Universitat Oberta de Catalunya

uoc.edu

Plataforma para la detección y control de puntos finales en la red usando tecnologías Zero Trust y NAC

Linck Tello Flores
Máster Universitario en Ciberseguridad y Privacidad
Trabajo Final de Máster. Seguridad Empresarial

Víctor Méndez Muñoz
Victor García Font

24/Enero/2024

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Plataforma para la detección y control de puntos finales en la red usando tecnologías Zero Trust y NAC*
Nombre del autor:	Linck Tello Flores
Nombre del consultor:	Víctor Méndez Muñoz
Nombre del PRA:	Victor García Font
Fecha de entrega (mm/aaaa):	02/2024
Titulación o programa:	Máster Universitario en Ciberseguridad y Privacidad
Área del Trabajo Final:	Seguridad Empresarial
Idioma del trabajo:	Castellano
Palabras clave	Respuesta a incidentes, Zero Trust, Ciberseguridad

Resumen del Trabajo

El control de los dispositivos conectados a la red es una tarea que se realiza de forma rutinaria y generalmente es complicada de gestionar debido a lo cual se pierde el control ya sea por las continuas altas y bajas de dispositivos y/o por el uso de dispositivos personales en las organizaciones. Esto hace que cuando ocurre un incidente de seguridad, la labor de rastrear el origen se dificulta perdiendo tiempo para responder a una amenaza. Por ende, ante el aumento de las incidencias relacionadas con la falta de visibilidad de los dispositivos conectados a la red, se propone el desarrollado una plataforma de ciberseguridad para la detección y control de activos basada en sensores de red para la identificación, evaluación y respuesta autónoma que garantice la seguridad en la red usando la arquitectura de confianza cero.

La plataforma se implementa mediante la instalación de sensores locales que envían datos a un grupo de servidores en la nube para su almacenamiento, monitoreo y análisis utilizando software de código abierto como OpenSearch, Grafana, Zeek y ClickHouse. Asimismo, el departamento de informática cuenta con una consola de administración desde donde gestionará los dispositivos, podrá conocer su estado y realizar acciones de control mediante la aplicación de políticas de seguridad predefinidas por especialistas en ciberseguridad para evitar posibles compromisos de seguridad para lo cual la plataforma se integra vía API a fabricantes de antimalware para evaluar si el activo está protegido y en caso contrario, generar acciones de respuesta automáticas.

Abstract

The device control connected to the network is a task that is carried out routinely and is generally complicated to manage, due to which control is lost either due to the continuous input and output of devices and/or due to the use of personal devices in organizations.This means that when a security incident occurs, the task of tracing the origin is difficult, losing time to respond to a threat. Therefore, given the increase in incidents related to the lack of visibility of devices connected to the network, it is proposed to develop a cybersecurity platform for the detection and control of assets based on network sensors for identification, evaluation and autonomous response that guarantees network security, using zero trust architecture.

The platform is implemented by installing local sensors that send data to a group of cloud servers for storage, monitoring and analysis using open source software such as OpenSearch, Grafana, Zeek and ClickHouse. Likewise, the

IT department has an administration console from which it will manage the devices, be able to know their status and carry out control actions by applying predefined security policies by cybersecurity specialists to avoid possible security compromises for which the platform integrates via API with antimalware manufacturers to evaluate whether the asset is protected and, if not, generate automatic response actions.

Índice

Lista de figuras

1. Introducción

1.1. Contexto y justificación del trabajo

El desarrollo de la plataforma propuesta se basa en una serie de desafíos y necesidades críticas que enfrentan las empresas en la actualidad en relación con la protección de la información, el cual es el recurso más valioso de la organización. Para lograr esto se requiere contar con mecanismos que permitan una adecuada gestión de las amenazas de seguridad partiendo de la detección y control de los dispositivos conectados a la red que es un elemento clave de una arquitectura de confianza cero[1] o Zero Trust Architecture descrita por el Instituto Nacional de Estándares y Tecnología en la publicación Especial 800-207.

Así mismo, según el Informe sobre el coste de una vulneración de datos de 2023 realizada por el Ponemon Institute y patrocinada, analizada y publicada por IBM® Security “el coste total promedio de una vulneración de datos alcanzó un máximo histórico de 4.45 millones de dólares en 2023 y que representa un aumento del 2.3% con respecto al año 2022”. En el mismo informe se identifica que “La IA y la automatización de la seguridad demostraron ser inversiones importantes para reducir costes y minimizar el tiempo necesario para identificar y contener las vulneraciones de datos”.

Como podemos ver en los informes descritos, estos desafíos y necesidades hacen que la creación de la plataforma de ciberseguridad propuesta sea relevante ya que las empresas se encuentran con la complejidad en la gestión de dispositivos conectados a la red, incluyendo dispositivos de uso personal o BYOD lo que ha aumentado la complejidad de su gestión y por tanto la pérdida de control sobre los equipos de punto final que se conectan a la red.

Los departamentos de informática o tecnologías de la información buscan conocer y controlar los dispositivos conectados a la red por lo que implementan plataformas para la gestión de activos como pueden ser algunos comerciales

[1] Arquitectura de Confianza Cero o Zero Trust Architecture.

como ManageEngine [2], SolarWinds[3] y otros de origen open-source como OCS Inventory[4] y GLPI[5]; sin embargo, estas herramientas son productos aislados de la seguridad o la ciberseguridad en la organización y que además requieren agentes que deben instalarse en todos los dispositivos o dependen de realizar escaneos continuos a la red, por cuando no permiten tener información actualizada y en línea cuando se trata de implementar una política de seguridad en la organización, es decir, hablamos de contar con información en tiempo real para la toma de decisiones mediante la creación de políticas de seguridad y/o control de acceso a la red.

Así mismo, para asegurar la protección de los puntos finales los departamentos de informática o tecnologías de la información adquieren soluciones de protección antivirus o antimalware para evitar que las amenazas que están en constante evolución y que se vuelven cada vez más sofisticadas impacten en la red y en la información. Sin embargo, se encuentra nuevamente en el mismo problema que generan las plataformas de gestión de activos y cuando ocurre un incidente de ciberseguridad, sea un ataque de malware como el ransomware o una intrusión se vuelve compleja la identificación del origen del problema.

La falta de visibilidad sobre los dispositivos conectados a la red empresarial dificultará no solamente la adopción de una arquitectura de confianza cero que ayudará a la organización a mejorar los niveles de seguridad, sino también dificulta principalmente la detección de incidentes de seguridad, lo que puede resultar en la pérdida de tiempo valioso para rastrear y responder a amenazas, lo que aumenta el riesgo de daños significativos a la organización.

Por esto mismo que la capacidad para generar respuestas autónomas o automáticas ante las amenazas descritas es crucial para minimizar el tiempo de exposición a riesgos en la organización y para ello el departamento de informática requiere de una plataforma que pueda tomar decisiones basadas en políticas de seguridad predefinidas por especialistas en ciberseguridad

[2] https://www.manageengine.com/products/asset-explorer/
[3] https://www.solarwinds.com/web-help-desk/use-cases/it-asset-management
[4] https://ocsinventory-ng.org/
[5] https://glpi-project.org/es/

(playbooks) las mismas que actuarán en forma autónoma primero realizando la evaluación del dispositivo y en caso de no estar conforme conectarse vía API con los cortafuegos y conmutadores de red para establecer acciones de respuesta y contención a posibles incidentes de seguridad.

La plataforma propuesta busca optimizar los recursos del departamento de informática al desarrollar sensores que se encarguen de realizar el monitoreo continuo de la red y de realizar la identificación de los dispositivos que se conectan a ella mediante el análisis del tráfico de red y la detección de las conexiones IP, permitiendo la identificación y administración eficiente de los dispositivos conectados, sin necesidad de desplegar ningún agente para comprobar si este cumple con contar con la protección antimalware que ha adquirido la organización usando para ello una conexión vía API a los distintos fabricantes de seguridad lo que a su vez permitirá maximizar la productividad y reducir costos en la gestión de la seguridad y la gestión de los activos.

Otro punto que debemos considerar es que, en muchos sectores, existen regulaciones estrictas de ciberseguridad que las empresas deben cumplir, por lo que la plataforma puede ayudar a garantizar el cumplimiento de estas normativas al proporcionar una herramienta de identificación, evaluación y respuesta autónoma todos los días del año los siete días a la semana.

1.2. Objetivos del trabajo

Los principales objetivos de este trabajo de fin de máster son los siguiente:

A Nivel Investigación:

- Investigar como usar las herramientas de código abierto OpenSearch[6], Zeek[7], Grafana[8] y ClickHouse[9] para implementar una pila de almacenamiento y análisis de datos.
- Investigar sobre las capacidades que ofrecen los sensores para la detección de activos en la red.
- Investigar y estudiar el uso de APIs para el acceso al backend de plataformas de seguridad como los antimalware, cortafuegos y conmutadores de red.

A Nivel de Implementación:

- Aprender a instalar y configurar OpenSearch, Zeek, Grafana y ClickHouse.
- Aprender a configurar programas para el almacenamiento y monitoreo de logs.
- Aprender a configurar sensores de red en modo promiscuo o en equipos de comunicación en modo Port-Mirror o SPAN.
- Desarrollar un nuevo producto comunitario y comercial basado en la plataforma a desarrollar qué permita atender los requerimientos de diferentes tipos de empresas frente a los problemas detectados.

El trabajo por desarrollar tiene características de investigación y búsqueda de información sobre las herramientas de código abierto a utilizar en el proyecto, pero además pretende finalizar con la implementación de una plataforma funcional que cumpla con los siguientes objetivos:

[6] https://opensearch.org/docs/latest/
[7] https://docs.zeek.org/en/master/about.html
[8] https://grafana.org/
[9] https://clickhouse.com/docs/en/install

Identificación: Mediante el análisis del tráfico de red realizado por los sensores se identifican los dispositivos que se conectan a la red, está información descubierta por los sensores es enviada como registros a los servidores de almacenamiento de logs para el monitoreo y análisis de la información.

Evaluación: Mediante la integración con distintas plataformas de antimalware que use la organización usando para ello APIS se elaborarán resultados de análisis del mismo con el fin de tomar decisiones basados en las políticas de seguridad predefinidas por el los especialistas de ciberseguridad que incluirá la plataforma.

Respuesta: Toma de decisiones autónomas las 24 horas, los 7 días de la semana con el fin de evitar posibles incidentes de seguridad generados por activos que incumplen las políticas de seguridad, para ello un agente de respuesta instalado en los sensores se comunicarán vía API a los equipos de seguridad perimetrales como los cortafuegos para incluir al dispositivo detectado en una política restrictiva o en el caso de un conmutador bloquear el puerto de red, poner en cuarentena, entre otras acciones.

Interfaz visual simple: Mediante una interfaz simple el departamento de informática tendrá una plataforma donde podrá visualizar en tiempo real los dispositivos conectados a la red y a su vez podrá verificar cuales de ellos no cumplen con las políticas de protección antimalware en la organización.

Mejora de la gestión de la ciberseguridad en la organización: Mediante la identificación de los activos se pretende mejorar la gestión de la ciberseguridad en las organizaciones de una forma simple y eficiente.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

En el mundo empresarial actual, la seguridad de la información y la ciberseguridad se han convertido en un componente crítico para la supervivencia y el éxito de las organizaciones, salvo algunas excepciones que están fuera del alcance del presente proyecto de TFM. El uso de las tecnologías de la información para proteger a las empresas contra intrusiones y pérdida de datos no sólo permite resguardar la confidencialidad y la integridad de la información, sino que también tiene un impacto significativo en la sostenibilidad empresarial. Este TFM aborda este problema mediante la creación de un prototipo de producto que contribuirá a la sostenibilidad, abordando aspectos como la eficiencia operativa, la responsabilidad social y la gestión de recursos.

La implementación de sistemas de seguridad avanzados ayuda a reducir el riesgo de interrupciones en las operaciones, minimizando el tiempo de inactividad, bajando los riesgos y de esta forma optimizando la continuidad del negocio. Estos aspectos no sólo tienen un impacto positivo en la sostenibilidad operativa, sino que también contribuyen a la eficiencia en el uso de recursos, al disminuir el desperdicio de tiempo y energía asociado con la recuperación de incidentes de seguridad, más aún en tiempos donde el uso de la energía en distintas partes del mundo es un tema de agenda de empresa y gobiernos.

Así mismo, la adopción de medidas proactivas para proteger la información de la empresa también refleja una responsabilidad social y ética empresarial. La seguridad de la información no solo se trata de proteger los activos de la empresa, sino también de salvaguardar la información confidencial de los clientes y socios de negocios. Al utilizar las tecnologías de la información para implementar políticas de seguridad robustas, las empresas demuestran su compromiso con la protección de la privacidad y la confidencialidad, contribuyendo así a su reputación y credibilidad. Este enfoque ético no sólo fortalece la posición de la empresa en el mercado, sino que también tiene un impacto positivo en la sostenibilidad a largo plazo al construir relaciones de confianza y fomentar una cultura de seguridad. Estos aspectos son aún más

relevantes en las pequeñas y medianas empresas donde los recursos económicos son escasos para la adopción de tecnologías de vanguardia.

En cuanto a la diversidad de género en el ámbito de la seguridad informática no solo es una cuestión de equidad, sino también una estrategia efectiva para fortalecer la resiliencia de las empresas. El presente TFM no limita ni excluye la participación de nadie, es decir, los diversos géneros sociales pueden tener un rol activo en el desarrollo de mejoras al diseño y desarrollo del proyecto, lo que puede llevar incluso a añadir mejoras y/o soluciones más creativas y completas. De este modo, al contribuir a brindar igualdad de oportunidades, mejoramos la capacidad para enfrentar amenazas cibernéticas de manera más efectiva.

1.4. Enfoque y método seguido

De acuerdo con los objetivos señalados el proyecto se dividirá en tres partes:

La primera parte consiste en la investigación y elección de las herramientas de código abierto a utilizar en el proyecto y que se presentarán en el siguiente entregable.

La segunda parte consiste en la implementación del clúster de servidores para el almacenamiento y monitoreo de datos, así como el desarrollo del sensor que se instalará en la red del cliente final el cual se irá presentando en los siguientes entregables y cuya estrategia a seguir consiste en la adaptación de módulos o programas de las aplicaciones de código fuente para adaptarlas al proyecto. Es decir, usaremos piezas de software y no se realizarán implementaciones completas de los programas descritos ya que estas tienen diferentes funcionalidades y casos de uso algunas de las cuales están fuera del alcance del presente proyecto.

La tercera parte consiste en elaborar un prototipo de la consola de administración que usarán los departamentos de informática para gestionar y monitorear la seguridad de su organización basada en la identificación permanente de activos conectados a la red. En esta etapa se desarrollará un producto nuevo usando para ello las herramientas de programación Python y Vue tanto para el backend como para el frontend.

Finalmente, se debe tener en cuenta que el proyecto podrá ser modificado en base a los resultados de la elección final de las herramientas a usar en los siguiente entregables para ajustar los objetivos a resultados reales debido al tiempo que se tiene para el desarrollo del proyecto.

1.5. Planificación del trabajo

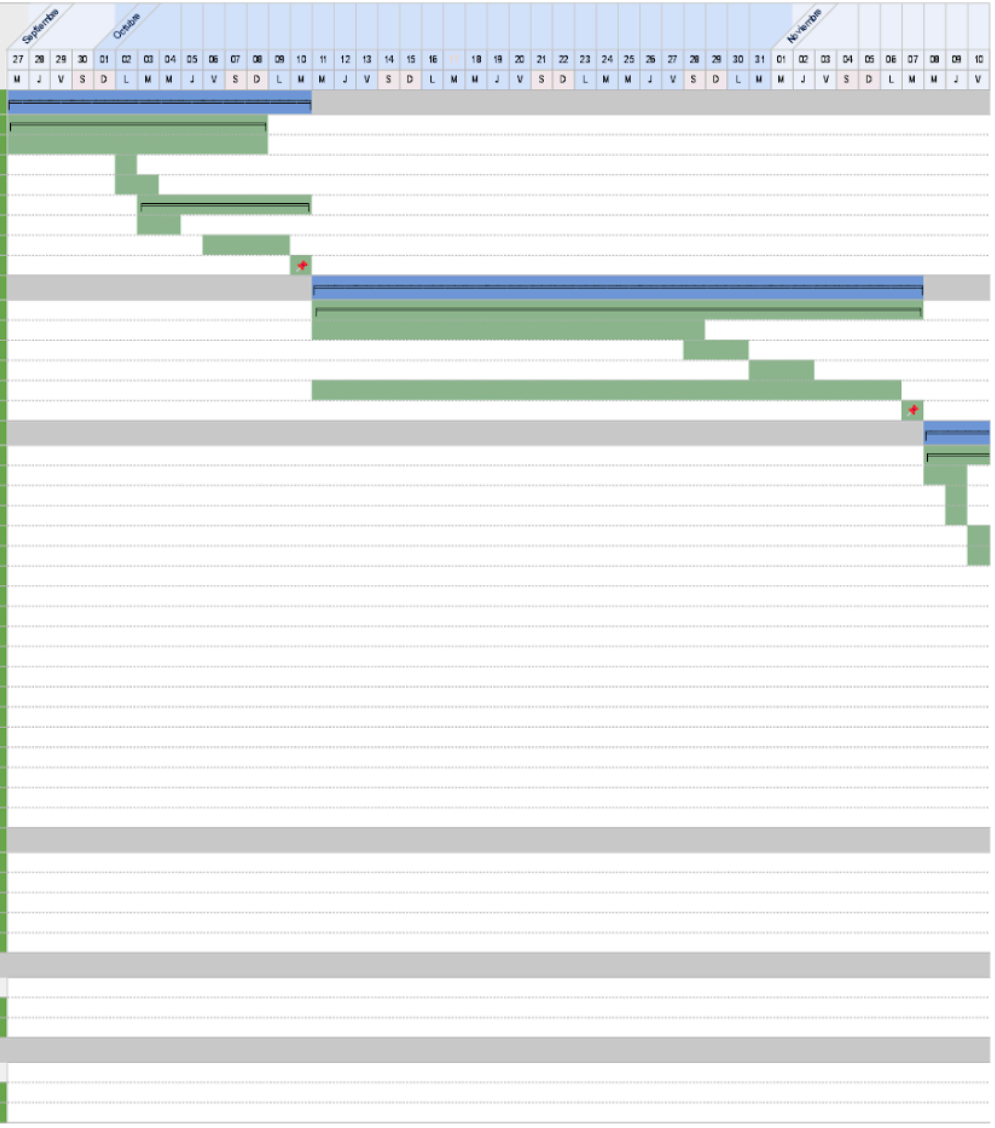
# EDT	TÍTULO DE LA TAREA	FECHA DE INICIO	FECHA DE FIN	DURACIÓN	HITO	% COMPLETADO DE LA TAREA
1	**PEC1. Plan de trabajo**	**27/09/23**	**10/10/23**	**13**		**100 %**
1.1	**Planificación del Proyecto**	27/09/23	08/10/23	11		100 %
1.1.1	└ Definición del nombre del proyecto	27/09/23	08/10/23	12		100 %
1.1.2	└ Definición de la metodología a usar	02/10/23	02/10/23	1		100 %
1.1.2	└ Definición de objetivos	02/10/23	03/10/23	2		100 %
1.2	**Elaboración del Cronograma de trabajo**	03/10/23	10/10/23	7		100 %
1.2.1	└ Definición de tareas	03/10/23	04/10/23	2		100 %
1.2.2	└ Definición de entregables	06/10/23	09/10/23	4		100 %
1.2.3	└ Entrega de la PEC1	10/10/23	10/10/23	1	x	100 %
2	**PEC2. Entrega de Seguimiento - Investigación**	**11/10/23**	**07/11/23**	**26**		**100 %**
2.1	**Investigación**	11/10/23	07/11/23	26		100 %
2.1.1	└ Investigación sobre ZTA y NAC	11/10/23	28/10/23	18		100 %
2.1.2	└ Investigación del uso de APIs	28/10/23	30/10/23	3		100 %
2.1.3	└ Investigación sobre las herramientas a usar	31/10/23	02/11/23	3		100 %
2.1.4	└ Elaborar los documentos PEC2	11/10/23	06/11/23	27		100 %
2.1.5	└ Entrega de la PEC2	07/11/23	07/11/23	1	x	100 %
3	**PEC3. Entrega de Seguimiento - Implementación**	**08/11/23**	**05/12/23**	**27**		**100 %**
3.1	**Preparación del entorno nube**	08/11/23	17/11/23	9		100 %
3.1.1	└ Instalación de software central	08/11/23	09/11/23	2		100 %
3.1.2	└ Instalación de ClickHouse	09/11/23	09/11/23	1		100 %
3.1.3	└ Instalación de OpenSearch	09/11/23	09/11/23	1		100 %
3.1.4	└ Instalación de Grafana	10/11/23	12/11/23	3		100 %
3.1.5	└ Creación de Dashboards	10/11/23	17/11/23	8		100 %
3.2	**Desarrollo del Sensor**	18/11/23	26/11/23	8		100 %
3.2.1	└ Creación de Repositorio RPM	18/11/23	20/11/23	3		100 %
3.2.2	└ Pruebas Iniciales	20/11/23	21/11/23	2		100 %
3.2.3	└ Afinamiento del Sensor	21/11/23	22/11/23	2		100 %
3.2.4	└ Creación de archivos de configuración	22/11/23	22/11/23	1		100 %
3.2.5	└ Creación de manuales de instalación	23/11/23	26/11/23	4		100 %
3.3	**Diseño de la Consola de Cliente Final**	15/11/23	02/12/23	17		100 %
3.3.1	└ Elaboración del diseño de la interface	15/11/23	26/11/23	12		100 %

3.3.2	└ Pruebas iniciales de la versión 1.0 Alpha	26/11/23	28/11/23	3		100 %
3.3.3	└ Elaboración de conectores API	28/11/23	02/12/23	5		100 %
3.4	**Elaboración del entregable**	18/11/23	05/12/23	17		100 %
3.4.1	└ Elaborar los documentos PEC3	18/11/23	05/12/23	18		100 %
3.4.2	└ Entrega de la PEC3	05/12/23	05/12/23	1	x	100 %
4	**PEC4. Elaboración de la memoria final**	**06/12/23**	**09/01/24**	**33**		**100 %**
4.1	**Elaboración de la memoria final**	06/12/23	09/01/24	33		100 %
4.1.1	└ Revisión del documento final	06/12/23	04/01/24	30		100 %
4.1.2	└ Elaborar conclusiones del proyecto	01/01/24	05/01/24	5		100 %
4.1.3	└ Elaborar memoria final	05/01/24	08/01/24	4		100 %
4.1.5	└ Entrega de la memoria final PEC4	09/01/24	09/01/24	1	x	100 %
5	**Presentación en video**	**10/01/24**	**16/01/24**	**6**		**100 %**
5.1	**Elaboración del video**	10/01/24	16/01/24	6		100 %
5.1.1	└ Creación del material para el video	10/01/24	15/01/24	6		100 %
5.1.2	└ Entrega del video de TFM	16/01/24	16/01/24	1	x	100 %
6	**Defensa del TFM**	**22/01/24**	**26/01/24**	**4**		**100 %**
6.1	**Preparación de la presentación PPT**	22/01/24	26/01/24	4		100 %
6.1.1	└ Elaboración de la presentación	22/01/24	25/01/24	4		100 %
6.1.2	└ Defensa Final	26/01/24	26/01/24	1	x	100 %

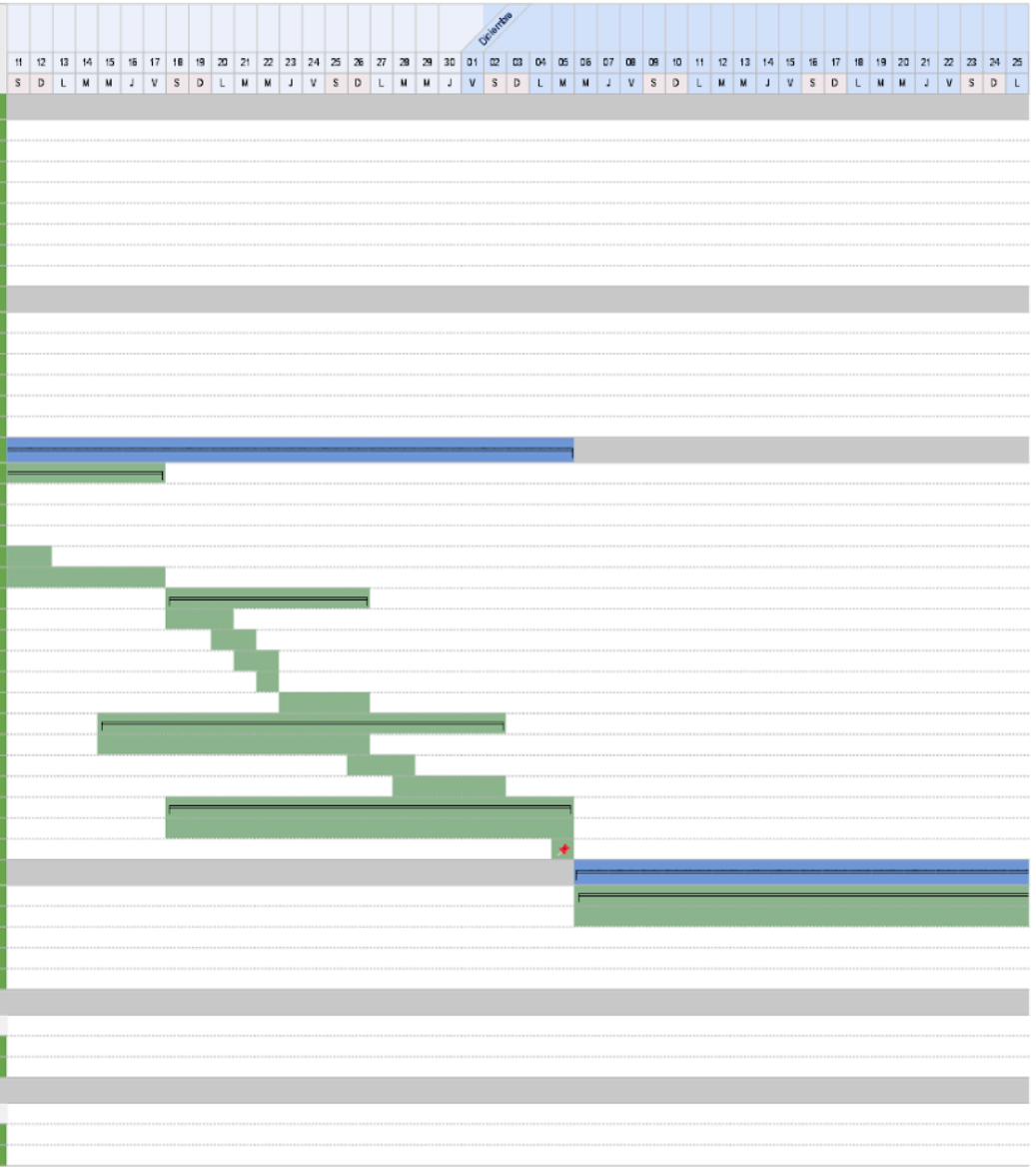
Figura 1: Diagrama de Gantt del trabajo de fin de máster

1.6. Cronograma del trabajo

# EDT	TÍTULO DE LA TAREA	RESPONSABLE DE LA TAREA	FECHA DE INICIO	FECHA DE FIN	DURACIÓN	HITO	% COMPLETADO DE LA TAREA
1	**PEC1. Plan de trabajo**		**27/09/23**	**10/10/23**	**13**		**100 %**
1.1	**Planificación del Proyecto**		**27/09/23**	**08/10/23**	**11**		**100 %**
1.1.1	└Definición del nombre del proyecto	Linck Telio Flores	27/09/23	08/10/23	12	☐	100 %
1.1.2	└Definición de la metodología a usar	Linck Telio Flores	02/10/23	02/10/23	1	☐	100 %
1.1.2	└Definición de objetivos	Linck Telio Flores	02/10/23	03/10/23	2	☐	100 %
1.2	**Elaboración del Cronograma de trabajo**		**03/10/23**	**10/10/23**	**7**		**100 %**
1.2.1	└Definición de tareas	Linck Telio Flores	03/10/23	04/10/23	2	☐	100 %
1.2.2	└Definición de entregables	Linck Telio Flores	06/10/23	09/10/23	4	☐	100 %
1.2.3	└Entrega de la PEC1	Linck Telio Flores	10/10/23	10/10/23	1	☑	100 %
2	**PEC2. Entrega de Seguimiento - Investigación**		**11/10/23**	**07/11/23**	**26**		**100 %**
2.1	**Investigación**		**11/10/23**	**07/11/23**	**26**		**100 %**
2.1.1	└Investigación sobre ZTA yNAC	Linck Telio Flores	11/10/23	28/10/23	18	☐	100 %
2.1.2	└Investigación del uso de APIs	Linck Telio Flores	28/10/23	30/10/23	3	☐	100 %
2.1.3	└Investigación sobre las herramientas a usar	Linck Telio Flores	31/10/23	02/11/23	3	☐	100 %
2.1.4	└Elaborar los documentos PEC2	Linck Telio Flores	11/10/23	06/11/23	27	☐	100 %
2.1.5	└Entrega de la PEC2	Linck Telio Flores	07/11/23	07/11/23	1	☑	100 %
3	**PEC3. Entrega de Seguimiento - Implementación**		**08/11/23**	**05/12/23**	**27**		**100 %**
3.1	**Preparación del entorno nube**		**08/11/23**	**17/11/23**	**9**		**100 %**
3.1.1	└Instalación de Software central	Linck Telio Flores	08/11/23	09/11/23	2	☐	100 %
3.1.2	└Instalación de ClickHouse	Linck Telio Flores	09/11/23	09/11/23	1	☐	100 %
3.1.3	└Instalación de OpenSearch	Linck Telio Flores	09/11/23	09/11/23	1	☐	100 %
3.1.4	└Instalación de Grafana	Linck Telio Flores	10/11/23	12/11/23	3	☐	100 %
3.1.5	└Creación de Dashboards	Linck Telio Flores	10/11/23	17/11/23	8	☐	100 %
3.2	**Desarrollo del Sensor**		**18/11/23**	**26/11/23**	**8**		**100 %**
3.2.1	└Creación de Repositorio RPM	Linck Telio Flores	18/11/23	20/11/23	3	☐	100 %
3.2.2	└Pruebas Iniciales	Linck Telio Flores	20/11/23	21/11/23	2	☐	100 %
3.2.3	└Afinamiento del Sensor	Linck Telio Flores	21/11/23	22/11/23	2	☐	100 %
3.2.4	└Creación de archivos de configuración	Linck Telio Flores	22/11/23	22/11/23	1	☐	100 %
3.2.5	└Creación de manuales de instalación	Linck Telio Flores	23/11/23	26/11/23	4	☐	100 %
3.3	**Diseño de la Consola de Cliente Final**		**15/11/23**	**02/12/23**	**17**		**100 %**
3.3.1	└Elaboración del diseño de la interface	Linck Telio Flores	15/11/23	26/11/23	12	☐	100 %
3.3.2	└Pruebas iniciales del primer Prototipo	Linck Telio Flores	26/11/23	28/11/23	3	☐	100 %
3.3.3	└Elaboración de conectores API	Linck Telio Flores	28/11/23	02/12/23	5	☐	100 %
3.4	**Elaboración del entregable**		**18/11/23**	**05/12/23**	**17**		**100 %**
3.4.1	└Elaborar los documentos PEC3	Linck Telio Flores	18/11/23	05/12/23	18	☐	100 %
3.4.2	└Entrega de la PEC3	Linck Telio Flores	05/12/23	05/12/23	1	☑	100 %
4	**PEC4. Elaboración de la memoria final**		**06/12/23**	**09/01/24**	**33**		**100 %**
4.1	**Elaboración de la memoria final**		**06/12/23**	**09/01/24**	**33**		**100 %**
4.1.1	└Revisión del documento final	Linck Telio Flores	06/12/23	04/01/24	30	☐	100 %
4.1.2	└Elaborar conclusiones del proyecto	Linck Telio Flores	01/01/24	05/01/24	5	☐	100 %
4.1.3	└Elaborar memoria final	Linck Telio Flores	05/01/24	08/01/24	4	☐	100 %
4.1.5	└Entrega de la memoria final PEC4	Linck Telio Flores	09/01/24	09/01/24	1	☑	100 %
5	**Presentación en video**		**10/01/24**	**16/01/24**	**6**		**0 %**
5.1	**Elaboración del video**		**10/01/24**	**16/01/24**	**6**		**0 %**
5.1.1	└Creación del material para el video	Linck Telio Flores	10/01/24	15/01/24	6	☐	100 %
5.1.2	└Entrega del video de TFM	Linck Telio Flores	16/01/24	16/01/24	1	☑	100 %
6	**Defensa del TFM**		**22/01/24**	**24/01/24**	**2**		**0 %**
6.1	**Preparación de la presentación PPT**		**22/01/24**	**24/01/24**	**2**		**0 %**
6.1.1	└Elaboración de la presentación	Linck Telio Flores	22/01/24	24/01/24	3	☐	100 %
6.1.2	└Defensa Final	Linck Telio Flores	24/01/24	24/01/24	1	☑	100 %



# EDT	TÍTULO DE LA TAREA	RESPONSABLE DE LA TAREA	FECHA DE INICIO	FECHA DE FIN	DURACIÓN	HITO	% COMPLETADO DE LA TAREA
1	**PEC1. Plan de trabajo**		**27/09/23**	**10/10/23**	**13**		**100 %**
1.1	**Planificación del Proyecto**		**27/09/23**	**08/10/23**	**11**		**100 %**
1.1.1	└Definición del nombre del proyecto	Linck Telio Flores	27/09/23	08/10/23	12	☐	100 %
1.1.2	└Definición de la metodología a usar	Linck Telio Flores	02/10/23	02/10/23	1	☐	100 %
1.1.2	└Definición de objetivos	Linck Telio Flores	02/10/23	03/10/23	2	☐	100 %
1.2	**Elaboración del Cronograma de trabajo**		**03/10/23**	**10/10/23**	**7**		**100 %**
1.2.1	└Definición de tareas	Linck Telio Flores	03/10/23	04/10/23	2	☐	100 %
1.2.2	└Definición de entregables	Linck Telio Flores	06/10/23	09/10/23	4	☐	100 %
1.2.3	└Entrega de la PEC1	Linck Telio Flores	10/10/23	10/10/23	1	☑	100 %
2	**PEC2. Entrega de Seguimiento - Investigación**		**11/10/23**	**07/11/23**	**26**		**100 %**
2.1	**Investigación**		**11/10/23**	**07/11/23**	**26**		**100 %**
2.1.1	└Investigación sobre ZTA yNAC	Linck Telio Flores	11/10/23	28/10/23	18	☐	100 %
2.1.2	└Investigación del uso de APIs	Linck Telio Flores	28/10/23	30/10/23	3	☐	100 %
2.1.3	└Investigación sobre las herramientas a usar	Linck Telio Flores	31/10/23	02/11/23	3	☐	100 %
2.1.4	└Elaborar los documentos PEC2	Linck Telio Flores	11/10/23	06/11/23	27	☐	100 %
2.1.5	└Entrega de la PEC2	Linck Telio Flores	07/11/23	07/11/23	1	☑	100 %
3	**PEC3. Entrega de Seguimiento - Implementación**		**08/11/23**	**05/12/23**	**27**		**100 %**
3.1	**Preparación del entorno nube**		**08/11/23**	**17/11/23**	**9**		**100 %**
3.1.1	└Instalación de Software central	Linck Telio Flores	08/11/23	09/11/23	2	☐	100 %
3.1.2	└Instalación de ClickHouse	Linck Telio Flores	09/11/23	09/11/23	1	☐	100 %
3.1.3	└Instalación de OpenSearch	Linck Telio Flores	09/11/23	09/11/23	1	☐	100 %
3.1.4	└Instalación de Grafana	Linck Telio Flores	10/11/23	12/11/23	3	☐	100 %
3.1.5	└Creación de Dashboards	Linck Telio Flores	10/11/23	17/11/23	8	☐	100 %
3.2	**Desarrollo del Sensor**		**18/11/23**	**26/11/23**	**8**		**100 %**
3.2.1	└Creación de Repositorio RPM	Linck Telio Flores	18/11/23	20/11/23	3	☐	100 %
3.2.2	└Pruebas Iniciales	Linck Telio Flores	20/11/23	21/11/23	2	☐	100 %
3.2.3	└Afinamiento del Sensor	Linck Telio Flores	21/11/23	22/11/23	2	☐	100 %
3.2.4	└Creación de archivos de configuración	Linck Telio Flores	22/11/23	22/11/23	1	☐	100 %
3.2.5	└Creación de manuales de instalación	Linck Telio Flores	23/11/23	26/11/23	4	☐	100 %
3.3	**Diseño de la Consola de Cliente Final**		**15/11/23**	**02/12/23**	**17**		**100 %**
3.3.1	└Elaboración del diseño de la interface	Linck Telio Flores	15/11/23	26/11/23	12	☐	100 %
3.3.2	└Pruebas iniciales del primer Prototipo	Linck Telio Flores	26/11/23	28/11/23	3	☐	100 %
3.3.3	└Elaboración de conectores API	Linck Telio Flores	28/11/23	02/12/23	5	☐	100 %
3.4	**Elaboración del entregable**		**18/11/23**	**05/12/23**	**17**		**100 %**
3.4.1	└Elaborar los documentos PEC3	Linck Telio Flores	18/11/23	05/12/23	18	☐	100 %
3.4.2	└Entrega de la PEC3	Linck Telio Flores	05/12/23	05/12/23	1	☑	100 %
4	**PEC4. Elaboración de la memoria final**		**06/12/23**	**09/01/24**	**33**		**100 %**
4.1	**Elaboración de la memoria final**		**06/12/23**	**09/01/24**	**33**		**100 %**
4.1.1	└Revisión del documento final	Linck Telio Flores	06/12/23	04/01/24	30	☐	100 %
4.1.2	└Elaborar conclusiones del proyecto	Linck Telio Flores	01/01/24	05/01/24	5	☐	100 %
4.1.3	└Elaborar memoria final	Linck Telio Flores	05/01/24	08/01/24	4	☐	100 %
4.1.5	└Entrega de la memoria final PEC4	Linck Telio Flores	09/01/24	09/01/24	1	☑	100 %
5	**Presentación en video**		**10/01/24**	**16/01/24**	**6**		**0 %**
5.1	**Elaboración del video**		**10/01/24**	**16/01/24**	**6**		**0 %**
5.1.1	└Creación del material para el video	Linck Telio Flores	10/01/24	15/01/24	6	☐	100 %
5.1.2	└Entrega del video de TFM	Linck Telio Flores	16/01/24	16/01/24	1	☑	100 %
6	**Defensa del TFM**		**22/01/24**	**24/01/24**	**2**		**0 %**
6.1	**Preparación de la presentación PPT**		**22/01/24**	**24/01/24**	**2**		**0 %**
6.1.1	└Elaboración de la presentación	Linck Telio Flores	22/01/24	24/01/24	3	☐	100 %
6.1.2	└Defensa Final	Linck Telio Flores	24/01/24	24/01/24	1	☑	100 %



# EDT	TÍTULO DE LA TAREA	RESPONSABLE DE LA TAREA	FECHA DE INICIO	FECHA DE FIN	DURACIÓN	HITO	% COMPLETADO DE LA TAREA
1	**PEC1. Plan de trabajo**		**27/09/23**	**10/10/23**	**13**		**100 %**
1.1	**Planificación del Proyecto**		**27/09/23**	**08/10/23**	**11**		**100 %**
1.1.1	Definición del nombre del proyecto	Linck Tello Flores	27/09/23	08/10/23	12	☐	100 %
1.1.2	Definición de la metodología a usar	Linck Tello Flores	02/10/23	02/10/23	1	☐	100 %
1.1.2	Definición de objetivos	Linck Tello Flores	02/10/23	03/10/23	2	☐	100 %
1.2	**Elaboración del Cronograma de trabajo**		**03/10/23**	**10/10/23**	**7**		**100 %**
1.2.1	Definición de tareas	Linck Tello Flores	03/10/23	04/10/23	2	☐	100 %
1.2.2	Definición de entregables	Linck Tello Flores	06/10/23	09/10/23	4	☐	100 %
1.2.3	Entrega de la PEC1	Linck Tello Flores	10/10/23	10/10/23	1	☑	100 %
2	**PEC2. Entrega de Seguimiento - Investigación**		**11/10/23**	**07/11/23**	**26**		**100 %**
2.1	**Investigación**		**11/10/23**	**07/11/23**	**26**		**100 %**
2.1.1	Investigación sobre ZTA y NAC	Linck Tello Flores	11/10/23	28/10/23	18	☐	100 %
2.1.2	Investigación del uso de APIs	Linck Tello Flores	28/10/23	30/10/23	3	☐	100 %
2.1.3	Investigación sobre las herramientas a usar	Linck Tello Flores	31/10/23	02/11/23	3	☐	100 %
2.1.4	Elaborar los documentos PEC2	Linck Tello Flores	11/10/23	06/11/23	27	☐	100 %
2.1.5	Entrega de la PEC2	Linck Tello Flores	07/11/23	07/11/23	1	☑	100 %
3	**PEC3. Entrega de Seguimiento - Implementación**		**08/11/23**	**05/12/23**	**27**		**100 %**
3.1	**Preparación del entorno nube**		**08/11/23**	**17/11/23**	**9**		**100 %**
3.1.1	Instalación de Software central	Linck Tello Flores	08/11/23	09/11/23	2	☐	100 %
3.1.2	Instalación de ClickHouse	Linck Tello Flores	09/11/23	09/11/23	1	☐	100 %
3.1.3	Instalación de OpenSearch	Linck Tello Flores	09/11/23	09/11/23	1	☐	100 %
3.1.4	Instalación de Grafana	Linck Tello Flores	10/11/23	12/11/23	3	☐	100 %
3.1.5	Creación de Dashboards	Linck Tello Flores	10/11/23	17/11/23	8	☐	100 %
3.2	**Desarrollo del Sensor**		**18/11/23**	**26/11/23**	**8**		**100 %**
3.2.1	Creación de Repositorio RPM	Linck Tello Flores	18/11/23	20/11/23	3	☐	100 %
3.2.2	Pruebas Iniciales	Linck Tello Flores	20/11/23	21/11/23	2	☐	100 %
3.2.3	Afinamiento del Sensor	Linck Tello Flores	21/11/23	22/11/23	2	☐	100 %
3.2.4	Creación de archivos de configuración	Linck Tello Flores	22/11/23	22/11/23	1	☐	100 %
3.2.5	Creación de manuales de instalación	Linck Tello Flores	23/11/23	26/11/23	4	☐	100 %
3.3	**Diseño de la Consola de Cliente Final**		**15/11/23**	**02/12/23**	**17**		**100 %**
3.3.1	Elaboración del diseño de la interface	Linck Tello Flores	15/11/23	26/11/23	12	☐	100 %
3.3.2	Pruebas iniciales del primer Prototipo	Linck Tello Flores	26/11/23	28/11/23	3	☐	100 %
3.3.3	Elaboración de conectores API	Linck Tello Flores	28/11/23	02/12/23	5	☐	100 %
3.4	**Elaboración del entregable**		**18/11/23**	**05/12/23**	**17**		**100 %**
3.4.1	Elaborar los documentos PEC3	Linck Tello Flores	18/11/23	05/12/23	18	☐	100 %
3.4.2	Entrega de la PEC3	Linck Tello Flores	05/12/23	05/12/23	1	☑	100 %
4	**PEC4. Elaboración de la memoria final**		**06/12/23**	**09/01/24**	**33**		**100 %**
4.1	**Elaboración de la memoria final**		**06/12/23**	**09/01/24**	**33**		**100 %**
4.1.1	Revisión del documento final	Linck Tello Flores	06/12/23	04/01/24	30	☐	100 %
4.1.2	Elaborar conclusiones del proyecto	Linck Tello Flores	01/01/24	05/01/24	5	☐	100 %
4.1.3	Elaborar memoria final	Linck Tello Flores	05/01/24	08/01/24	4	☐	100 %
4.1.5	Entrega de la memoria final PEC4	Linck Tello Flores	09/01/24	09/01/24	1	☑	100 %
5	**Presentación en video**		**10/01/24**	**16/01/24**	**6**		**0 %**
5.1	**Elaboración del video**		**10/01/24**	**16/01/24**	**6**		**0 %**
5.1.1	Creación del material para el video	Linck Tello Flores	10/01/24	15/01/24	6	☐	100 %
5.1.2	Entrega del video de TFM	Linck Tello Flores	16/01/24	16/01/24	1	☑	100 %
6	**Defensa del TFM**		**22/01/24**	**24/01/24**	**2**		**0 %**
6.1	**Preparación de la presentación PPT**		**22/01/24**	**24/01/24**	**2**		**0 %**
6.1.1	Elaboración de la presentación	Linck Tello Flores	22/01/24	24/01/24	3	☐	100 %
6.1.2	Defensa Final	Linck Tello Flores	24/01/24	24/01/24	1	☑	100 %

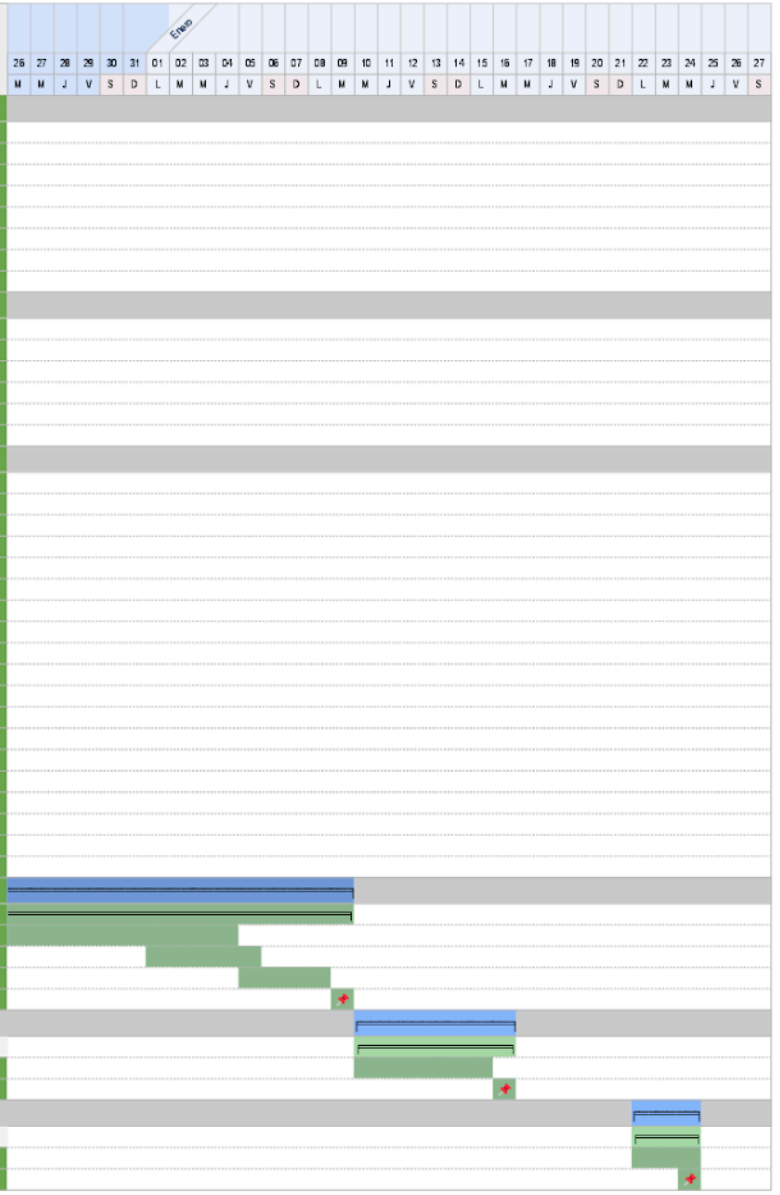


Figura 2: Cronograma del trabajo - Diagrama de Gantt

1.7. Estado del arte

La adecuada gestión de las amenazas de seguridad partiendo de la detección y control de los dispositivos conectados a la red es un elemento clave de una Arquitectura de Confianza Cero o Zero Trust Architecture (ZTA) descrita por el Instituto Nacional de Estándares y Tecnología (NIST) en la publicación especial 800-207 de agosto de 2020 donde aborda el tema y aclara la forma como se debe realizar la implementación de la confianza cero (zero trust) en las organizaciones.

Igualmente el SANS Institute en el informe SANS 2022: “Avanzando hacia un estado de zero trust” en la página 4 a 5, “Adoptando zero trust” hace una descripción de los elementos clave identificados por NIST en la que se indica que [un] elemento clave de una ZTA es “La capacidad de identificar e inventariar los recursos [activos]…..” con las que cuenta la organización con el fin de alinear a la empresa a una arquitectura de confianza cero para permitir garantizar la seguridad de la red y es esto uno de los objetivos del presente trabajo al realizar la detección y control posterior del dispositivo que se conecta a la red.

Así mismo, en la actualidad existen en el mercado fabricantes de soluciones de Control de Acceso a la Red o Network Admission Control (NAC), siendo uno de los principales el fabricante CISCO, quien en el Framework Deployment Guide indica que “*NAC utiliza la infraestructura de red para hacer cumplir la política de seguridad en todos los dispositivos que buscan acceder a los recursos informáticos de la red, limitando así los daños causados por amenazas de seguridad emergentes, como virus, gusanos y software espía*.”, sin embargo, para implementar la tecnología de CISCO NAC se requiere contar con equipos de comunicaciones que soporten dicha tecnología lo que muchas veces no ocurre puesto que las empresas utilizan una variedad de marcas y modelos de equipamiento de comunicaciones haciendo que la implementación sea finalmente un asunto netamente económico por el costo que significa cambiar el equipamiento de red.

El control de acceso a la red suele implementarse a nivel de enlace (capa dos) o a nivel de red (capa tres) del modelo de interconexión de estándares abiertos (OSI). Los mecanismos de aplicación varían entre los distintos productos, y algunos tienen múltiples opciones, encontramos por ejemplo en el mercado productos como MACMON NAC[10], ForeScout[11], FortiNAC[12] entre otros productos propietarios que brinda una serie de características para el control de acceso a la red y a nivel de open-source encontramos a PacketFence[13] que cuenta con características similares a los productos comerciales.

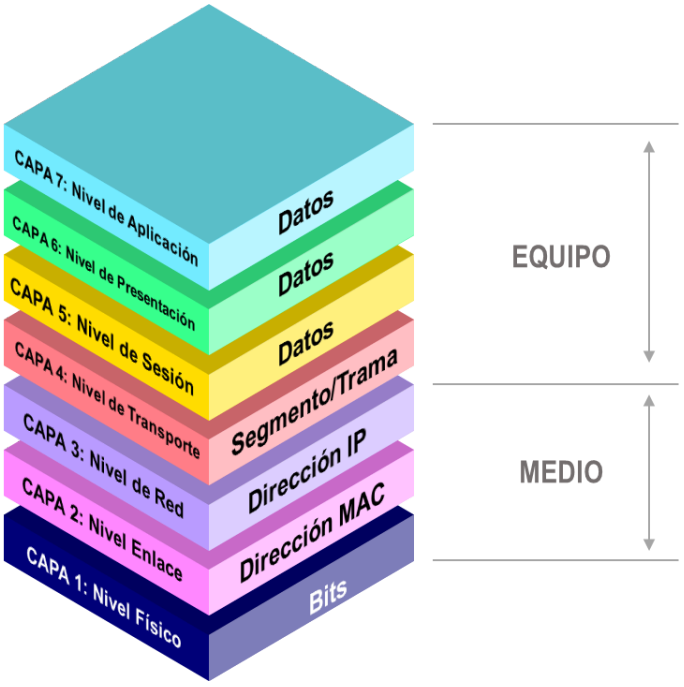


Figura 3: El modelo OSI (Elaboración propia)

Las soluciones NAC se pueden utilizar para diversos fines, pero algunos de los principales casos de uso incluyen:

- NAC para acceso de invitados y socios
- NAC para BYOD
- NAC para IoT
- NAC para respuesta a incidentes (IR)

[10] https://www.macmon.eu/en/
[11] https://www.forescout.com/
[12] https://www.fortinet.com/lat/products/network-access-control
[13] https://www.packetfence.org/

Como vemos tanto la arquitectura ZTA y NAC persiguen objetivos que se entrelazan, siendo uno de ellos parte del objetivo propuesto que es la de detectar y controlar todos los dispositivos que se conectan a la red mediante una variedad de integraciones y capacidades de analítica en una primera versión y posteriormente usando el análisis de big data y la inteligencia artificial para la toma de decisiones rápidas que un analista de seguridad humano no puede realizar ya sea porque no cuenta con la información en tiempo real, por la cantidad de datos que se deben procesar o por el tiempo de cobertura que se debe tener para una adecuada protección de la red que es de 24 horas los 7 días de la semana.

2. Fase de investigación

En el pasado, las redes empresariales se construían asumiendo que las amenazas provenían desde fuera de la organización y que todos los usuarios internos son igualmente dignos de confianza. Este fue el caso en un momento, cuando los puntos finales (estaciones, portátiles) eran administrados por la empresa, y antes de que la expectativa general de acceso a Internet se convirtiera en un requisito de trabajo. En aquellos días, una buena seguridad significaba una buena seguridad perimetral brindado por un **cortafuegos** y tal vez algún software de antivirus de escritorio.

Es bien sabido que las cosas han cambiado drásticamente, creando la necesidad de medidas de seguridad adicionales dentro de la propia red. Ya no es fácil controlar qué dispositivos están conectados a la red: las computadoras portátiles de los usuarios compartidas con otros miembros de la familia, las tabletas e incluso los teléfonos inteligentes pueden "conectarse y funcionar", obtener una dirección dinámica y acceder a casi cualquier cosa, desde el correo grupal, acceso a recursos compartidos y aplicaciones críticas para el negocio.

Además, diversas razones comerciales han provocado que la red se abra a invitados, contratistas temporales y socios de subcontratación, entre otros, independientemente de si estos usuarios tienen puntos finales no administrados o incluso inmanejables.

La problemática es recurrente relativa a la protección de los entornos corporativos frente a la infiltración de los ciberdelincuentes. Los departamentos de informática y en especial los profesionales encargados de la ciberseguridad continúan enfrentando una sucesión ininterrumpida de intrusiones, incluso en situaciones en las que se han implementado planes y medidas de seguridad de alta complejidad. Según el informe anual “El estado del ransomware 2023” de Sophos[14] revela que el 66% de organizaciones encuestadas indican haber sufrido un incidente de ransomware en 12 meses y el 82% de vulneraciones de

[14] https://www.sophos.com/es-es/content/state-of-ransomware

datos afectaron a datos almacenados en el cloud (en entornos públicos, privados o múltiples) según el informe de “Informe sobre el coste de una vulneración de datos de 2023 IBM Security [15].

Esta problemática se ha visto intensificada de manera significativa al analizar la composición de los entornos organizacionales que se vieron aceleradas desde el inicio de la COVID-19, como por ejemplo en:

- El aumento en el uso de tecnologías y servicios en la nube, así como en aplicaciones de terceros.
- La fuerza de trabajo cada vez más remota, móvil y global que requiere la capacidad de realizar su trabajo desde cualquier lugar.
- La ampliación de la superficie de ataque del entorno corporativo para incluir una amplia gama de aplicaciones de cara al exterior.
- El uso de dispositivos personales en las organizaciones debido principalmente a la fuerza de trabajo remota, socios de negocios e invitados.

Superar las necesidades de seguridad de estos distintos elementos puede ser una tarea muchas veces titánica para cualquier organización, desde las pequeñas hasta las grandes compañías. Muchos han intentado utilizar tecnologías diseñadas para arquitecturas locales complejas, pero se está descubriendo que no son adecuadas para proteger la empresa moderna. Si bien algunos equipos de seguridad son buenos con las implementaciones personalizadas o usando servicios de terceros, si un ciberdelincuente, tiene acceso a la red y se apodera de la cuenta incorrecta con los permisos correctos, podría pasar rápidamente a tomar el control de la red.

Para combatir este tipo de problemas, las organizaciones deben aprovechar las tecnologías que permiten un control granular del acceso de los usuarios y un acceso con mínimos privilegios, limitando las acciones que puede realizar un ciberdelincuente si consigue acceder a la red. Para ello en la actualidad existen

[15] https://es.newsroom.ibm.com/announcements?item=122679

dos modelos de seguridad que abordaremos para la implementación del proyecto propuesto las cuales son:

1) El modelo de seguridad basado en el **Control de Acceso a la Red o NAC**.
2) Y el modelo de seguridad conocido como **Zero Trust Architecture, ó ZTA**.

Como veremos a continuación el control de acceso a la red o NAC es vital para las organizaciones, ya que les permite supervisar a los usuarios y dispositivos autorizados y no autorizados que intentan acceder a su red. Este control es crucial para garantizar que sólo los usuarios autorizados puedan acceder a las redes, datos, dispositivos y recursos de software. Además, el **NAC** desempeña un papel importante en la implementación de una arquitectura de confianza cero (ZTA).

2.1. Network Access Control o NAC

El Control de Acceso a la red o NAC es un conjunto de tecnologías y soluciones basadas en una iniciativa industrial liderada por Cisco Systems®[16]. NAC utiliza la infraestructura de red para hacer cumplir la política de seguridad en todos los dispositivos que buscan acceder a los recursos informáticos de la red, limitando así los daños causados por amenazas de seguridad emergentes, como virus, gusanos y software espía. Los clientes que utilizan NAC pueden permitir el acceso a la red solo a dispositivos terminales confiables y compatibles (por ejemplo computadoras, servidores, tabletas, etc.) y pueden restringir el acceso de dispositivos que no cumplan.

Las soluciones NAC se han convertido en una herramienta valiosa para mejorar la seguridad de la red, ya que sirven para abordar el aumento de los dispositivos Bring Your Own Device (BYOD) e Internet de las cosas (IoT), además de ayudar a mitigar las amenazas avanzadas de día cero, segmentar

[16] https://www.cisco.com/c/en/us/products/security/what-is-network-access-control-nac.html

el tráfico de producción y de invitados, simplificar el aprovisionamiento de dispositivos como teléfonos VoIP y más.

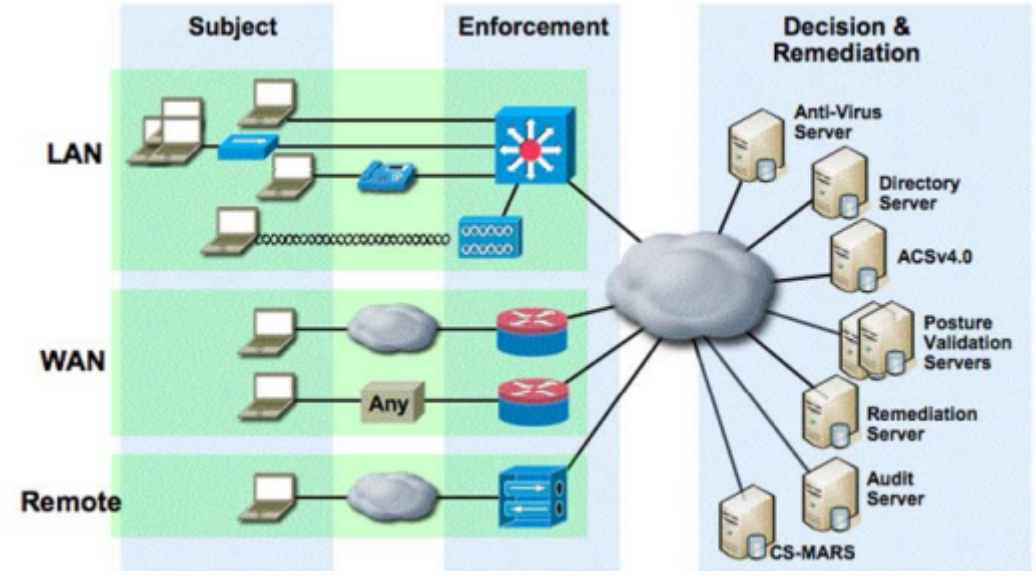


Figura 4: Escenario de implementación de NAC de CISCO

Anteriormente, los usuarios y dispositivos eran autenticados en cuanto a quiénes o qué eran, pero no en cuanto a su condición, así mismo NAC:

- Ayuda a garantizar que sólo las estaciones de trabajo cliente en buen estado tengan acceso completo a la red.
- Trabaja con software antivirus, de administración de parches y de firewall personal para evaluar la condición, llamada postura de un cliente antes de permitirle el acceso a la red.
- Ayuda a garantizar que un cliente de red tenga un conjunto de firmas de virus actualizadas, los parches más recientes del sistema operativo y no esté infectado. Si el cliente requiere una actualización de la firma antivirus o una actualización del sistema operativo.
- Indica al cliente que complete las actualizaciones necesarias. Si el cliente se ha visto comprometido o si se produce un brote de virus en la red, NAC coloca al cliente en un segmento de red en cuarentena.

Después de que el cliente haya completado su proceso de actualización o desinfección, el cliente se verifica nuevamente y se devuelve a un estado saludable con acceso normal a la red.

2.2. Arquitectura Zero Trust o ZTA

Este nuevo modelo de seguridad está diseñado para ayudar a las organizaciones a crecer a la velocidad que deseen, ampliar la superficie de ataque a cualquier tamaño y con ello seguir proporcionando los controles necesarios para limitar a los equipos y usuarios acceso sólo a los datos y aplicaciones que necesitan.

Zero Trust Architecture está basada en la idea de privilegios mínimos y una evaluación continua de la confianza, la ZTA cambia el modelo para muchas organizaciones. Normalmente, los controles de seguridad se han centrado en defensas perimetrales que, una vez que entra un usuario autenticado y por correlación el equipo desde donde se conecta a la red al usuario se le conceden acceso a una amplia gama de recursos con la esperanza de que acceda sólo a los que necesita. Desafortunadamente, los atacantes se han dado cuenta de esto y construyen sus planes en torno al abuso de las credenciales robadas para causar estragos en el entorno de las organizaciones.

Como se mencionó en la introducción, el Instituto Nacional de Estándares y Tecnología, NIST, realizó una Publicación Especial 800-207 para abordar la necesidad de aclarar e implementar zero trust. Zero Trust o Confianza Cero (ZT) "*es el término para un conjunto en evolución de paradigmas de ciberseguridad que trasladan las defensas de perímetros estáticos basados en redes para centrarse en los usuarios,* ***los activos y los recursos****. Una arquitectura de confianza cero (ZTA) utiliza principios de confianza cero para planificar la infraestructura y los flujos de trabajo industriales y empresariales. La confianza cero supone que no se otorga ninguna confianza implícita a los activos o cuentas de usuario basándose únicamente en su ubicación física o de red (es decir, redes de área local frente a Internet) o en función de la propiedad de los activos (empresarial o personal)*" [17].

[17] https://csrc.nist.gov/pubs/sp/800/207/final

2.3. Requerimientos de la Interfaces de programación de aplicaciones o APIs

Hemos indicado que se usarán las APIs de los distintos fabricantes de seguridad para realizar las comprobaciones de seguridad necesarias para lograr realizar la identificación, evaluación y respuesta autónoma de seguridad en la plataforma.

Estas APIs deberán cumplir con los siguientes requerimientos:

a) Deben ser APIs públicas.
b) Deben contar con documentación accesible y libre para el desarrollo.
c) Deben permitir la conexión vía HTTPS.
d) Deben permitir la autenticación segura ya sea con un usuario y contraseña o vía Tokens.
e) Deben devolver el resultado de las consultas en formato JSON.
f) Deben permitir realizar consultas a la lista de equipos administrados y obtener el detalle de los mismos.
g) Deben permitir interactuar con la plataforma para crear políticas o reglas de seguridad.

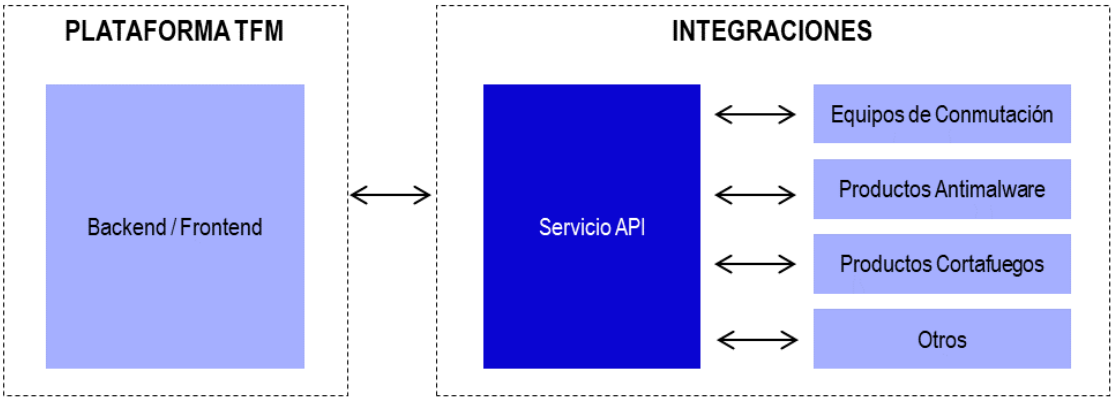


Figura 5: Representación general de la integración del prototipo vía APIs (Elaboración propia)

2.4. Componentes del proyecto

La plataforma proporciona funciones de SIEM para el almacenamiento, análisis de datos de registro, monitoreo y evaluación de puntos finales que se conectan a la red y mediante la integración con productos de terceros como antimalware, cortafuegos, equipos de conmutación y otros se ejecutarán respuestas automatizadas basadas las políticas se seguridad predefinidas por especialistas en ciberseguridad.

La solución está basada en un Sensor de red que se instala en la red de los clientes y está se comunica el SIEM Central que forma parte de los Componente Centrales de la plataforma qué está compuesto además por el Backend (Panel de Administración Central) y el Frontend (Consola de administración para la empresa), así mismo estos componentes se comunican (flujo de información) entre sí las cuales se muestran en forma general en la figura 6.

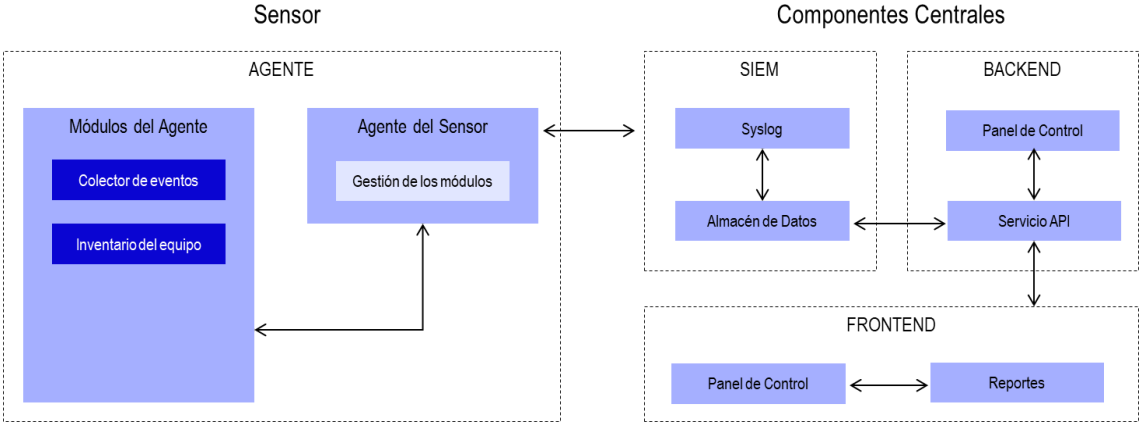


Figura 6: Arquitectura general y vista del flujo de información

2.5. Arquitectura

Los sensores ubicados en la red de las empresas se ejecutarán en modo port-mirror o promiscuo y se encargarán de detectar todos los puntos finales que obtienen una dirección IP en la red, estos datos son enviados al SIEM Central que forma parte de los Componentes Centrales y es gestionado por el proveedor de la solución. El SIEM Central se encarga de analizar la

información entrante y guarda los resultados en el almacén de datos luego de su indexación con la finalidad que desde el Frontend vía los servicios API del Backend se realicen consultas tanto a los datos almacenados como hacia los fabricantes (Integraciones) vía API; así mismo, se encargará de la gestión propia de la plataforma con el fin de administrar la seguridad de la organización.

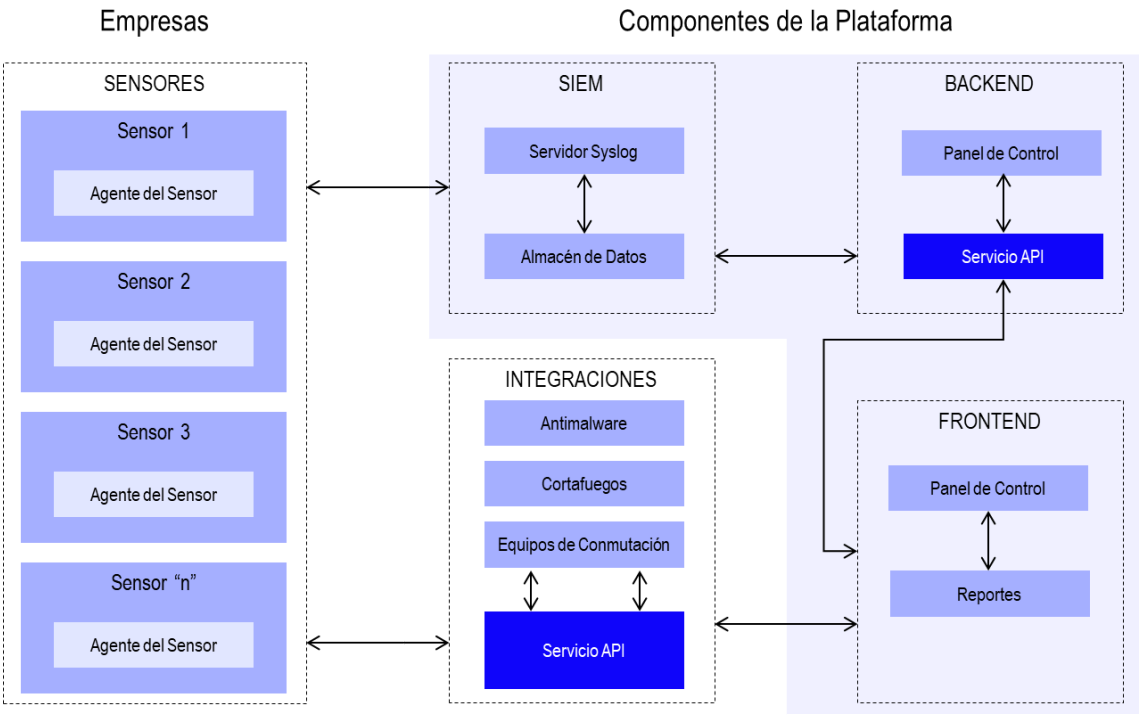


Figura 7: Arquitectura de la plataforma en desarrollo

2.6. Herramientas de código abierto analizadas

Para el desarrollo de la plataforma propuesta se requieren de distintas herramientas y software desde el sistema operativo donde se instalarán los distintos componentes a desarrollar como las herramientas y software de análisis de los datos que son generados por los sensores o para su almacenamiento y posterior consulta.

En este capítulo se analizan las herramientas de código abierto OpenSearch, ClickHouse, Grafana y Zeek las mismas que permitirán realizar el análisis de datos, el almacenamiento de registros, la visualización de datos y la detección de conexiones a la red respectivamente, para qué sirven, cómo funcionan y cuál es el propósito de las herramientas en el desarrollo del presente trabajo.

OpenSearch

OpenSearch[18] es una suite de análisis y búsquedas de código abierto con licencia de Apache 2.0. El proyecto OpenSearch, creado por Amazon, es un proyecto bifurcado basado en versiones antiguas de Elasticsearch y Kibana.

OpenSearch proporciona búsquedas de alto rendimiento gracias a la librería de Apache Lucene, admitiendo diversas capacidades de búsqueda y análisis, como lo son la búsqueda de k vecinos más cercanos (KNN), SQL, detección de anomalías, Machine Learning Commons, análisis de rastreos, búsqueda de texto completo y otras características.

OpenSearch está formado por una arquitectura distribuida, en donde cada uno de los actores o aplicaciones interactúan con un clúster de OpenSeach.

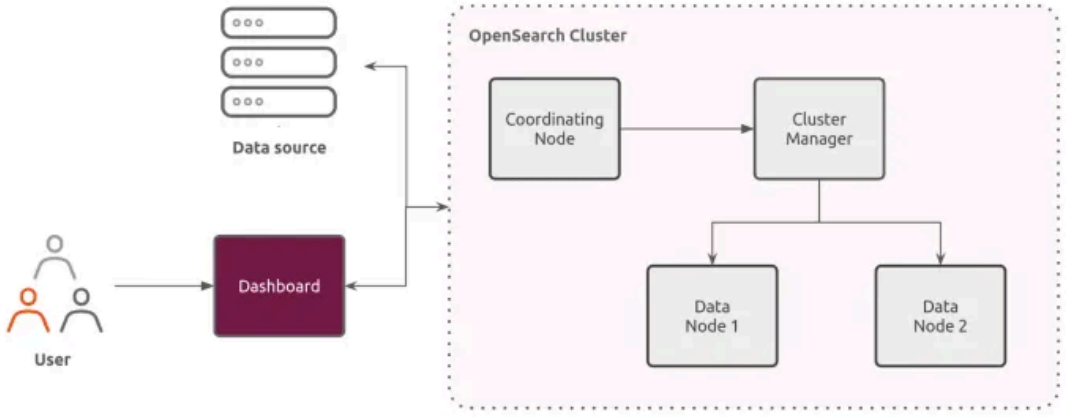


Figura 8: Arquitectura de OpenSearch
(Fuente: https://medium.com/@diego.coder/introducci%C3%B3n-a-opensearch-e324e88c1891)

[18] https://www.opensearch.org/

Con OpenSearch, puede realizar los siguientes casos de uso:

			
Búsqueda de texto completo rápida y escalable	**Monitoreo de infraestructura y aplicaciones**	**Gestión de información de eventos y seguridad**	**Seguimiento del estado operativo**
Ayuda a los usuarios a encontrar la información correcta dentro de su aplicación, sitio web o catálogo de lago de datos.	Almacena y permite analizar fácilmente datos de registro y establece alertas automáticas en caso de rendimiento deficiente.	Centraliza los registros para permitir el monitoreo de seguridad y el análisis forense en tiempo real.	Utiliza registros, métricas y seguimientos de observabilidad para monitorear aplicaciones y los negocios en tiempo real.

Para la plataforma a desarrollar el caso de uso es la de la Gestión de información de eventos y seguridad.

ClickHouse

Clickhouse[19] es un sistema de gestión de bases de datos (DBMS) orientado a columnas de código abierto que se utiliza para el procesamiento analítico en línea (OLAP) creado por Yandex. Actualmente, impulsa la segunda plataforma de análisis web más grande, Yandex Metrica. También puede considerarse el primer almacén de datos SQL de código abierto que jamás haya igualado la escalabilidad y el rendimiento de bases de datos como Veryica y Snowflake, según sus desarrolladores.

Lanzado en código abierto en 2016, ClickHouse es utilizado por Yandex para fines de KPI y monitoreo de accesibilidad del sitio. También se ha implementado en el experimento LHCb del CERN, donde almacena y procesa

[19] https://clickhouse.com/

metadatos en 10 mil millones de eventos que albergan más de 1000 atributos en un evento.

Clickhouse es utilizado principalmente por analistas/ingenieros de DevOps/Desarrolladores, Startups que buscan análisis de alta calidad con bajo capital y empresas que pagan grandes cantidades de dinero por la arquitectura.

Cómo funciona Clickhouse

A diferencia de la mayoría de las bases de datos propietarias, el desarrollo de Clickhouse es impulsado por una comunidad comprometida compuesta por cientos de colaboradores enfocados en crear una mejor funcionalidad y resolver problemas que pueden degradar su rendimiento.

Al utilizar todo el hardware disponible para procesar cada consulta, la aplicación puede procesar desde 100 millones hasta más de mil millones de filas y gigas de datos por ciclo de servidor de un segundo.

Clickhouse permite a las empresas y desarrolladores agregar servidores a sus clústeres sin inyectar muchos recursos en la modificación de DBMS.

Características de Clickhouse

Estas son algunas de las principales características de ClickHouse DBMS:

1. Ofrece escalabilidad lineal
2. Almacenamiento y procesamiento de petabytes de datos
3. Compresión de datos
4. Optimización de HDD
5. Tolerancia a fallos
6. Alto rendimiento, por ejemplo, el procesamiento de consultas distribuidas y paralelas
7. Soporte para SQL

```
SELECT
    toStartOfMonth(upload_date) AS month,
    sum(view_count) AS `Youtube Views`,
    bar(sum(has_subtitles) / count(), 0.55, 0.7, 100) AS `% Subtitles`
FROM youtube
WHERE (month >= '2020-08-01') AND (month <= '2021-08-01')
GROUP BY month
ORDER BY month ASC

13 rows in set. Elapsed: 0.823 sec Processed 1.07 billion rows, 11.75 GB (1.30 billion rows/s., 14.27 GB/s.)
```

Figura 9: Ejemplo de consulta SQL en ClickHouse

Ventajas de ClickHouse

- Procesamiento distribuido en varios servidores
- Es fácil de configurar y tiene buena documentación y comunidad
- ClickHouse es eficaz cuando se trabaja con tablas desnormalizadas/anchas
- Soporte de índice
- Escaneos rápidos que se pueden utilizar para consultas en tiempo real
- Utilización de múltiples núcleos en procesamiento paralelo para consultas únicas
- Línea de comandos fácil de usar

ClickHouse es una excelente solución para trabajar con grandes volúmenes de datos y realizar análisis en tiempo real. Con su arquitectura de columnas y su amplia gama de funciones analíticas integradas, ClickHouse es una herramienta poderosa para el desarrollo del presente proyecto.

Grafana

Grafana[20] es una plataforma interactiva y open source de visualización de datos desarrollada por Grafana Labs. Está plataforma permite a los usuarios ver los

[20] https://grafana.com/

datos a través de tablas y gráficos que se unifican en un panel de control (o en varios) para facilitar la interpretación y la comprensión. También permite realizar consultas y configurar avisos sobre la información y los indicadores desde el lugar en el que se almacena dicha información, ya sea en entornos de servidores tradicionales, clústeres de Kubernetes y varios servicios de nube, entre otros. De esta forma, podrá analizar los datos e identificar las tendencias y las inconsistencias con mayor facilidad, por lo que sus procesos serán más eficientes.

Figura 10: Panel de control elaborado en Grafana

Grafana se diseñó según los principios del open source (basado en la licencia Apache 2.0[21]), y con la creencia de que los datos deberían ser accesibles para todo tipo de empresas y no solo para unas pocas. Esto fomenta una cultura por la cual cualquier persona que necesite los datos puede encontrarlos y utilizarlos fácilmente, lo cual permite que los equipos sean más abiertos, innovadores y colaborativos.

[21] https://es.wikipedia.org/wiki/Apache_License

Paneles de control

Los paneles de control de Grafana otorgan un nuevo significado a los datos recolectados desde varias fuentes, ya que se pueden compartir con otros equipos o miembros, lo que permite que colaboren y realicen análisis más amplios de los datos y sus implicaciones. Los paneles de control se pueden diseñar según se requiera tanto individualmente como en equipo y permite personalizarlos para crear las visualizaciones que se desee mediante las funciones de consulta y transformación avanzadas.

Grafana permite que los datos se trasladen entre los equipos y sus miembros sin problemas para que puedan observarlos de forma óptima y, de esta manera, llegar a la raíz de un problema u obtener un resultado rápidamente y resolverlo.

La información de los paneles de control de Grafana puede compartirse:

- En toda una organización, incluso con aquellos colegas que no utilizan la herramienta.
- En toda la comunidad de Grafana, en cualquier parte del mundo.
- Donde sea que se encuentre: puede ver los paneles de control en cualquier dispositivo y desde cualquier parte.

Características clave

- Paneles: permite visualizar los datos de la forma que se desee mediante histogramas, gráficos, mapas geográficos, mapas de calor, etc.
- Plugins: permite procesar información de forma inmediata mediante una API fácil de usar a través de plugins en los paneles que se conectan a las fuentes de datos sin necesidad de trasladarlos. También permite crear plugins de fuentes de datos para obtener indicadores de cualquier API personalizada.
- Alertas: permite crear, consolidar y controlar todas sus alertas en una única interfaz.
- Transformaciones: permite realizar cambios de nombre, resúmenes, combinaciones y cálculos en todas las fuentes de datos y consultas.

- Anotaciones: permite utilizar eventos completos de diferentes fuentes de datos para hacer anotaciones en los gráficos.
- Editor de paneles: brinda una interfaz de usuario uniforme para configurar y personalizar los paneles.

Zeek

Zeek[22] (antes Bro) es una herramienta de código abierto para supervisar la seguridad de la red. Zeek tiene una larga historia en el mundo del código abierto y la seguridad digital. Vern Paxson comenzó a desarrollar el proyecto en la década de 1990 con el nombre de "Bro" como medio para comprender lo que ocurría en las redes de su universidad y laboratorio nacional. Vern y el equipo de liderazgo del proyecto cambiaron el nombre de Bro a Zeek a finales de 2018 para celebrar su expansión y desarrollo continuo.

Zeek no es un dispositivo de seguridad activo, como un cortafuegos o un sistema de prevención de intrusiones. Más bien, Zeek se asienta en un "sensor", una plataforma de hardware, software, virtual o en la nube que observa de forma silenciosa y discreta el tráfico de la red. Zeek interpreta lo que ve y crea registros de transacciones compactos y de alta fidelidad, contenido de archivos y resultados totalmente personalizados, adecuados para la revisión manual en disco o en una herramienta más fácil de analizar, como un sistema de gestión de eventos de seguridad e información (SIEM).

Ventajas para el proyecto

Zeek ofrece muchas ventajas a los equipos de seguridad y redes que desean comprender mejor cómo se utiliza su infraestructura y es esta característica del diseño de Zeek que vamos a usar en el proyecto mediante la recolección y análisis de los datos, puesto que Zeek está diseñado para observar el tráfico de red en tiempo real.

[22] https://zeek.org/

Cuando los equipos de seguridad van a realizar su trabajo generalmente recurren a cuatro tipos de fuentes de datos cuando se trata de detectar y responder a actividades sospechosas y maliciosas.

a) Fuente de datos de terceros (p.e. Fuente de datos sobre inteligencia de amenazas).
b) Datos de red.
c) Datos de la infraestructura y aplicaciones y
d) Datos de puntos finales.

Zeek es principalmente una plataforma para recopilar y analizar la segunda forma de datos: **Los datos de red**. Estos datos se recolectan en forma de registros que resumen los protocolos y archivos vistos atravesando la red. Se debe tener en cuenta que Zeek no recopila datos de contenido en formato Pcap, aunque sí los puede leer e interpretar, sin embargo, otros proyectos de código abierto que están orientados a otro tipo de trabajo como la detección de intrusos como Suricata[23] o Snort[24] si pueden hacerlo.

Registros

Los registros generados por Zeek se guardan en formato tradicional, así como en formato JSON y son las siguientes:

Registros	Descripción
conn.log	Registra las conexiones de red.
dns.log	Registra las consultas DNS.
http.log	Registra el tráfico HTTP.

[23] https://suricata.io/
[24] https://www.snort.org/

files.log	Registra los archivos observados en la inspección del tráfico de red.
ftp.log	Registra el resumen de la actividad realizada mediante el Protocolo de Transferencia de Archivos (FTP)
ssl.log	Registra la actividad vista en los protocolos SSL/TLS
x509.log	Registra la captura de detalles sobre los certificados intercambiados durante determinadas negociaciones TLS.
smtp.log	Registra la actividad del protocolo SMTP.
ssh.log	Registra la actividad de las sesiones SSH.
pe.log	Registra el tránsito de archivos ejecutables o PE.
dhcp.log	Registra la actividad del protocolo DHCP.
ntp.log	Registra la actividad del protocolo NTP.
smb_files.log	Registra la actividad del protocolo SMB (CIFS).
irc.log	Registra la actividad del protocolo IRC.
rdp.log	Registra la actividad del protocolo RDP.
ldap.log y ldap_search.log	Registra la actividad del protocolo LDAP.
traceroute.log	Registra la actividad de los protocolos ICMP y UDP.
tunnel.log	Registra la actividad del tráfico generado por Túneles y VPN

dpd.log	Registra la actividad del protocolo DPD.
known_*.log y software.log	Registran la actividad de algunos aspectos de la red local, como certificados SSL/TLS, direcciones IP de host, servicios y aplicaciones.
weird.log y notice.log	Registran la actividad de cosas qué parecen fuera de lo normal o no convencional.
capture_loss.log	Registra el análisis del tráfico perdido.
reporter.log	Registra el análisis de advertencias y errores internos.

2.6.1. Tabla resúmen de las herramientas analizadas

A continuación se realiza un resúmen del propósito de las herramientas analizadas y su uso en la plataforma a desarrollar.

Herramienta	Propósito	Ventajas	Inconvenientes
OpenSearch	OpenSearch es un motor de búsqueda y análisis distribuido, basado en Elasticsearch. Se utilizará para el desarrollo de la interface de Administración Central de la plataforma y permite indexar y buscar en grandes volúmenes de datos en tiempo real, siendo especialmente útil en aplicaciones de	• Es un proyecto de código abierto el cual permite accesibilidad y brinda posibilidades para su modificación y adaptación al proyecto planteado. • Es altamente escalable el cual permite manejar grandes volúmenes de información y	• La curva de aprendizaje para aprovechar al máximo las capacidades de OpenSearch es media-alta. • La configuración sobre todo en entornos de Cluster puede

	búsqueda y análisis de registros que los sensores van a generar y enviar.	escalamiento horizontal al brindar la posibilidad de añadir nodos cuando se implementa un cluster de servidores. ● Es rápido para la búsqueda de información y sumamente eficiente. ● Se integra fácilmente con otras herramientas y plataformas lo que facilita la inclusión en cualquier tipo de entorno. ● Es altamente configurable lo que permite adaptarlo a cualquier necesidad y casos de uso.	ser complejo por lo qué se requiere de personal con experiencia para su implementación ● Aunque no se ha planteado en el trabajo la integración con Elasticsearch se debe indicar qué pueden aparecer problemas de compatibilidad con las versiones más recientes de Elasticsearch. ● La escalabilidad y el uso de recursos de hardware así como el mantenimiento a gran escala requieren de mucha potencia y ancho de banda lo que requiere de una

			cuidadosa planificación y gestión.
ClickHouse	ClickHouse es un sistema de gestión de bases de datos basado en columnas diseñado para realizar consultas analíticas de manera eficiente en grandes conjuntos de datos. Es ideal para el tipo de plataforma que se desarrollará puesto que se requerirá del análisis rápido de datos, análisis de registros y generación de informes.	● Cuenta con optimización para realizar consultas analíticas como las planteadas en el TFM de manera extremadamente veloz. ● Usa una estructura de almacenamiento de datos columnar, lo qué significa qué sólo se leen las columnas necesarias para una consultas, método que reduce significativamente el tiempo de respuesta. ● La escalabilidad horizontal está creada desde su diseño, permitiendo añadir nodos a implementaciones de cluster lo que mejora enormemente el rendimiento.	● La curva de aprendizaje puede representar un problema ya qué se requieren de conocimientos especializados para la configuración de clusters. ● No es ideal para realizar operaciones de actualización de datos, para este fin para el desarrollo de las interfaces de consultas de datos qué requieran alta operaciones de actualización/eliminación debe plantearse usar base de datos convencionales como

		• Usa algoritmos de compresión de datos qué reducen el espacio de disco para el almacenamiento lo qué es muy adecuado cuando se manejan grandes volúmenes de datos. • El soporte para datos de series temporales que son parte de los datos que se reciben de los sensores hace qué su uso sea adecuado para el análisis de los datos en tiempo real y para la generación de reportes históricos. • Es fácil de usar ya que usa un tipo de lenguaje SQL estándar por lo qué facilita la creación de interfaces de consultas. • Maneja distintos tipos de datos como CSV, JSON y otros	PostgreSQL o MySQL. • El uso de recursos de hardware puede ser intensivo por lo que la optimización de la implementación es crucial.

		lo qué facilita la integración con las otras herramientas analizadas. ● Es de código abierto y está licenciado bajo Apache 2.0 lo qué brinda una gran ventaja para su integración en plataforma como la planteada en el TFM. ● Brinda funciones de replicación de datos y tolerancia a fallos lo qué es ideal para aplicaciones que manejan grande volúmenes de información.	
Grafana	Grafana es una plataforma de código abierto que permitirá realizar la visualización y monitoreo de datos centralizada. Permitirá crear paneles interactivos y cuadros de mando para visualizar métricas, registros y datos qué serán enviados por los sensores, facilitando la supervisión y la toma de	● Proporciona una interfaz gráfica intuitiva y fácil de usar para la creación de paneles para la visualización de datos. ● Soporta múltiples fuentes de datos, desde archivos a bases de datos lo qué lo hace versátil y compatible con	● Al igual que las anteriores herramientas la curva de aprendizaje puede ser un problema para su uso.. ● La configuración con ciertos tipos de fuentes de datos puede

	decisiones ya sea para ejecutar modificaciones a la plataforma o para desarrollar nuevas versiones del producto.	diferentes tecnologías. ● Incluye paneles y gráficos pre-configurados lo que permite la personalización visual para la representación de los datos de acuerdo a lo que se requiera. ● Permite la creación de alertas en base a umbrales definidos, función que es crucial para la detección temprana de problemas y para la realización de medidas proactivas. ● Es altamente adaptable y flexible lo que permite una alta adaptabilidad a diferentes preferencias y necesidades. ● Permite mostrar datos en tiempo real lo que es esencial para aplicaciones como la planteada en el presente TFM.	resultar compleja. ● La gestión de los permisos y usuarios sobre todo en implementación grandes es complejo. ● Gran consumo de recursos especialmente en entornos con muchos usuarios y grandes conjuntos de datos. ● En ciertas ocasiones se requiere la manipulación directa de los archivos de configuración en formato JSON para ciertas operaciones avanzadas.

		• Brinda alta escalabilidad por lo que puede ser implementado en pequeños y grandes proyectos.	
Zeek	Zeek es una plataforma de análisis de red de código abierto y es el componente principal de la plataforma que se instalará en los sensores. Se encargará de monitorizar y analizar el tráfico de red en tiempo real, lo que proporcionará información detallada sobre la actividad, la seguridad y posibles amenazas en la red monitorizada. Zeek para efecto práctico es valioso en entornos de ciberseguridad y monitoreo de redes, objetivos que forman parte del presente TFM.	• Realiza un análisis profundo del tráfico de red lo que es ideal para cumplir con los objetivos del presente TFM. • Permite detectar amenazas en base a patrones y comportamientos sospechosos en el tráfico de la red. • Registra las actividades de la red con metadatos completos lo que permite el análisis posterior del tráfico de la red. • Es altamente flexible y personalizable que unido a que es de código abierto con una comunidad activa de usuarios facilita su integración en proyectos como el	• Tiene una curva de aprendizaje alta, sobre todo para aquellos usuarios que no están familiarizados con los scripts de configuración. • Para una adecuada implementación requiere de altos conocimientos técnicos en redes y seguridad de la información. • Dependiendo de su implementación y alcance de los registros a analizar requiere de alto

		planteado en el presente TFM. ● Se integra con otros tipos de sistemas de seguridad como IDS y SIEMS lo que permite obtener una visión completa de la postura de seguridad de la red. ● Está basado en protocolos el cual le permite realizar un análisis profundo de una variedad de protocolos de red lo que facilita la detección de anomalías y comportamiento inusuales específicos de cada protocolo analizado. ● Permite la monitorización del tráfico de la red en tiempo real, siendo este uno de los objetivos principales del presente TFM. ● Tiene un bajo impacto en el rendimiento de la red ya que se	consumo de CPU, memoria y espacio de disco.. ● La actualización de la plataforma puede requerir de esfuerzos y siempre debe validarse los cambios de versiones para mantener la compatibilidad con configuraciones existentes.

		ejecuta de una manera no intrusiva, observando el tráfico de red sin requerir modificar ni afectar el rendimiento de la red. ● Permite su uso para el análisis forense mediante la reconstrucción de eventos y la identificación de la causa de incidentes de seguridad.	

3. Fase de implementación

En este capítulo se realiza la descripción detallada de todo el proceso de implementación del prototipo del proyecto.

3.1. Herramientas seleccionadas para el desarrollo del prototipo

De acuerdo a la Figura 7 (Arquitectura de la plataforma en desarrollo) el prototipo tiene varios componentes las mismas que tienen sus propios requerimientos que permitirán realizar las demás actividades del proyecto.

El proyecto está basado en el uso de herramientas de código abierto dentro de las cuales el principal componente es el sistema operativo que usaremos para el despliegue de toda la infraestructura. Tenemos varias distribuciones de sistemas operativos de código abierto basados en Linux como CentOS, AlmaLinux, Rocky Linux, Oracle Linux, Debian y Ubuntu. En este punto se ha decidido usar para el proyecto la distribución AlmaLinux versión 8.

Debido a que no es parte de este proyecto hacer un estudio de las distintas distribuciones de Linux y sus ventajas y desventajas si es importante señalar las razones de dicha elección.

a) Al tener como uno de los objetivos la creación de una plataforma comunitaria y comercial basado en el proyecto se requiere mantener y gestionar todos los programas que se usarán en el proyecto y que no cuentan con compilaciones binarias como es el caso de Zeek.
b) Se implementará un Repositorio de Paquetes o RPM Package Manager[25] qué permitirá instalar, actualizar, desinstalar, verificar y solicitar programas de una forma simple y rápida, haciendo innecesario para los usuarios finales gastar tiempo y recursos para la compilación de los programas desde el código fuente de los mismos.

[25] https://es.wikipedia.org/wiki/RPM_Package_Manager

c) Control de las funcionalidades del producto que se vayan agregando en el futuro y facilidad para su actualización.

3.2. Aplicaciones de código abierto seleccionados

Se han elegido las siguientes aplicaciones de código abierto para los distintos componentes las cuales son:

a) **SENSORES**
   AlmaLinux, Zeek y Promtail.
b) **ALMACENAMIENTO DE LOGS Y DATOS**
   AlmaLinux, Grafana Loki y Clickhouse.
c) **BACKEND**
   AlamaLinux, Grafana, Nginx, Python y FastApi.
d) **FRONTEND**
   AlmaLinux, Nginx, Vue.

3.3. Configuración del entorno de trabajo

Se utilizará un entorno de trabajo virtual para lo cual se ha dispuesto del siguiente equipamiento de hardware y software.

a) Servidor INTEL NUC8v7PNH con:
   - 4 CPUs x Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz
   - 16 GB de RAM
   - Disco de 520 GB SSD
   - Disco de 1 TB SATA
   - 2 interfaces de red Gigabit
b) VMware ESXi-7.0U1c-17325551-standard (VMware, Inc.)

Este entorno ha sido creado con la finalidad de simular el entorno de producción final que se requerirá para la puesta en marcha del producto final.

En este servidor virtual se ha realizado la instalación de las máquinas virtuales necesarias para las demás fases del proyecto.

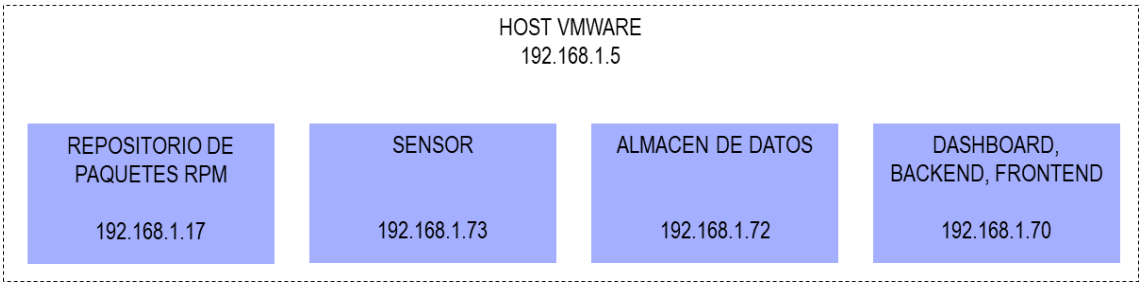


Figura 11: Distribución de las máquinas virtuales y direcciones ip asignadas

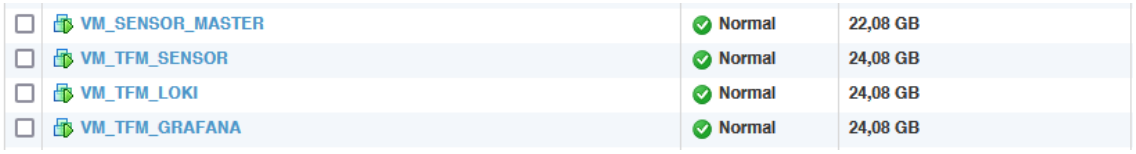


Figura 12: Listado de máquinas virtuales creadas en el host de VMWare

3.4. Especificaciones de la APIs a usar

Las APIs a usar en el proyecto son de dos tipos:

1. APIs de terceros
2. APIs de desarrollo propio

En este capítulo trataremos de las **APIs de terceros** y más adelante en el capítulo correspondiente al desarrollo del prototipo se tratará sobre las APIs de desarrollo propio.

La APIs a usar para la conexión a los datos almacenados corresponden a Grafana Loki que es el sistema de agregación de registros donde se almacenan los registros que son enviados por los Sensores, la conexión a los datos se realizará vía Grafana Loki HTTP API[26].

[26] https://grafana.com/docs/loki/latest/reference/api/

Grafana Loki HTTP API

Grafana Loki expone una API HTTP para enviar, consultar y recuperar datos de registro. Estas endpoints o direcciones de consumo de las APIs de Grafana Loki están compuesta por:

Endpoint Generales

- GET /ready
- GET /log_level
- GET /metrics
- GET /config
- GET /services
- GET /loki/api/v1/status/buildinfo
- GET /loki/api/v1/format_query

Endpoints de Consulta

- GET /loki/api/v1/query
- GET /loki/api/v1/query_range
- GET /loki/api/v1/labels
- GET /loki/api/v1/label/<name>/values
- GET /loki/api/v1/series
- GET /loki/api/v1/index/stats
- GET /loki/api/v1/index/volume
- GET /loki/api/v1/index/volume_range
- GET /loki/api/v1/tail

Existen muchos otros endpoints los cuales se pueden consultar en la documentación de Grafana Loki.

Las consultas a los endpoints de Grafana Loki se realizan mediante el Lenguaje de Consultas de LOG o LogQL[27] de Grafana. LogQL es el lenguaje de consulta inspirado en PromQL de Grafana Loki. Las consultas actúan como

[27] https://grafana.com/docs/loki/latest/query/

si fueran un grep[28] distribuido para agregar fuentes de registro. LogQL utiliza etiquetas y operadores para filtrar la información a solicitar.

Para nuestro proyecto usaremos Grafana para las consultas y creación de Paneles de control centralizados, así como Python para el desarrollo de las APIs internas de consultas en el Backend y Frontend que serán accesibles por los clientes.

A continuación se muestran algunas consultas a Grafana Loki HTTP API desde la línea de comandos:

a) Listamos las etiquetas que están almacenadas en Grafana Loki.

```
# curl -G -s  "http://localhost:3100/loki/api/v1/labels" | jq
{
  "status": "success",
  "data": [
    "filename",
    "host",
    "job"
  ]
}
```

b) Listamos los nombres de host de los Sensores que están enviando datos..

```
# curl -G -s  "http://localhost:3100/loki/api/v1/label/host/values" | jq
{
  "status": "success",
  "data": [
    "vm-tfm-sensor"
  ]
}
```

[28] https://es.wikipedia.org/wiki/Grep

4. Fase de instalación

En este capítulo se realiza la descripción de la instalación del software necesario en cada uno de las máquinas virtuales indicadas en el capítulo anterior.

4.1. Creación del repositorio de paquetes para la distribución de Zeek

Este paso es importante en el proyecto ya que permitirá el mantenimiento y distribución centralizada Zeek que es un paquete que si bien podemos instalar desde el código fuente no sería una buena elección para el presente proyecto puesto que los sensores deberán desplegarse constantemente en distintas redes y la forma de despliegue debe ser transparente para el cliente final.

Para este fin usaremos el paquete Mock[29] el cual es un administrador de entorno de compilación chroot 'simple' para crear RPMs, este a diferencia de rpmbuild[30] permite usar un entorno separado en el equipo para la compilación y creación de los paquetes RPM sin requerir instalar paquetes adicionales en equipo base.

Instalación de Mock

Los paquetes que se crearán se alojarán en la carpeta /var/lib/mock. El proceso de configuración e instalación del paquete Mock están contenidos en el Anexo 11.1.

Creación del repositorio de distribución

Finalizado el proceso de creación de los paquetes de distribución rpm para Zeek, el siguiente paso es crear el RPM Package Manager del proyecto. Para este fin se utilizará como medio de descarga un servidor web basado en Nginx cuyo proceso de instalación y configuración está contenido en el Anexo 11.2.

[29] https://rpm-software-management.github.io/mock/
[30] https://www.redhat.com/sysadmin/create-rpm-package

4.2. Instalación del almacén de datos (Grafana Loki y ClickHouse)

Este paso realizaremos la instalación del almacén de logs o datos para lo cual utilizaremos Grafana Loki para los Logs que envían los sensores y ClickHouse para los datos de análisis que se procesarán en el backend del proyecto.

Instalación de Grafana Loki

La instalación de Grafana Loki lo haremos usando el repositorio oficial de Grafana, para este fin se deberán seguir los pasos contenidos en el Anexo 11.3.

Configuración de Grafana Loki

Grafana Loki se encargará de almacenar la información enviada por los sensores vía Promtail, para este fin realizaremos sólo un cambio en el archivo de configuración de Grafana Loki **/etc/Grafana Loki/config.yml** para indicar donde se encontrará el directorio de almacenamiento de los datos, el mismo que para efectos del presente proyecto será **/data**, esto debido a que en forma predeterminada el directorio es /tmp.

El procedimiento para configurar se encuentra contenido en el Anexo 11.4.

Es posible ver el funcionamiento de Grafana Loki desde un navegador web usando por ejemplo los datos de acceso del servidor donde se ha instalado ingresando la siguiente URL http://192.168.1.73:3100/services

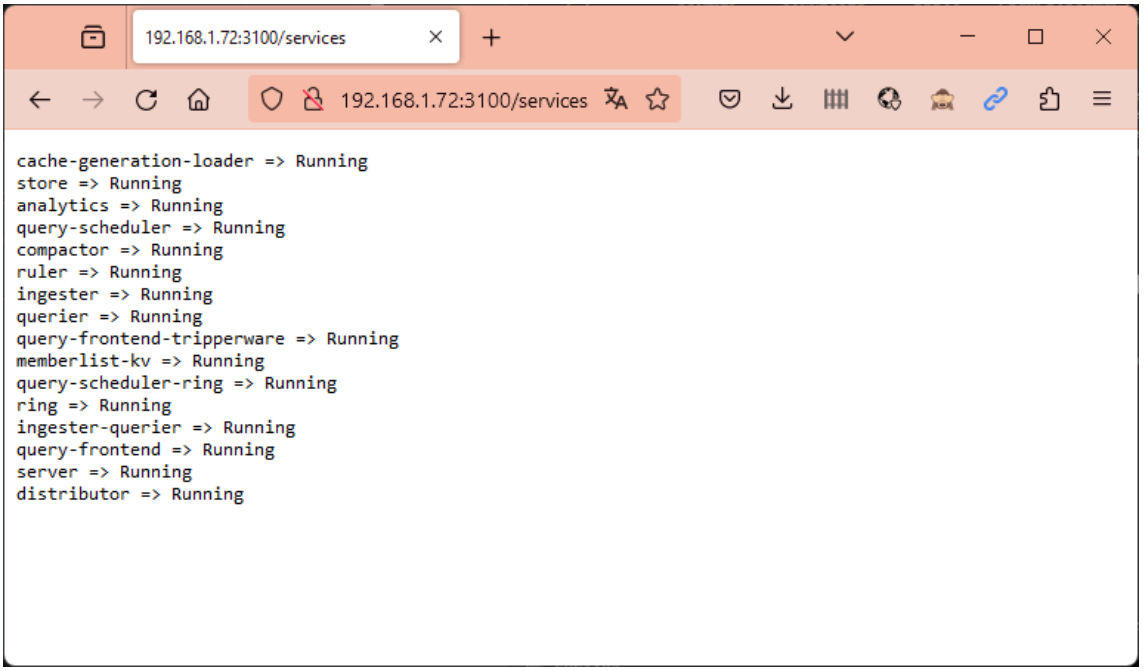


Figura 13: Vista vía web del funcionamiento de los servicios de Grafana Loki

Instalación de ClickHouse

La instalación de ClickHouse se ha realizado usando el repositorio oficial del fabricante, para este fin se deberán seguir los pasos contenidos en el Anexo 11.5:

Para verificar el correcto funcionamiento del servidor ClickHouse se usa el cliente clickhouse-client como se muestra en el siguiente cuadro.

```
# clickhouse-client
ClickHouse client version 23.10.5.20 (official build).
Connecting to localhost:9000 as user default.
Connected to ClickHouse server version 23.10.5 revision 54466.

Warnings:
 * Linux transparent hugepages are set to "always". Check
/sys/kernel/mm/transparent_hugepage/enabled
 * Linux threads max count is too low. Check /proc/sys/kernel/threads-max
 * Maximum number of threads is lower than 30000. There could be problems with
handling a lot of simultaneous queries.

vm-tfm-Grafana Loki.innovare.local :)
```

4.3. Instalación y Configuración del Sensor

Para el funcionamiento del sensor se requiere instalar dos paquetes:

a) Zeek : Encargado de crear los archivos de captura de paquetes en sus distintos registros.
b) Promtail : Servicio basado en agente encargado de enviar el contenido de los logs locales de Zeek a una instancia centralizada de Grafana Loki.

Instalación de Zeek

La instalación de Zeek es un procedimiento simple puesto que se usará el Repositorio de Paquetes RPM creado previamente, para este fin se deberán seguir los pasos contenidos en el Anexo 11.6

Configuración de Zeek

Para el presente proyecto se analizará el contenido del registro **conn.log**[31] el cual cuenta con los siguientes campos:

Campo	Descripción	Tipo
ts	Marca de tiempo que indica cuándo ocurrió la conexión.	time
uid	Identificador único de la conexión	string
id.orig_h	Dirección IP de origen	addr
id.orig_p	Puerto de origen	port
id.resp_h	Dirección IP de destino	addr
id.resp_p	Puerto de destino	port
proto	Protocolo utilizado en la conexión	enum
service	Servicio asociado con la conexión (si es conocido)	string
duration	Duración de la conexión	interval

[31] https://docs.zeek.org/en/v6.1.0/logs/conn.html

orig_bytes	Número de bytes transferidos desde el origen	count
resp_bytes	Número de bytes transferidos hacia el destino	count
conn_state	Estado de la conexión indicando además la secuencia de la conexión.	string
local_orig	Indicador si origen es local (verdadero o falso)	bool
local_resp	Indicador si la respuesta local (verdadero o falso)	bool
missed_bytes	Número de bytes que no pudieron ser capturados o registrados	count
history	Historial de eventos asociados a la conexión	string
orig_pkts	Número de paquetes enviados desde el origen	count
orig_ip_bytes	Número total de bytes enviados desde el origen	count
resp_pkts	Número de paquetes enviados desde el destino	count
orig_l2_addr	Dirección de la mac address del origen	string
resp_l2_addr	Dirección de la mac address del destino	string

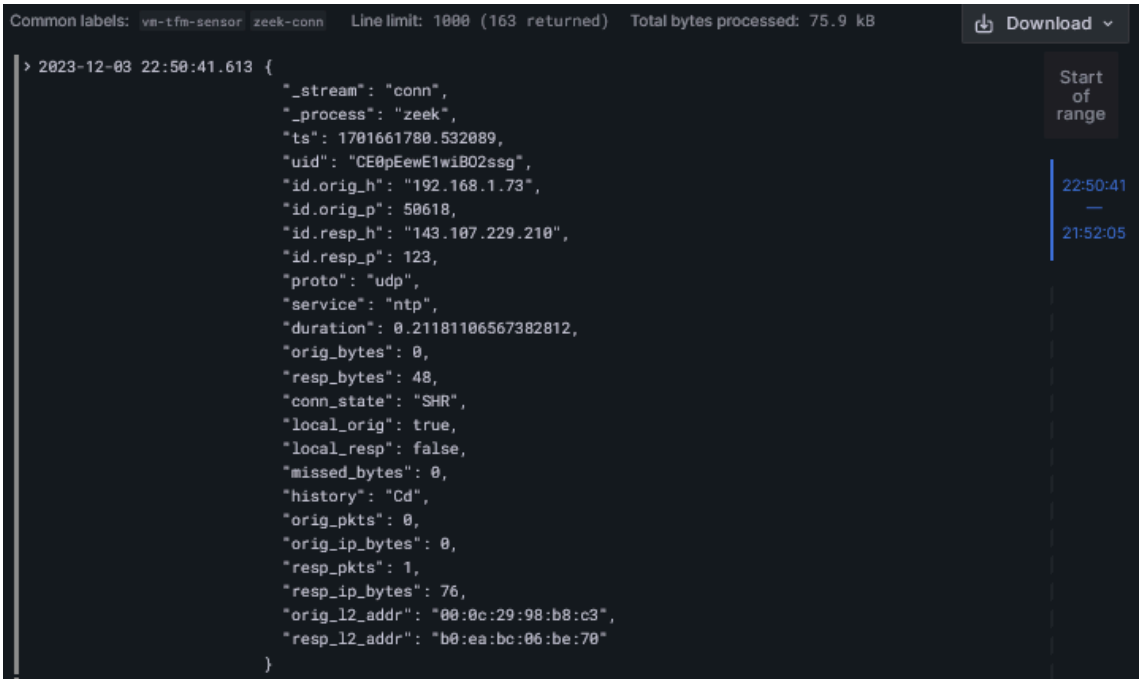


Figura 14: Ejemplo de un registro dentro de conn.log capturado por el sensor

A continuación realizaremos algunas modificaciones en los archivos de configuración de Zeek para que funcione de acuerdo a lo que requerimos en el proyecto. Los archivos son:

a) **/opt/zeek/etc/networks.cfg**

En este archivo añadiremos la lista de redes que analizaremos, es decir, acotaremos la detección sólo a las direcciones IP que son asignadas a los equipos de la empresa donde se instalará el sensor. Para efectos de las pruebas para el prototipo la red es la **192.168.1.0/24**.

```
192.168.1.0/24
```

b) **/opt/zeek/etc/node.cfg**

En este archivo añadiremos la interface de red qué se encuentra conectado a la red de la empresa. Para efectos de las pruebas para el prototipo es la interface **ens192**.

```
[zeek]
type=standalone
host=localhost
interface=ens192
```

c) **/opt/zeek/share/zeek/site/local.zeek**

En este archivo de configuración se configurarán dos aspectos que se requiere para el proyecto, uno es la de generar registros en formato JSON y otra detectar las direcciones mac de las conexiones con el fin de usarlos para el análisis y respuesta que buscamos. Para este fin se añadirá al final del archivo local.zeek las siguientes entradas:

```
# Output to JSON format
@load policy/tuning/json-logs.zeek

# LOG MAC Address
@load policy/protocols/conn/mac-logging.zeek
```

A continuación realizaremos el despliegue de las configuraciones realizadas e iniciaremos el monitoreo de la red.

```
# /opt/zeek/bin/zeekctl deploy
checking configurations ...
installing ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping zeek ...
starting ...
starting zeek ...
```

Para comprobar qué Zeek esté funcionando ejecutamos el comando “ps afx” y también verificamos con el comando “tail -f /opt/zeek/logs/current/conn.log” que se estén generando los registros de detección.

```
# /opt/zeek/bin/zeekctl deploy
11526 ?        S      0:00 /usr/bin/bash /opt/zeek/share/zeekctl/scripts/run-zeek -1 -i
ens192 -U .status -p zeekctl -p zeekct…
  11532 ?        Sl     0:00  \_ /opt/zeek/bin/zeek -i ens192 -U .status -p zeekctl -p
zeekctl-live -p standalone -p local -p ze…

# tail -f /opt/zeek/logs/current/conn.log
{"_stream":"conn","_process":"zeek","ts":1701667448.055519,"uid":"CkwXHl1URCNwl7
k5O6","id.orig_h":"192.168.1.49","id.orig_p":35301,"id.resp_h":"192.168.1.255","id.resp
_p":15600,"proto":"udp","conn_state":"S0","local_orig":true,"local_resp":true,"missed_b
ytes":0,"history":"D","orig_pkts":1,"orig_ip_bytes":63,"resp_pkts":0,"resp_ip_bytes":0,"or
ig_l2_addr":"24:4b:03:7e:01:9e","resp_l2_addr":"ff:ff:ff:ff:ff:ff"}
{"_stream":"conn","_process":"zeek","ts":1701667451.877726,"uid":"C5uB0g2ABHrWv
e3oye","id.orig_h":"192.168.1.73","id.orig_p":43488,"id.resp_h":"200.89.75.198","id.res
p_p":123,"proto":"udp","service":"ntp","duration":0.04592490196228027,"orig_bytes":0,
"resp_bytes":48,"conn_state":"SHR","local_orig":true,"local_resp":false,"missed_bytes":
0,"history":"Cd","orig_pkts":0,"orig_ip_bytes":0,"resp_pkts":1,"resp_ip_bytes":76,"orig_l
2_addr":"00:0c:29:98:b8:c3","resp_l2_addr":"b0:ea:bc:06:be:70"}
```

Instalación de Promtail

La instalación de Promtail lo haremos usando el repositorio oficial de Grafana, para este fin se deberán seguir los siguientes pasos contenidos en el Anexo 11.7:

Configuración de Promtail

Promtail realizará la lectura de los archivos de Log generados por Zeek y se encargará de enviar la información recogida hacia el servidor de datos (Grafana Loki) que se ha instalado previamente en el servidor 192.168.1.72. Ver detalles en el Anexo 11.7

Para comprobar el el funcionamiento de Promtail desde un navegador web usando se podrá usar los siguiente datos de acceso http://192.168.1.73:9080

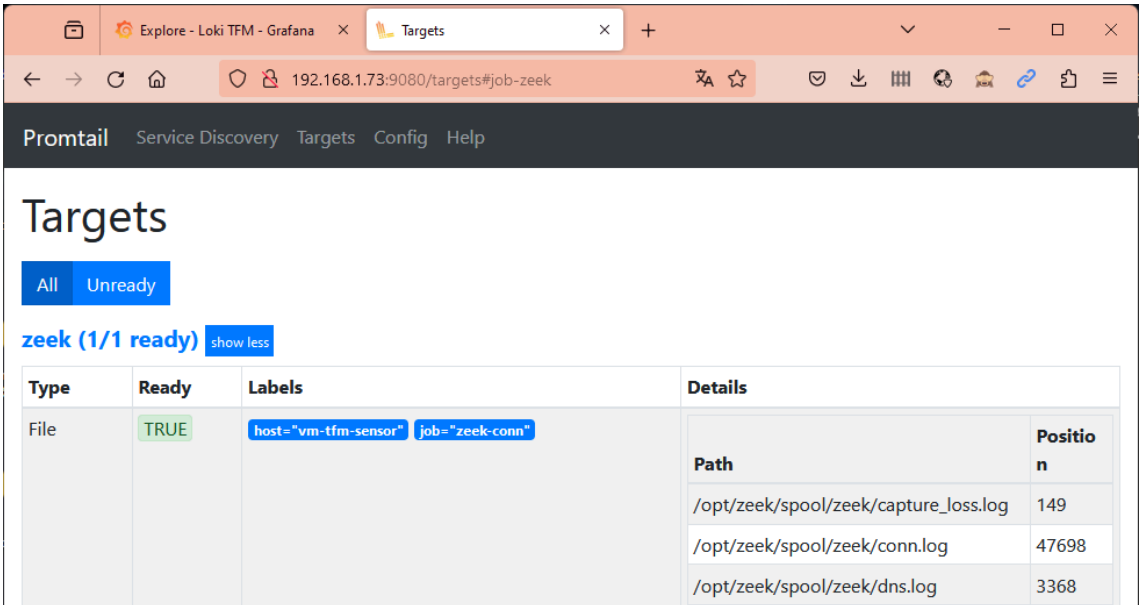


Figura 15: Vista vía web del funcionamiento de Promtail en el sensor

4.4. Instalación de Grafana para la creación de paneles de monitoreo, backend y frontend

Los registros enviados por los Sensores a Grafana Loki y ClickHouse necesitan ser visualizados para su posterior análisis y gestión, esta funcionalidad lo hace Grafana, cuya instalación se realiza con los siguientes pasos:

Instalación de Granfana

La instalación de Grafana lo haremos usando el repositorio oficial de Grafana Labs, para este fin se deberán seguir los siguientes pasos contenidos en el Anexo 11.8:

Configuración de Grafana

Para el presente proyecto sólo se trabajó con un servidor Grafana, sin embargo, en entornos de producción debe considerarse crear una red distribuida para brindar alta disponibilidad y tolerancia a fallos.

Para comprobar el funcionamiento de Grafana se ingresa mediante un navegador web al URL http://192.168.1.70:3000/ con el usuario “admin” y clave “admin”.

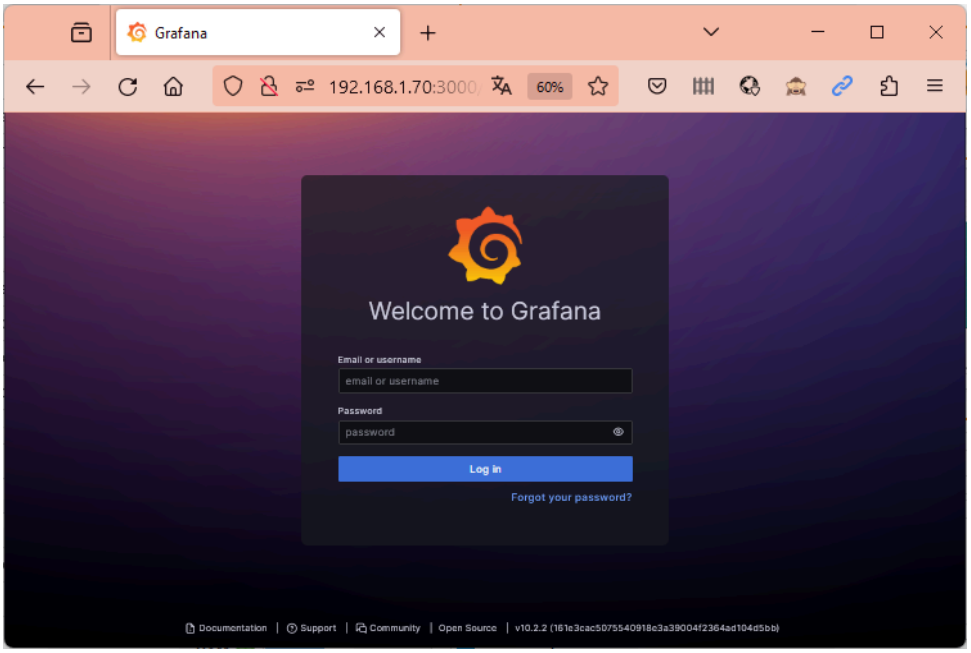


Figura 16: Acceso a Grafana desde un navegador web

Luego de haber ingresado a Grafana se deberá crear una conexión hacia la fuente de datos que en el proyecto es Grafana Loki, para lo cual debemos ir al menú principal, opción Connections / Data sources, seleccionar el botón Add new data source, buscar Grafana Loki y configurar la URL de conexión http://192.168.1.72:3100.

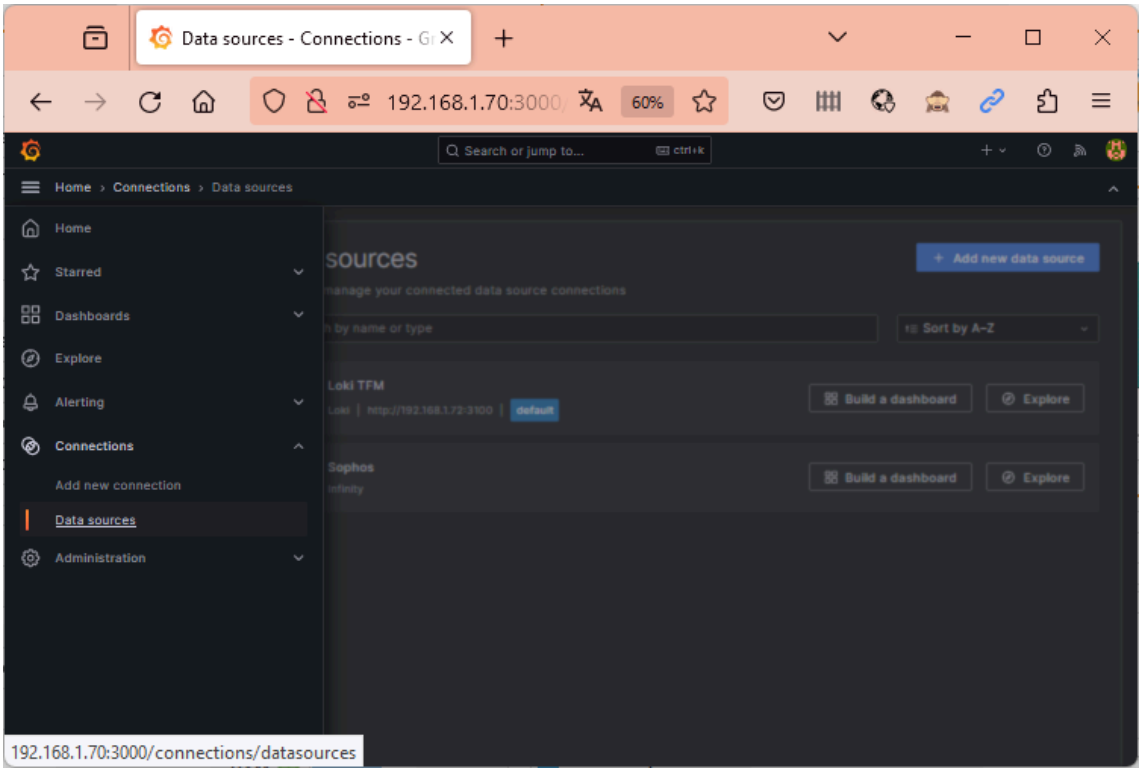


Figura 17: Creación de una fuente de datos en Grafana

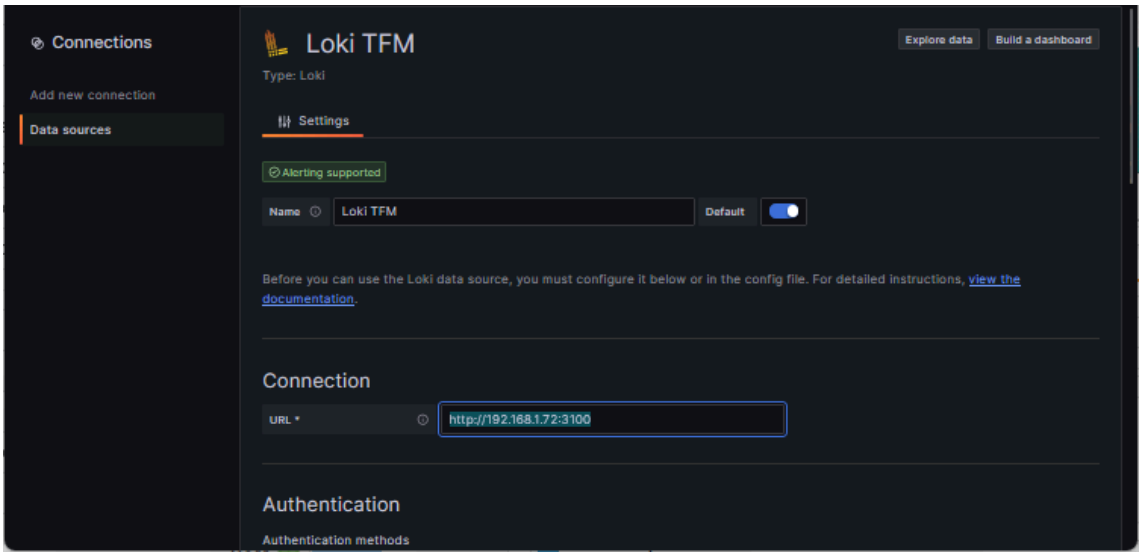


Figura 18: Configuración de la fuente de datos en Grafana

Luego de haber creado y configurado la fuente de datos se podrá realizar la exploración a Grafana Loki para validar si el Sensor está enviando la información detectada.

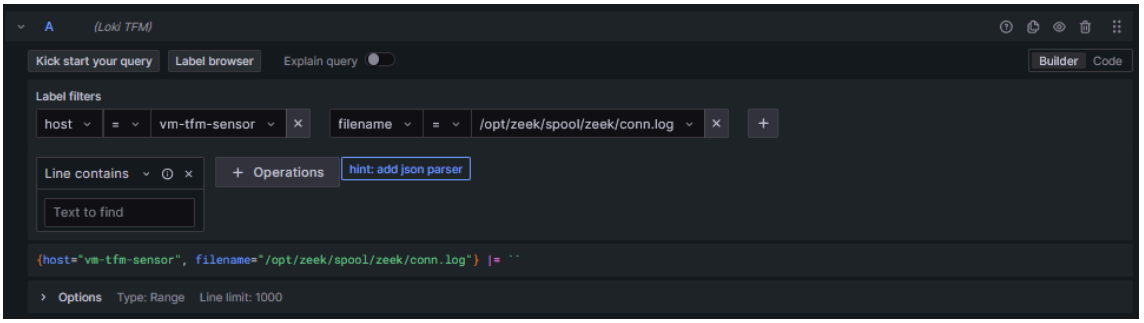


Figura 19: Creación del filtro para ver los registros de “conn.log” de Zeek

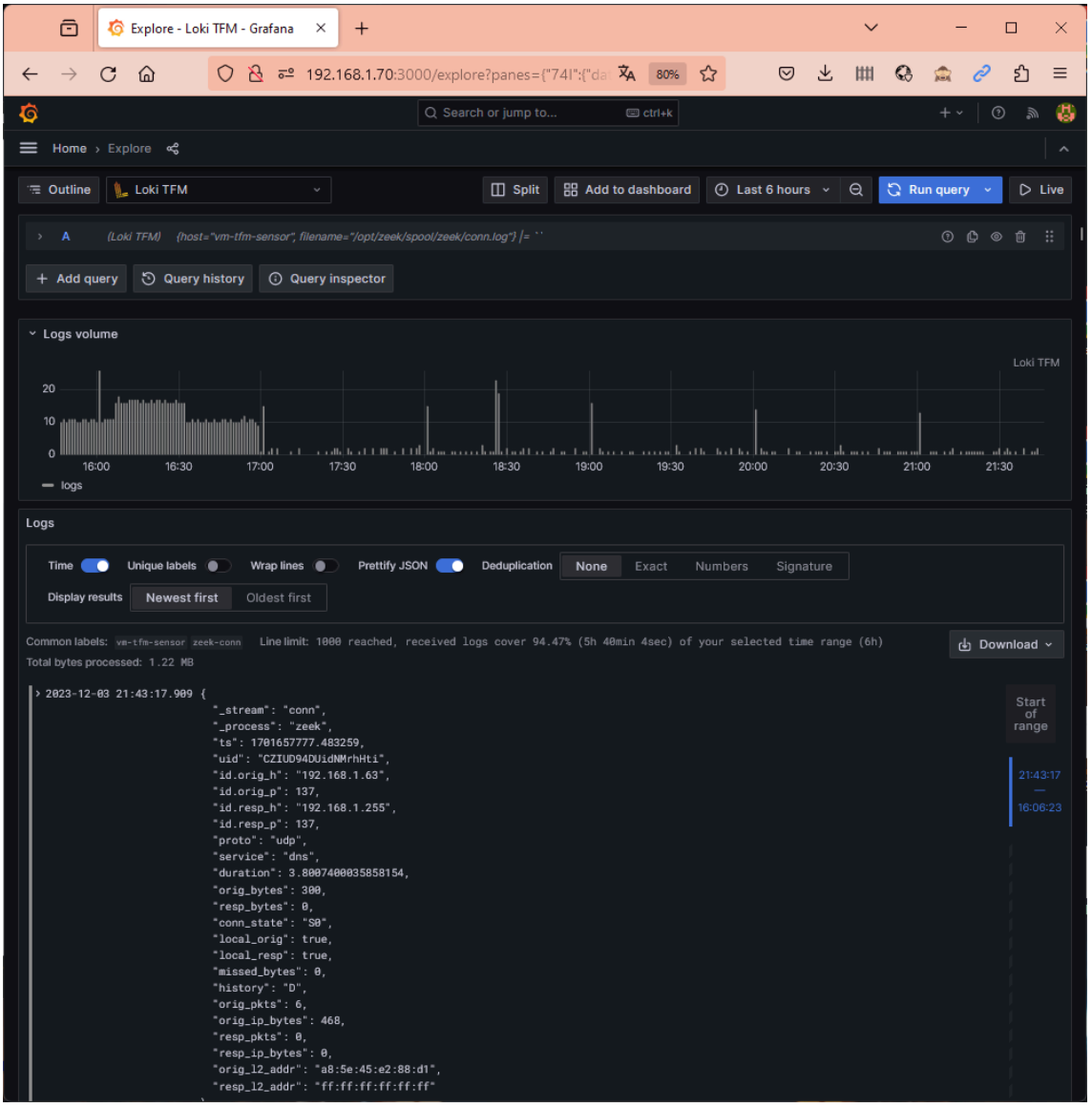


Figura 20: Vista del volúmen de registros y un registro JSON a detalle

5. Fase de desarrollo del prototipo

En este capítulo se realizará la descripción de las actividades que se han realizado para el desarrollo del primer prototipo de la plataforma.

El primer prototipo no será completamente funcional puesto que para el desarrollo de la plataforma se requiere realizar un proyecto propio que deberá involucrar el diseño final, el costeo, tiempos de desarrollo, metodología a usar, etc.

5.1. Prototipo de Consola de cliente final

Desde la consola de cliente final del proyecto el usuario final podrá hacer seguimiento de las detección y comprobaciones de seguridad qué están realizando los sensores ubicados en distintos puntos de la red.

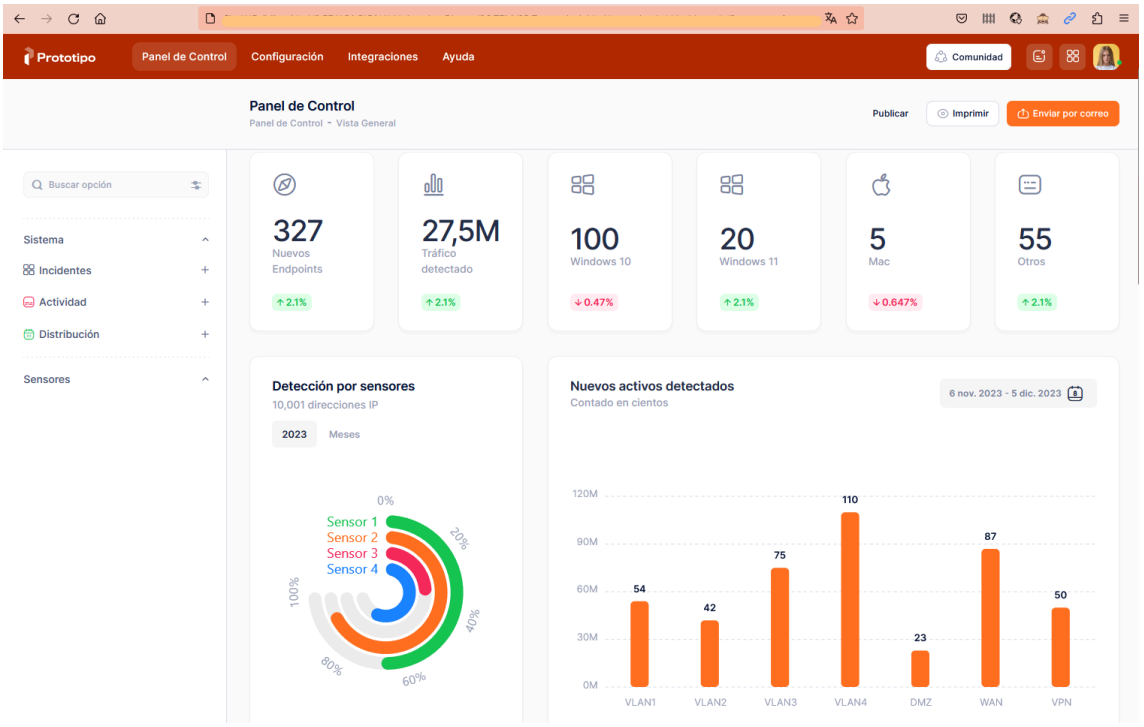


Figura 21: Vista del Panel de Control de cliente final

6. Conclusiones y trabajos futuros

A continuación se describen las conclusiones finales del TFM luego de haber obtenido los resultados de la implementación del prototipo.

- Se ha logrado cumplir con el objetivo principal señalado en el proyecto que es la de reconocer los equipos de punto final que se conectan a la red, esto se ha logrado gracias a Zeek.

- Durante la implementación y pruebas se ha comprobado que con Zeek no solamente podemos detectar la dirección IP de los puntos finales que se conectan a la red sino también de la dirección MAC del equipo lo que permitirá añadir funcionalidades adicionales usando este dato que es único para realizar otros análisis se seguridad.

- Se ha comprobado qué las herramientas usadas para la recolección de log y almacenamiento como son Protmail, Loki y Grafana permitirán realizar un ahorro en el consumo de recursos de procesamiento en la plataform requerida puesto que a diferencia de la pila ELK aunque no fue analizada en el proyecto es conocido por requerir bastantes recursos por lo que la elección de las herramientas de código abierto han sido desde nuestro punto de vista adecuadas para el proyecto ya que nos ha permitido avanzar sin tener que realizar cambios en las herramientas seleccionadas.

- Tanto OpenSearch como ClickHouse van a permitir realizar el adecuado dimensionamiento de la plataforma central para el tratamiento de los datos ya que en las pruebas realizadas se ha comprobado que se generan miles de registros y para una rápida evaluación de la seguridad se requiere también respuestas rápidas a las consultas de los datos que se recopilan ya sea para generar respuestas automatizadas como para realizar evaluaciones de seguridad y el análisis de los datos para la

creación de otras funcionalidades que pueden no haber sido previstas en el presente trabajo.

Así también se han logrado cumplir con los objetivos trazados inicialmente y ha permitido visualizar y encontrar otros usos que pueden darse a la plataforma propuesta ya sea agregando nuevas funcionalidades o añadiendo más herramientas de código abierto como por ejemplo para realizar la evaluación de vulnerabilidades a los activos encontrados con el fin de mejorar aún más la seguridad en las empresas.

6.1. Análisis crítico de la planificación y metodología utilizada

Con respecto a la planificación esta se ha ejecutado sin contratiempos, sin embargo, para el desarrollo de una versión Alpha de la plataforma propuesta se requiere más tiempo para el análisis y desarrollo del software tanto para los sensores como para las herramientas de administración central y administrador de usuario final las cuales a pesar de no ser parte del alcance del presente TFM podría tomarse en cuenta para la elaboración de un nuevo proyecto de desarrollo y lanzamiento de una primera versión comercial usando como insumo el presente trabajo.

También debemos indicar que la gracias a la metodología de desarrollo del TFM establecido por la UOC se ha podido planificar al inicio del trabajo un cronograma completo de desarrollo así como realizar los ajustes necesarios en la medida de las necesidades de cada entregable lo que nos ha permitido seguir la línea de tiempo trazada e ir cumpliendo a tiempo cada uno de los hitos planificados.

Respecto a los impactos de sostenibilidad, ético-social y de diversidad estás se han logrado cumplir puesto que se han identificados impactos positivos a los cuales el presente trabajo debe llegar y aplicarse.

6.2. Trabajo futuros

Al haberse conseguido los objetivos planteados en el presente trabajo de TFM realizamos una revisión de los trabajos futuros qué pueden realizarse usando como insumo la información y documentación elaborada.

- Elaborar procedimientos automatizados para la creación de máquinas virtuales basadas en VMWare, ISO u otras para facilitar la distribución de los sensores en modo de paquetes ya pre-configurados puesto que el objetivo del trabajo es el desarrollo de una plataforma comercial qué pueda ser vendida como una nueva herramienta para la mejorar de la seguridad en las empresas.

- Investigar si los sensores pueden realizar las mismas funciones usando contenedores Docker lo que también permitiría una adecuada distribución y desarrollo centralizado de los mismos.

- Desarrollar la plataforma de administración centralizada qué permita la creación de instancias para el manejo de clientes donde se van a distribuir los sensores de tal modo qué puedan generarse controles de uso ya sea para pruebas de concepto o para la venta de licencias de uso.

- Desarrollar la interfaz de usuario final para la administración de la plataforma y la integración con los diferentes fabricantes de herramientas de seguridad de punto final como productos de antivirus, de punto final, de detección y respuesta de incidentes u otros.

- Desarrollar los programas para la creación, mantenimiento y mejora de las políticas de seguridad predefinidas qué se deberán aplicar en los sensores mediante la comunicación vía API con las diferentes herramientas de seguridad.

- Evaluar los diferentes proveedores SaaS para la instalación y operación de la plataforma centralizada.

7. Glosario

BYOD[32]: Bring your own device («trae tu propio dispositivo» en inglés), abreviado BYOD, es una política empresarial consistente en que los empleados lleven sus propios dispositivos personales (portátiles, tabletas, móviles…) a su lugar de trabajo para tener acceso a recursos de la empresa tales como correos electrónicos, bases de datos y archivos en servidores así como datos y aplicaciones personales…

Sensores de Red[33] : Una red de sensores (del inglés sensor network) es una red de ordenadores pequeñísimos («nodos»), equipados con sensores, que colaboran en una tarea o más tareas comunes.

API[34] : Interfaz de Programación de Aplicaciones o API es un conjunto de reglas definidas que permiten que diferentes aplicaciones se comuniquen entre sí.

Modo Promiscuo[35] : En informática, el modo promiscuo es aquel en el que una computadora conectada a una red compartida, tanto la basada en cable de cobre como la basada en tecnología inalámbrica, captura todo el tráfico que circula por ella.

Port-Mirror[36] : Port Mirroring (puerto espejo) se utiliza en un switch de red o en un router para enviar una copia de los paquetes de red vistos en los puertos especificados (puertos de origen) a otros puertos especificados (puertos de destino

[32] https://es.wikipedia.org/wiki/Bring_your_own_device
[33] https://es.wikipedia.org/wiki/Red_de_sensores
[34] https://www.ibm.com/mx-es/topics/api
[35] https://es.wikipedia.org/wiki/Modo_promiscuo
[36] https://community.fs.com/es/blog/port-mirroring-explained-basis-configuration-and-fa-qs.html

SPAN[37] : El Analizador de puertos conmutados (SPAN), que también se denomina duplicación de puertos o duplicación de tráfico, le permite monitorear el tráfico de red que entra o sale de un conjunto de puertos. Luego puede pasar este tráfico a un puerto de destino en el mismo enrutador.

NAC[38] : Network Access Control o Control de Acceso a la red.

OSI[39] : Open Systems Interconnection es el modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), conocido como “modelo OSI”.

PCAP[40] : El .pcap extensión de archivo se asocia principalmente con Wireshark; un programa usado para el análisis de redes. .pcap archivos son archivos de datos creados mediante el programa y que contiene el paquete de datos de una red.

JSON[41] : Es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.

---

37 https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5000/interfaces/710x/b-interfaces-hardware-component-cg-ncs5000-710x/configuring-traffic-mirroring.html

38 https://www.cisco.com/en/US/solutions/ns340/ns394/ns171/ns466/ns617/net_design_guidance0900aecd80417226.pdf

39 https://es.wikipedia.org/wiki/Modelo_OSI

40 https://es.wikipedia.org/wiki/Pcap_(interfaz)

41 https://es.wikipedia.org/wiki/JSON

8. Bibliografía

1.Rose S, Borchert O, Mitchell S, Connelly S. Zero Trust Architecture. 2020 Aug 11 [cited 2023 Nov 5]; Available from: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

2.Spitzner L. Informe SANS 2022: Avanzando hacia un estado de zero trust [Internet]. SANS Institute. 2022 [cited 2023 Nov 5]. Available from: https://www.sans.org/white-papers/sans-2022-report-moving-to-a-state-of-zero-trust-spanish/

3.Bozzano JGF. Elaboración de un plan de aplicación de tecnologías SASE y Zero Trust [Internet]. OpenAccess UOC. 2023. Available from: https://openaccess.uoc.edu/handle/10609/147291

4.Luis GBJ. Análisis del modelo de seguridad Zero Trust y las consideraciones generales aplicables a cualquier organización pública en Colombia. [Internet]. Repository UNAD. 2022. Available from: https://repository.unad.edu.co/handle/10596/48929

5.García NAL. Diseño e implementación de una red LAN y WLAN con sistema de control de acceso mediante servidores AAA [Internet]. Repositorio de Tesis PUCP. 2012. Available from: https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/1445

6.Page B. Informe técnico sobre la seguridad Verify Zero Trust [Internet]. Cisco. 2022. Available from: https://www.cisco.com/c/es_mx/support/docs/security-vpn/security-vpn/218443-verify-zero-trust-security-whitepaper.html

7.Rose S, Borchert O, Mitchell S, Connelly S. SP 800-207, Zero Trust Architecture [Internet]. CSRC. 2020. Available from: https://csrc.nist.gov/pubs/sp/800/207/final

8.Sharatan. Network Admission Control (NAC) Framework Deployment Guide

[Internet]. Cisco. 2006 [cited 2023 Nov 6]. Available from: https://www.cisco.com/en/US/solutions/ns340/ns394/ns171/ns466/ns617/net_design_guidance0900aecd80417226.pdf

9.Bromiley M. Informe SANS 2022: Avanzando hacia un estado de zero trust. Sans [Internet]. 2022 Aug [cited 2023 Nov 5];4–5. Available from: https://sansorg.egnyte.com/dl/5BFn7d3UWq

10.Security I. Las brechas de seguridad de datos cuestan a las empresas una media de 4,24 millones de dólares por incidente, según el informe Cost of Data Breach de IBM [Internet]. IBM España News Room. 2021 [cited 2023 Nov 5]. Available from: https://es.newsroom.ibm.com/announcements?item=122679

11.Security I. Informe sobre el coste de una vulneración de datos de 2023. IBM Security. 2023 Jul;

9. Anexos

9.1. Instalación de Mock

Instalación del paquete Mock

Para este procesos se debe configurar el acceso a los paquetes adicionales para Linux Empresarial (o EPEL[42]).

a) Instalación de los paquetes EPEL y Mock

```
# yum install epel-release
# yum install mock
```

b) Asignación de permisos de usuario para el usuario "mock"

```
# adduser mock -g mock
# usermod -a -G mock mock
```

Descarga del archivo de especificaciones rpm y compilación de Zeek

Zeek mantiene un archivo de especificaciones rpm para que cualquier persona compile sus propios paquetes RPM.

a) Descargamos el paquete fuente de última versión de Zeek desde: https://download.opensuse.org/repositories/security:/zeek/CentOS_7/src/

```
# wget
https://download.opensuse.org/repositories/security:/zeek/CentOS_7/src/zeek-6.1.0-1.1.src.rpm
Connecting to downloadcontentcdn.opensuse.org
(downloadcontentcdn.opensuse.org)|151.101.1.91|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 60994401 (58M) [application/x-redhat-package-manager]
Saving to: 'zeek-6.1.0-1.1.src.rpm'

zeek-6.1.0-1.1.src.rpm          100%[==================================>]
```

[42] https://fedoraproject.org/wiki/EPEL/es

```
58.17M   512KB/s  in 2m 1s

2023-12-03 19:35:59 (492 KB/s) - ‘zeek-6.1.0-1.1.src.rpm’ saved [60994401/60994401]
```

b) Inicialización del entorno de compilación, este paso es recomendable para que las siguientes compilaciones sean más rápidas.

```
# mock -r almalinux-8-x86_64 --init
INFO: mock.py version 5.2 starting (python version = 3.6.8, NVR = mock-5.2-1.el8),
args: /usr/libexec/mock/mock -r almalinux-8-x86_64 --init
Start(bootstrap): init plugins
INFO: selinux enabled
Finish(bootstrap): init plugins
Start: init plugins
INFO: selinux enabled
Finish: init plugins
INFO: Signal handler active
Start: run
Start: clean chroot
Finish: clean chroot
Mock Version: 5.2
INFO: Mock Version: 5.2
Start(bootstrap): chroot init
INFO: calling preinit hooks
INFO: enabled root cache
INFO: enabled package manager cache
Start(bootstrap): cleaning package manager metadata
Finish(bootstrap): cleaning package manager metadata
INFO: Guessed host environment type: unknown
WARNING: Can't work with Podman, falling back to bootstrap installation:
'/usr/bin/podman' not installed
INFO: Package manager dnf detected and used (fallback)
Start(bootstrap): installing dnf tooling
No matches found for the following disable plugin patterns: local, spacewalk,
versionlock
AlmaLinux 8 - BaseOS                          547 kB/s | 3.0 MB     00:05
AlmaLinux 8 - AppStream                       1.2 MB/s |  11 MB     00:08
AlmaLinux 8 - PowerTools                      956 kB/s | 3.0 MB     00:03
AlmaLinux 8 - Extras                           16 kB/s |  20 kB     00:01
Dependencies resolved.
…
…
```

```
zlib-1.2.11-25.el8.x86_64
zstd-1.4.4-1.el8.x86_64
Finish: run
```

c) Proceso de compilación de los paquetes RPM de Zeek.

```
# mock -r almalinux-8-x86_64 --rebuild zeek-6.1.0-1.1.src.rpm
INFO: mock.py version 5.2 starting (python version = 3.6.8, NVR = mock-5.2-1.el8),
args: /usr/libexec/mock/mock -r almalinux-8-x86_64 --rebuild zeek-6.1.0-1.1.src.rpm
Start(bootstrap): init plugins
INFO: selinux enabled
Finish(bootstrap): init plugins
Start: init plugins
INFO: selinux enabled
Finish: init plugins
INFO: Signal handler active
Start: run
INFO: Start(zeek-6.1.0-1.1.src.rpm)  Config(almalinux-8-x86_64)
Start: clean chroot
Finish: clean chroot
Mock Version: 5.2
INFO: Mock Version: 5.2
Start(bootstrap): chroot init
INFO: calling preinit hooks
INFO: enabled root cache
INFO: enabled package manager cache
…
…
make[3]: Entering directory '/builddir/build/BUILD/zeek-6.1.0/build'
[  1%] Building CXX object auxil/binpac/src/CMakeFiles/binpac.dir/pac_parse.cc.o
[  1%] Building CXX object auxil/binpac/src/CMakeFiles/binpac.dir/__/pac_scan.cc.o
[  1%] Building CXX object auxil/binpac/src/CMakeFiles/binpac.dir/pac_action.cc.o
[  1%] Building CXX object auxil/binpac/src/CMakeFiles/binpac.dir/pac_analyzer.cc.o
…
…
[100%] Building CXX object
auxil/spicy/spicy/runtime/CMakeFiles/spicy-rt-debug-objects.dir/src/util.cc.o
[100%] Building CXX object
auxil/spicy/spicy/runtime/CMakeFiles/spicy-rt-debug-objects.dir/src/zlib.cc.o
[100%] Building C object
auxil/spicy/spicy/runtime/CMakeFiles/spicy-rt-debug-objects.dir/__/__/3rdparty/libb64/sr
c/cdecode.c.o
[100%] Building C object
```

```
auxil/spicy/spicy/runtime/CMakeFiles/spicy-rt-debug-objects.dir/__/__/3rdparty/libb64/src/cencode.c.o
make[3]: Leaving directory '/builddir/build/BUILD/zeek-6.1.0/build'
…
…
-- Installing: /builddir/build/BUILDROOT/zeek-6.1.0-1.1.x86_64/opt/zeek/bin/zeek-archiver
-- Up-to-date: /builddir/build/BUILDROOT/zeek-6.1.0-1.1.x86_64/opt/zeek/bin
-- Installing: /builddir/build/BUILDROOT/zeek-6.1.0-1.1.x86_64/opt/zeek/bin/zeek-client
-- Up-to-date: /builddir/build/BUILDROOT/zeek-6.1.0-1.1.x86_64/opt/zeek/lib/zeek/python
-- Installing: /builddir/build/BUILDROOT/zeek-6.1.0-1.1.x86_64/opt/zeek/lib/zeek/python/zeekclient
-- Installing: /builddir/build/BUILDROOT/zeek-6.1.0-1.1.x86_64/opt/zeek/lib/zeek/python/zeekclient/controller.py
-- Installing: /builddir/build/BUILDROOT/zeek-6.1.0-1.1.x86_64/opt/zeek/lib/zeek/python/zeekclient/config.py
…
…
Checking for unpackaged file(s): /usr/lib/rpm/check-files /builddir/build/BUILDROOT/zeek-6.1.0-1.1.x86_64
Wrote: /builddir/build/RPMS/zeek-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-core-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-devel-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-spicy-devel-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/libbroker-devel-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeekctl-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-zkg-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-btest-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-btest-data-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-client-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-debugsource-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-debuginfo-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-core-debuginfo-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-devel-debuginfo-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeek-spicy-devel-debuginfo-6.1.0-1.1.x86_64.rpm
Wrote: /builddir/build/RPMS/zeekctl-debuginfo-6.1.0-1.1.x86_64.rpm
Finish: rpmbuild zeek-6.1.0-1.1.src.rpm
Finish: build phase for zeek-6.1.0-1.1.src.rpm
INFO: Done(zeek-6.1.0-1.1.src.rpm) Config(almalinux-8-x86_64) 91 minutes 24 seconds
```

```
INFO: Results and/or logs in: /var/lib/mock/almalinux-8-x86_64/result
Finish: run
```

Dependiendo del tipo de equipo donde se realiza la compilación este proceso puede demorar de varios minutos a algunas horas, sin embargo, este proceso ahorrará bastante tiempo al momento de realizar la instalación de los sensores.

9.2. Instalación y configuración del RPM Package Manager

Para este procesos se deben seguir los siguientes pasos:

a) Instalación del servidor web Nginx.

```
# yum install nginx
```

b) Configuramos el directorio del servidor nginx como un repositorio de Paquetes RPM con el comando **createrepo**.

```
# createrepo /usr/share/nginx/html/
Directory walk started
Directory walk done - 17 packages
Temporary output repo path: /usr/share/nginx/html/.repodata/
Preparing sqlite DBs
Pool started (with 5 workers)
Pool finished
```

Comprobamos el funcionamiento ingresando desde un navegador a la dirección http://192.168.1.17

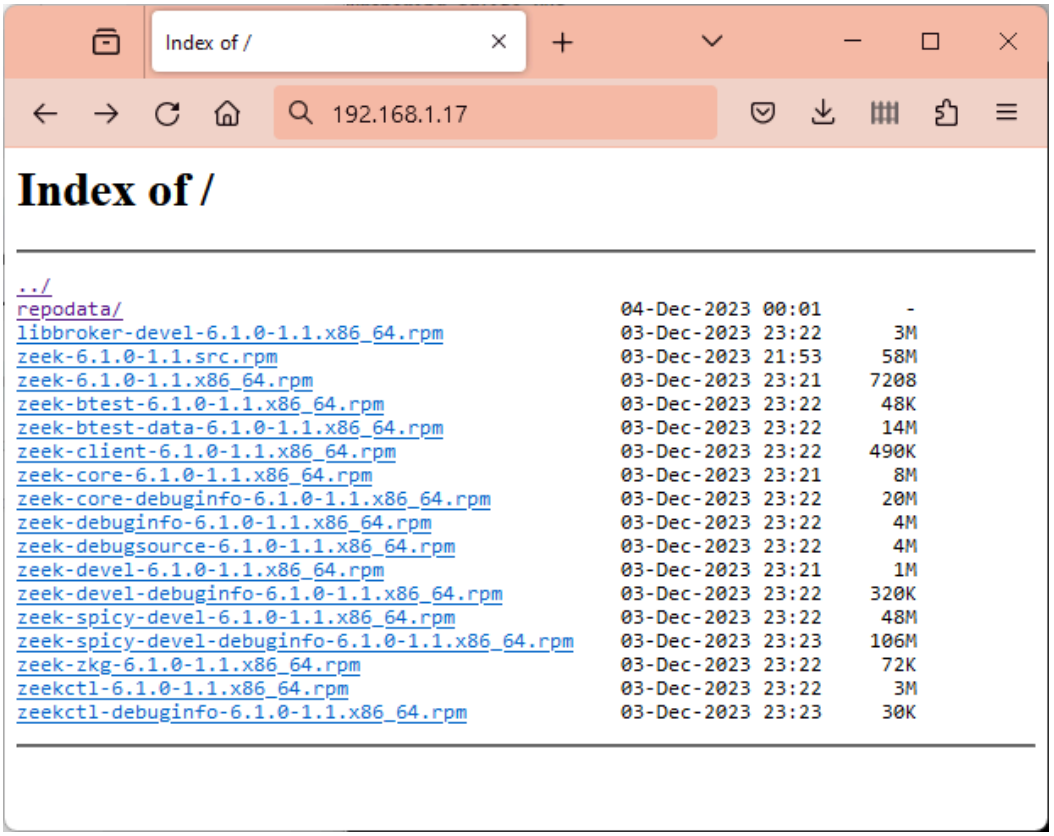


Figura 22: Vista del RPM Package Manager para el proyecto

Finalizado estos pasos la instalación y actualización de Zeek en los sensores será fácil ya que no se tendrá que compilar desde las fuentes cada vez qué tenga que instalar y/o distribuir un sensor.

9.3. Instalación de Grafana Loki

a) Creamos el archivo grafana.repo dentro de /etc/yum.repos.d/ con el siguiente contenido:

```
[grafana]
name=grafana
baseurl=https://rpm.grafana.com
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://rpm.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

b) Instalamos **Grafana Loki** usando el comando yum.

```
# yum install Grafana Loki
```

a) Verificamos la instalación ejecutando los siguientes comandos:

```
# Grafana Loki --version
Grafana Loki, version 2.9.2 (branch: HEAD, revision: a17308db6)
  build user:       root@eee92863de73
  build date:       2023-10-16T14:20:36Z
  go version:       go1.21.3
  platform:         linux/amd64
  tags:             netgo
```

9.4. Configuración de Grafana Loki

```
auth_enabled: false

server:
  http_listen_port: 3100
  grpc_listen_port: 9096

common:
  instance_addr: 127.0.0.1
  path_prefix: /data/Grafana Loki
  storage:
    filesystem:
      chunks_directory: /data/Grafana Loki/chunks
      rules_directory: /data/Grafana Loki/rules
  replication_factor: 1
  ring:
    kvstore:
      store: inmemory

query_range:
  results_cache:
    cache:
      embedded_cache:
        enabled: true
        max_size_mb: 100
```

```
schema_config:
  configs:
    - from: 2020-10-24
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h

ruler:
  alertmanager_url: http://localhost:9093
```

Luego de realizar los cambios se debe proceder a iniciar el servicio con el comando:

```
# systemctl start Grafana Loki

# Podemos visualizar el estado del funcionamiento de Grafana Loki con el comando:
#  systemctl status Grafana Loki
● Grafana Loki.service - Grafana Loki service
   Loaded: loaded (/etc/systemd/system/Grafana Loki.service; enabled; vendor preset:
disabled)
   Active: active (running) since Sat 2023-12-02 18:19:57 EST; 1 day 6h ago
 Main PID: 895 (Grafana Loki)
    Tasks: 8 (limit: 23142)
   Memory: 160.4M
   CGroup: /system.slice/Grafana Loki.service
           └─895 /usr/bin/Grafana Loki -config.file /etc/Grafana Loki/config.yml

dic 04 00:50:36 vm-tfm-Grafana Loki.innovare.local Grafana Loki[895]: level=info
ts=2023-12-04T05:50:36.456346987Z caller=table.go:318 msg="han>
…
…
…
…
```

9.5. Instalación y configuración del servidor ClickHouse

a) Añadimos el repositorio oficial con los siguientes comandos:

```
# yum install -y yum-utils
# yum-config-manager --add-repo
https://packages.clickhouse.com/rpm/clickhouse.repo
Agregando repositorio de: https://packages.clickhouse.com/rpm/clickhouse.repo
```

b) Instalamos el servidor y cliente de ClickHouse con los siguientes comandos:

```
# yum install -y clickhouse-server clickhouse-client
ClickHouse - Stable Repository                    1.2 kB/s | 833  B     00:00
ClickHouse - Stable Repository                    9.3 kB/s | 5.7 kB     00:00
Importando llave GPG 0x2B48D754:
 ID usuario: "ClickHouse Inc. Repositories Key <packages@clickhouse.com>"
 Huella    : 3A9E A119 3A97 B548 BE14 57D4 8919 F6BD 2B48 D754
 Desde     : https://packages.clickhouse.com/rpm/stable/repodata/repomd.xml.key
ClickHouse - Stable Repository                    160 kB/s | 251 kB     00:01
Dependencias resueltas.
…
…
Creating pid directory /var/run/clickhouse-server.
 chown -R clickhouse:clickhouse '/var/log/clickhouse-server/'
 chown -R clickhouse:clickhouse '/var/run/clickhouse-server'
 chown  clickhouse:clickhouse '/var/lib/clickhouse/'
 groupadd -r clickhouse-bridge
 useradd -r --shell /bin/false --home-dir /nonexistent -g clickhouse-bridge
clickhouse-bridge
 chown -R clickhouse-bridge:clickhouse-bridge '/usr/bin/clickhouse-odbc-bridge'
 chown -R clickhouse-bridge:clickhouse-bridge '/usr/bin/clickhouse-library-bridge'
Password for default user is empty string. See /etc/clickhouse-server/users.xml and
/etc/clickhouse-server/users.d to change it.
Setting capabilities for clickhouse binary. This is optional.
 chown -R clickhouse:clickhouse '/etc/clickhouse-server'

ClickHouse has been successfully installed.
Start clickhouse-server with:
 sudo clickhouse start
```

```
Start clickhouse-client with:
 clickhouse-client
…
…
Instalado:
  clickhouse-client-23.10.5.20-1.x86_64
clickhouse-common-static-23.10.5.20-1.x86_64
clickhouse-server-23.10.5.20-1.x86_64

¡Listo!
```

c) Configuramos el inicio de ClickHouse server en forma automática con los siguientes comandos:

```
# systemctl enable clickhouse-server
# systemctl start clickhouse-server
# systemctl status clickhouse-server
● clickhouse-server.service - ClickHouse Server (analytic DBMS for big data)
   Loaded: loaded (/usr/lib/systemd/system/clickhouse-server.service; enabled; vendor
preset: disabled)
   Active: active (running) since Mon 2023-12-04 01:09:43 EST; 4s ago
 Main PID: 14045 (clickhouse-serv)
    Tasks: 271 (limit: 23142)
   Memory: 113.8M
   CGroup: /system.slice/clickhouse-server.service
           ├─14044 clickhouse-watchdog --config=/etc/clickhouse-server/config.xml
--pid-file=/run/clickhouse-server/clickhouse->
           └─14045 /usr/bin/clickhouse-server --config=/etc/clickhouse-server/config.xml
--pid-file=/run/clickhouse-server/clic>

dic 04 01:09:42 vm-tfm-Grafana Loki.innovare.local systemd[1]: Starting ClickHouse
Server (analytic DBMS for big data)...
…
…
dic 04 01:09:42 vm-tfm-Grafana Loki.innovare.local clickhouse-server[14045]: Saved
preprocessed configuration to '/var/lib/clickhouse/p>
dic 04 01:09:43 vm-tfm-Grafana Loki.innovare.local systemd[1]: Started ClickHouse
Server (analytic DBMS for big data).
```

9.6. Instalación de Zeek

a) Creamos el archivo zeek-local.repo dentro de /etc/yum.repos.d/ con el siguiente contenido:

```
[zeek-local]
name=Zeek Repo Local
baseurl=http://192.168.1.17
enabled=1
```

b) Instalamos **Zeek** usando el comando yum. Zeek se instalará de forma predeterminada en el directorio **/opt/zeek**.

```
# yum install zeek libpcap-devel
```

c) Verificamos la instalación ejecutando los siguientes comandos:

```
# /opt/zeek/bin/zeek --version
/opt/zeek/bin/zeek version 5.2.1
```

9.7. Instalación y configuración de Protmail

a) Creamos el archivo grafana.repo dentro de /etc/yum.repos.d/ con el siguiente contenido:

```
[grafana]
name=grafana
baseurl=https://rpm.grafana.com
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://rpm.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

b) Instalamos **Promtail** usando el comando yum.

```
# yum install promtail
```

b) Verificamos la instalación ejecutando los siguientes comandos:

```
# promtail --version
promtail, version 2.9.2 (branch: HEAD, revision: a17308db6)
  build user:       root@eee92863de73
  build date:       2023-10-16T14:20:36Z
  go version:       go1.21.3
  platform:         linux/amd64
  tags:             promtail_journal_enabled
```

Para realizar la configuración del Protmail se requiere realizar algunas cambios en el archivo de configuración **/etc/promtail/config.yml** con los siguientes datos:

```
server:
  http_listen_port: 9080
  grpc_listen_port: 0
  log_level: debug

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://192.168.1.72:3100/Grafana Loki/api/v1/push

scrape_configs:
- job_name: zeek
  static_configs:
    - targets:
        - locahost
      labels:
        job: zeek-conn
        host: vm-tfm-sensor
        __path__: /opt/zeek/spool/zeek/*log
```

Luego de realizar los cambios se debe proceder a iniciar el servicio con el comando:

```
# systemctl start promtail

# Podemos visualizar el estado del funcionamiento de promtail con el comando:
# systemctl status promtail
● promtail.service - Promtail service
   Loaded: loaded (/etc/systemd/system/promtail.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-12-04 00:40:59 EST; 5s ago
 Main PID: 11956 (promtail)
    Tasks: 7 (limit: 23142)
   Memory: 20.0M
   CGroup: /system.slice/promtail.service
           └─11956 /usr/bin/promtail -config.file /etc/promtail/config.yml

dic 04 00:41:04 vm-tfm-sensor.innovare.local promtail[11956]: level=info
ts=2023-12-04T05:41:04.078025086Z caller=tailer.go:145>
…
…
```

9.8. Instalación y configuración de Grafana

a) Creamos el archivo grafana.repo dentro de /etc/yum.repos.d/ con el siguiente contenido:

```
[grafana]
name=grafana
baseurl=https://rpm.grafana.com
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://rpm.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

b) Instalamos **Grafana** usando el comando yum.

```
# yum install grafana
```

c) Verificamos la instalación ejecutando los siguientes comandos:

```
# grafana-server --version
```

```
Version 10.2.2 (commit: 161e3cac5075540918e3a39004f2364ad104d5bb, branch:
HEAD)
```

Para iniciar el servicio se usa el siguiente comando:

```
# systemctl start grafana-server

# Podemos visualizar el estado del funcionamiento de grafana con el comando:
# systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; vendor preset:
disabled)
   Active: active (running) since Sun 2023-12-03 02:16:00 EST; 1 day 18h ago
     Docs: http://docs.grafana.org
 Main PID: 14263 (grafana)
    Tasks: 19 (limit: 23142)
   Memory: 140.9M
   CGroup: /system.slice/grafana-server.service
           ├─14263 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini
--pidfile=/var/run/grafana/grafana-server.pid --packaging=rpm cfg:default.paths.logs=/var/l>
           └─14269
/var/lib/grafana/plugins/yesoreyeram-infinity-datasource/gpx_infinity_linux_amd64

dic 04 20:36:00 vm-tfm-grafana.innovare.local grafana[14263]: logger=cleanup
t=2023-12-04T20:36:00.027531166-05:00 level=info msg="Completed cleanup jobs"
duration=9.090638ms
…
```

Para comprobar el funcionamiento de los sensores y validar el envío de los logs a Loki podemos usar el explorador de Granada tal como se muestra en la siguiente imagen:

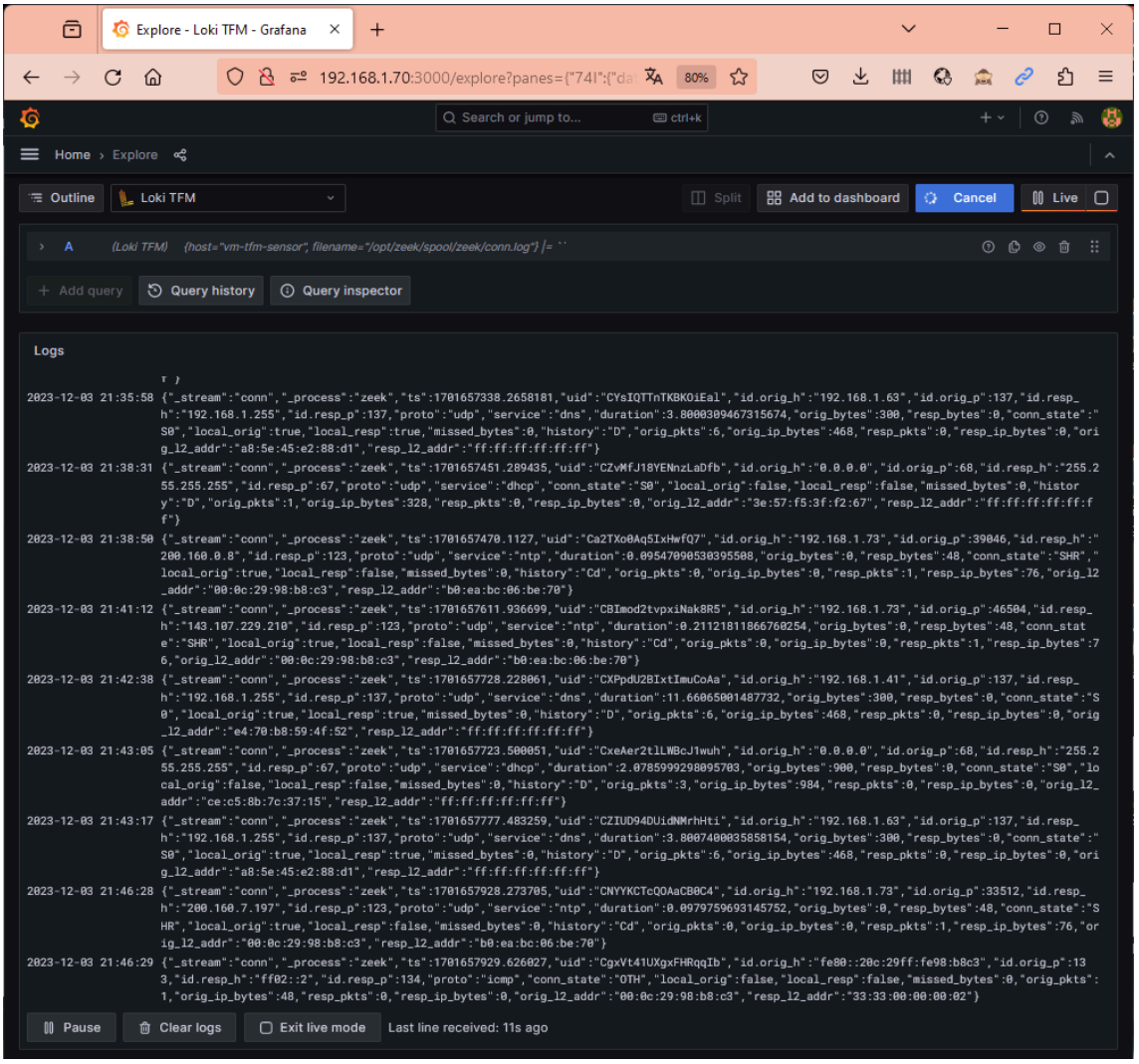


Figura 21: Vista de los registros “conn.log” de Zeek enviados a Grafana Loki