun así estoy cerca de alcanzar la perfección en lo que a seguridad respecta.

Hace falta mucho tiempo, experiencia e investigación para poder actualizar y así mejorar la seguridad y la administración que conlleva está infraestructura. A partir de este punto es muy recomendable, dependiendo de la economía de la empresa destino, la contratación de especialistas en los campos explicados a lo largo de este proyecto, para tener un equipo de administradores de sistemas que se repartirán la tarea para monitorizar, administrar y actualizar la red al completo.

Líneas futuras

Dicho esto me gustaría añadir ciertas funciones que quisiera implementar en la infraestructura en un futuro próximo a modo de mejora:

- La implementación de un Servidor PRTG para monitorizar la red: [Descarga gratuita de PRTG Network Monitor \(paessler.com\)](http://www.paessler.com)
- Integración de nuevas reglas en Wazuh, como puede ser la monitorización de integridad de ficheros importantes del sistema. Ejemplo: las claves ssh, si se añadiese la clave pública del atacante en nuestras claves autorizadas podría conectarse sin necesidad de averiguar la contraseña y no nos enteraríamos.
- Añadir NAS para que en caso de necesitar más almacenamiento para los usuarios nos sea más sencillo controlar el acceso (Hay NAS inteligentes).
- Añadir un servidor DHCP para asignar y reservar IPs de servidores más fácilmente y así cuando la empresa crezca en plantilla poder administrar usuarios de manera sencilla. Además de crear VLANs con dispositivos CISCO.
- Integrar un Firewall virtual de mayor nivel para no depender del propio de la empresa.

26. Bibliografía

1. <https://documentation.wazuh.com/current/index.html>
2. <https://guacamole.apache.org/doc/gug/guacamole-docker.html>
3. Javier Fernández Díaz-Guerra, “En línea sin interrupciones: propuesta de un sistema de alta disponibilidad para garantizar la conectividad y la continuidad del servicio”, Curso 2022-2023.
4. <https://github.com/boschkundendienst/guacamole-docker-compose>
5. Libro de Administración de sistemas operativos de Raul Lerma y Valentín Lacuesta para la creación de un dominio LDAP.
6. <https://chatgpt.com>

27. Anexos

Reglas de IMPEL-DOWN customizadas

Logins fallidos (PROXMOX y NO PROXMOX)

```
<!-- Failed Logins-->
<!-- Despues de 3 logins fallidos en menos de 3 minutos alerta (PROXMOX) -->
<rule id="900001" level="12" frequency="3" timeframe="180">
  <if_matched sid="87201"/><if_matched sid="5503">
    <description>Proxmox VE: authentication failed. 3 failed logins in less than 3 minutes</description>
    <mitre>
      <id>T1110</id>
    </mitre>
  </if_matched>
</rule>

<!-- su: Despues de 3 logins fallidos en menos de 3 minutos alerta cualquier agente (NO PROXMOX) -->
<rule id="900002" level="12" frequency="3" timeframe="180">
  <if_matched sid="5503"/><if_matched sid="87201">
    <match>authentication failure; logname=</match>
    <description>PAM: User login failed 3 times.</description>
    <mitre>
      <id>T1110.001</id>
    </mitre>
  </if_matched>
</rule>
<group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,ppg13_7.8,gdpr_IV_35.7.d,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
</rule>

<!-- ssh: Despues de varios logins fallidos alerta cualquier agente (NO PROXMOX) -->
<rule id="900003" level="12">
  <match>more authentication failures;|REPEATED login failures</match>
  <description>syslog: User missed the password more than one time</description>
  <mitre>
    <id>T1110</id>
  </mitre>
</rule>
<group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,ppg13_7.8,gdpr_IV_35.7.d,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
</rule>
```

Logins exitosos fuera de horario laboral (PROXMOX)

```
<!-- Successful Logins (PROXMOX) -->
<!-- Lunes a viernes de 00:00 am - 11:30 am 1 autenticacion exitosa alerta (PROXMOX) -->
<rule id="900005" level="12">
  <if_sid>87200</if_sid>
  <weekday>monday,tuesday,wednesday,thursday,friday</weekday>
  <time>00:00 am - 11:30 am</time>
  <match> successful auth for user </match>
  <description>Proxmox VE authentication succeeded.</description>
  <mitre>
    <id>T1078</id>
  </mitre>
</rule>
<group>authentication_success,pci_dss_10.2.5,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>

<!-- Lunes a viernes de 7:00 pm - 00:00 am 1 autenticacion exitosa alerta (PROXMOX) -->
<rule id="900006" level="12">
  <if_sid>87200</if_sid>
  <weekday>monday,tuesday,wednesday,thursday,friday</weekday>
  <time> 7:00 pm - 00:00 am</time>
  <match> successful auth for user </match>
  <description>Proxmox VE authentication succeeded.</description>
  <mitre>
    <id>T1078</id>
  </mitre>
</rule>
<group>authentication_success,pci_dss_10.2.5,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
</rule>

<!-- Fines de semana 1 autenticacion exitosa alerta (PROXMOX) -->
<rule id="900007" level="12">
  <if_sid>87200</if_sid>
  <weekday>weekends</weekday>
  <match> successful auth for user </match>
  <description>Proxmox VE authentication succeeded.</description>
  <mitre>
    <id>T1078</id>
  </mitre>
</rule>
<group>authentication_success,pci_dss_10.2.5,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
</rule>
```

Logins exitosos fuera de horario laboral (NO PROXMOX)

```
<!-- Successful Logins (NO PROXMOX) -->
<!-- Lunes a viernes de 00:00 am - 11:30 am 1 autenticacion exitosa alerta (NO PROXMOX) -->
<rule id="900008" level="12">
<if sid=5500</if_sid>
<weekday>monday,tuesday,wednesday,thursday,friday</weekday>
<time>00:00 am - 11:30 am</time>
<regex type="pcre2" negate="yes">cron:session</regex>
<match>session opened for user </match>
<description>PAM: Login session opened.</description>
<mitre>
<id>T1078</id>
</mitre>
<group>authentication_success,pci_dss_10.2.5,pgp13_7.8,pgp13_7.9,gdpr_IV_32.2,hipa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
</rule>

<!-- Lunes a viernes de 5:00 pm - 00:00 am 1 autenticacion exitosa alerta (NO PROXMOX) -->
<rule id="900009" level="12">
<if sid=5500</if_sid>
<weekday>monday,tuesday,wednesday,thursday,friday</weekday>
<time>7:00 pm - 00:00 am</time>
<regex type="pcre2" negate="yes">cron:session</regex>
<match>session opened for user </match>
<description>PAM: Login session opened.</description>
<mitre>
<id>T1078</id>
</mitre>
<group>authentication_success,pci_dss_10.2.5,pgp13_7.8,pgp13_7.9,gdpr_IV_32.2,hipa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
</rule>

<!-- Fines de semana 1 autenticacion exitosa alerta (NO PROXMOX) -->
<rule id="900010" level="12">
<if sid=5500</if_sid>
<weekday>weekends</weekday>
<regex type="pcre2" negate="yes">cron:session</regex>
<match>session opened for user </match>
<description>PAM: Login session opened.</description>
<mitre>
<id>T1078</id>
</mitre>
<group>authentication_success,pci_dss_10.2.5,pgp13_7.8,pgp13_7.9,gdpr_IV_32.2,hipa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
</rule>
```

Sudo to root exitosos

```
<!-- Successful sudo to root -->
<rule id="900011" level="12">
<if sid=5400</if_sid>
<regex> ; USER=root ; COMMAND=| ; USER=root ; TSID=\$+ ; COMMAND=</regex>
<description>Successful sudo to ROOT executed.</description>
<mitre>
<id>T1548.003</id>
</mitre>
<group>pci_dss_10.2.5,pci_dss_10.2.2,pgp13_7.6,pgp13_7.8,pgp13_7.13,gdpr_IV_32.2,hipa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,nist_800_53_AC.6,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
</rule>
```