Trabajo Fin de Grado en Ingeniería Informática

Herramienta de trazabilidad y protección para el sistema de correo electrónico de la Universidad de Zaragoza

Traceability and Protection Tool for the Email System of the University of Zaragoza

Autor

Héctor Arcega Vela

Directores

Ricardo Julio Rodríguez Fernández

Víctor Pérez Roche

ESCUELA DE INGENIERÍA Y ARQUITECTURA

Febrero 2025

RESUMEN

Los sistemas de correo electrónico son fundamentales en nuestro día a día. Su funcionamiento depende de servidores específicos que trabajan conjuntamente. No obstante, lograr la integración y coordinación de estos sistemas no es fácil y puede resultar confuso. Por ello, se establecen registros que almacenan toda actividad relacionada con dichas interacciones. La gestión manual de estos registros puede resultar tediosa, dado el gran volumen de datos cuyo formato no suele ser fácilmente interpretable por un humano. En este contexto, la implementación de un programa automatizado es indispensable para ofrecer una representación más accesible y manejable de los datos.

Este Trabajo de Fin de Grado desarrolla una herramienta diseñada para facilitar la interacción y análisis de los datos almacenados en los registros de servidores de correo. El proceso incluye la transformación de los datos a un formato estructurado, eliminando registros innecesarios y unificándolos para mejorar su comprensión. Esta herramienta permite realizar búsquedas entre los datos disponibles, teniendo así todos ellos accesibles en un mismo espacio. Además, cuenta con la capacidad de establecer relaciones entre todos los registros de correo electrónico, representando la información mediante un grafo que detalla los aspectos de la traza completa del envío de un mensaje. Esta funcionalidad posibilita inspeccionar en detalle cada etapa del proceso de envío, facilitando la identificación de posibles problemas o la realización de consultas detalladas de manera eficiente.

ABSTRACT

Email systems are essential in our daily lives. Their operation relies on specific servers that work together. However, achieving the integration and coordination of these systems is not easy and can be confusing. For this reason, logs are established to store all activity related to these interactions. Manually management of these logs can be tedious, due to the large amount of data whose format is not usually easily interpreted by humans. In this context, the implementation of an automated program is essential to provide a more accessible and manageable representation of the data.

In this Bachelor´s Degree Final Project, is developed a tool designed to improve the interaction and analysis of data stored in email server logs. The process includes the transformation of data into a structured format, removing unnecessary logs, and unifying them to improve their understanding. This tool allows searching among the available data, having it accessible in a single space. In addition, it has the ability to establish conections between all email logs, representing the information through a graph that details the aspects of the complete trace of the sending of a message. This feature makes it possible to inspect each stage of the sending process in detail, making easier to identify potential issues or perform detailed queries efficiently.

Índice

Índice de Figuras

Capítulo 1

Introducción

En la actualidad, todos los sistemas tecnológicos generan registros que documentan su actividad. Estos registros (conocidos en inglés como *logs*), contienen información sobre la ejecución de procesos, estado del sistema o cualquier otro dato relevante [1]. Los *logs* engloban una amplia variedad de categorías, incluyendo los asociados a redes virtuales privadas (VPN, *Virtual Protected Network*), accesos a sistemas, servicios de correo electrónico, entre otros.

A pesar de su utilidad como fuente de información detallada, abarcando acciones, eventos y errores, los *logs* suelen ser difíciles de analizar directamente por una persona. Esto se debe a la gran cantidad de información que contienen, generalmente presentada en formatos poco adecuados para el análisis y la explotación eficiente.

En muchas organizaciones, no se dispone de un proceso fácil y accesible para consultar estos *logs*, ya que, por lo general, se realiza mediante comandos ejecutados desde la terminal. Si bien este enfoque es funcional, dificulta considerablemente la extracción de información de manera ágil y eficiente.

Este Trabajo de Fin de Grado (TFG) aborda este problema en el contexto del Servicio de Informática y Comunicaciones de la Universidad de Zaragoza (SICUZ). Esta entidad dispone de un sistema de generación de *logs*. Sin embargo, la gestión de estos en ciertos casos es inexistente o limitada, como ocurre en el caso del sistema de correo electrónico.

Por ello, este TFG propone el diseño e implementación de una herramienta orientada a la gestión de los *logs* asociados al sistema de correo electrónico, centralizando la información, facilitando su búsqueda y permitiendo seguir la traza de un mensaje recibido. Con este enfoque, se pretende no sólo aumentar la eficiencia y facilitar la comprensión de la información, sino también optimizar el manejo de los datos mediante una solución más intuitiva y accesible.

1.1 Objetivos

El objetivo de este trabajo es el del desarrollo de una herramienta software, la cual, a partir de registros de correo electrónico de la Universidad de Zaragoza cedidos por el SICUZ, sea capaz de procesar los registros de actividad del sistema de correo electrónico y que ofrezca la posibilidad de realizar análisis avanzados de los mismos mediante funcionalidades de búsqueda. Además, la herramienta generará un grafo para representar visualmente la traza completa de un envío, ayudando a la identificación de los servidores por los que ha pasado, así como su destino actual. Esta herramienta busca facilitar el rastreo de un correo, incorporando mejoras en la búsqueda y relación de elementos potenciales como remitentes o destinatarios.

El propósito final de esta herramienta es optimizar el trabajo del personal de mantenimiento del correo electrónico, reduciendo significativamente los tiempos necesarios para consultar información. Se busca simplificar y agilizar la tarea de analizar manualmente los miles de registros diarios, ofreciendo una alternativa más intuitiva y eficiente.

1.2 Estructura del documento

Este documento está organizado en cinco capítulos. El Capítulo 1 contiene la introducción y los objetivos del proyecto. El Capítulo 2 describe todos los conocimientos necesarios para la comprensión del proyecto y qué herramientas se han usado para su realización. En el Capítulo 3, se detalla todo el proceso de diseño e implementación junto con aquellas alternativas que finalmente fueron descartadas del diseño final. A continuación, el Capítulo 4 muestra dos casos de uso de la herramienta desarrollada. Finalmente, en el Capítulo 5 se exponen las conclusiones del proyecto y posibles revisiones futuras para que sea una herramienta más completa.

Además, se incluye un anexo con la información referente a horas dedicadas al desarrollo del proyecto, junto con un diagrama de Gantt.

Capítulo 2

Conceptos previos

En este capítulo se comentan todos los aspectos y fundamentos que son necesarios conocer para el desarrollo de la herramienta y su correcta comprensión. Se tratan tanto los aspectos que involucran a la organización SICUZ como aquellos de los que depende la herramienta desarrollada.

2.1 Estructura del sistema base de partida

Para la elaboración de este trabajo se ha partido de los sistemas existentes en el SICUZ, todos ellos relacionados con *logs* y especialmente con aquellos provenientes de los diferentes servidores de correo electrónico con el servicio exim4 instalado.

La arquitectura usada en el entorno de correos electrónicos del SICUZ se estructuran en diferentes categorías, dependiendo del origen y el destino de los correos electrónicos. Los interesantes para el proyecto se describen a continuación.

2.1.1 Mensajes provenientes de cuentas de correo externas a la Universidad

Todo mensaje entrante desde el exterior de la Universidad de Zaragoza es recibido por el sistema `Lavadora`, donde es procesado junto a una gran cantidad de correos electrónicos provenientes de diversas instituciones públicas [2]. Este sistema se encarga de filtrar los mensajes no deseados (*spam*) y el resto de mensajes son posteriormente entregados al servicio de balanceado `relay2.unizar.es`, que, a su vez, se encarga de entregarlos a los sistemas que alojan los buzones de correo. Concretamente: `grio.intra`, `segre.intra`, `cinca.intra`, `queiles.intra`, `onsella.intra`. La Figura 1 muestra la arquitectura descrita.

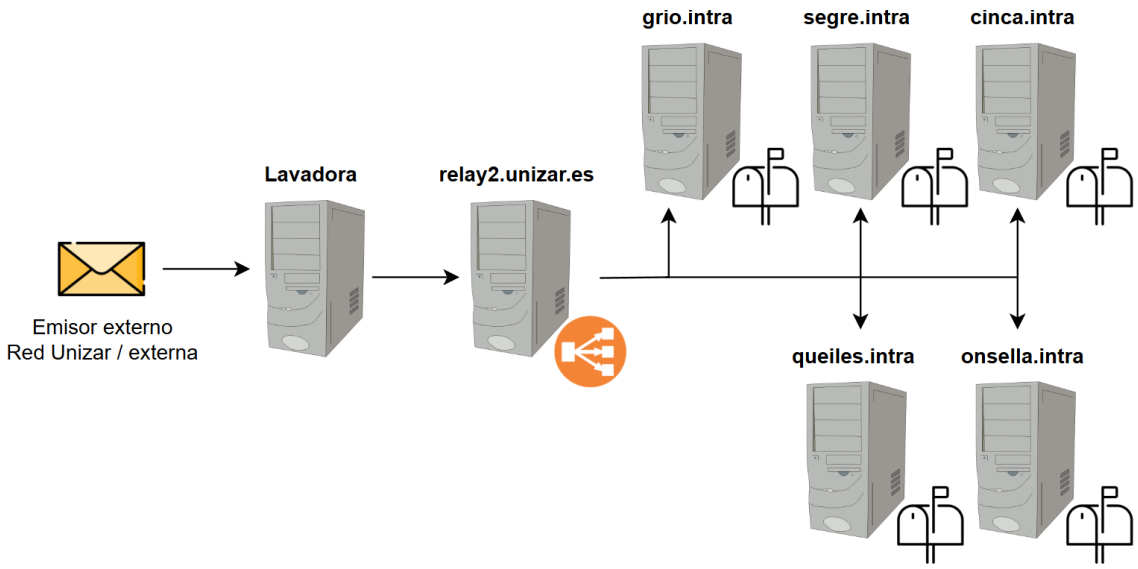

Figura 1: Arquitectura del SICUZ para mensajes provenientes de cuenta de correo externas a la Universidad

2.1.2 Mensajes provenientes desde cuentas @unizar.es desde el exterior de la red Universitaria

Los mensajes enviados desde cuentas @unizar.es que provienen del exterior de la red Universitaria, están sujetos a una política muy restrictiva en cuanto al flujo de envíos. Dichos mensajes son procesados por el sistema `leza.unizar.es`, que se encarga de conectar con los sistemas MX del dominio destino. En el caso de que el destinatario pertenezca al dominio @unizar.es, son redirigidos a uno de los sistemas de buzones previamente mencionados, donde se entregan localmente. Sin embargo, si el usuario ha habilitado una redirección (como por ejemplo a los servidores de Google), esta se procesa desde los sistemas de buzones. En la Figura 2 se representa la arquitectura desplegada para este tipo de envíos.

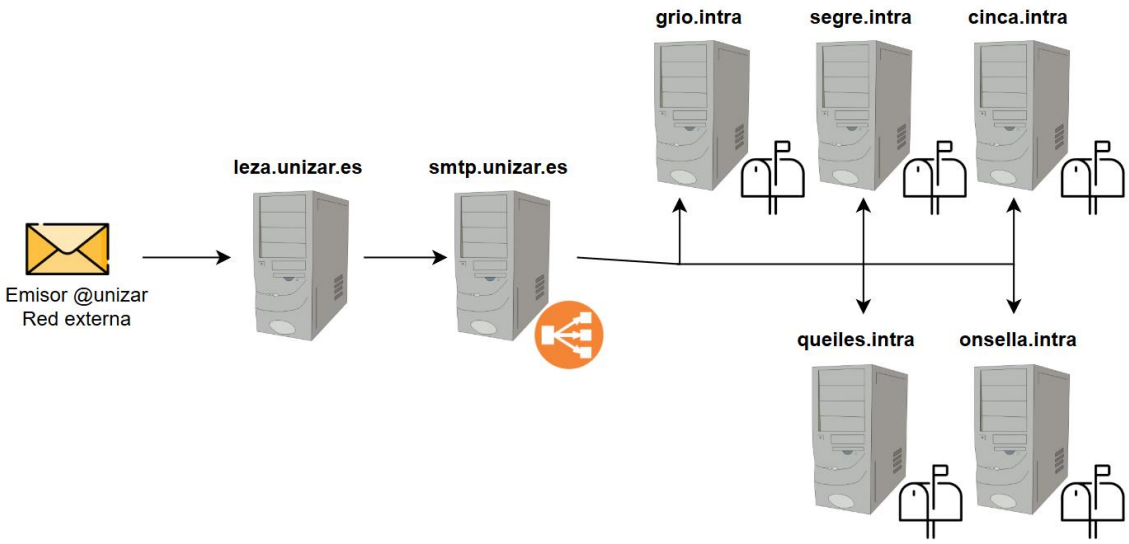

Figura 2: Arquitectura del SICUZ para mensajes provenientes desde cuentas @unizar.es desde el exterior de la red Universitaria

2.1.3 Mensajes provenientes desde cuentas @unizar.es desde el interior de la red Universitaria

Los mensajes enviados desde cuentas @unizar.es que provengan de la red de la Universidad de Zaragoza pasan por los servidores `vhuecha` o `visuela`, que tienen políticas menos restrictivas, y son procesados de forma similar al anterior caso. La Figura 3 representa dicho proceso.

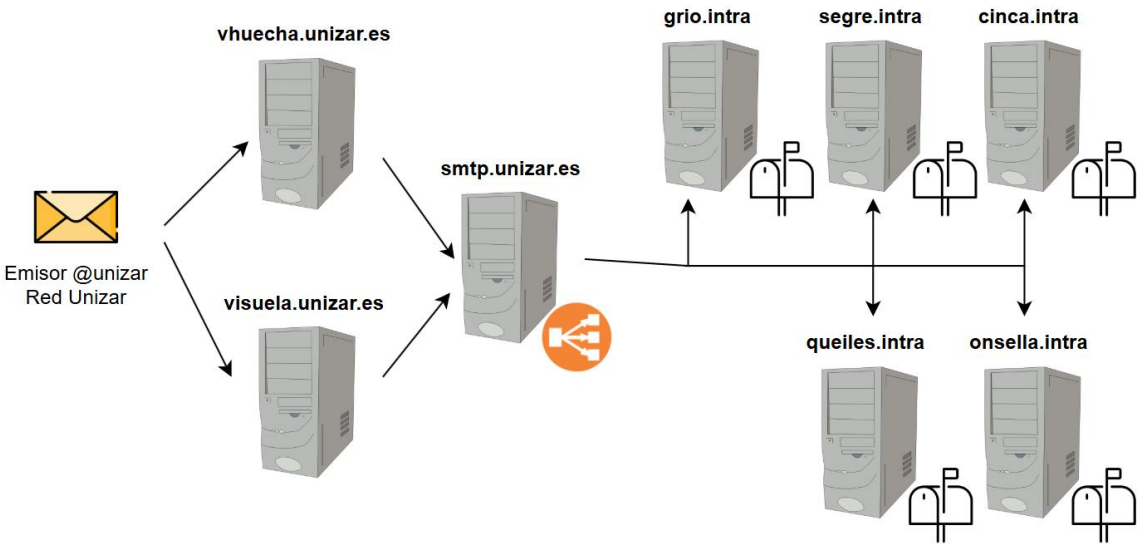

Figura 3: Arquitectura del SICUZ para mensajes provenientes desde cuentas @unizar.es desde el interior de la red Universitaria

2.1.4 Servicios de envío de correo masivo

Además de los servicios previamente mencionados, existen otros servidores configurados para permitir el envío de mensajes sin autenticación provenientes de determinadas direcciones IP o servicios como:

- SMTPSERVER: Este servicio sólo permite el envío a través de ciertas IP autorizadas. En él se usa el servidor `lunada`, que concede un máximo de 4000 mensajes diarios por usuario.
- SMTPSICUZ: A través de este servicio se realiza el envío no autenticado a sistemas considerados seguros como puede ser Moodle y Listas (envío masivo de correos a todos los estudiantes de ese conjunto). Para ello se usan los servidores `ega` y `pajares`, ambos sólo permitiendo envíos por parte de los servidores de Moodle y Listas.

Todos estos servidores mencionados registran su actividad mediante *logs* que se almacenan en diferentes formatos. Uno de estos formatos es el formato exim4, pero personalizado por el SICUZ con la intención de facilitar la comprensión de ciertos registros [3]. La arquitectura de ambos servicios se representa en la Figura 4.

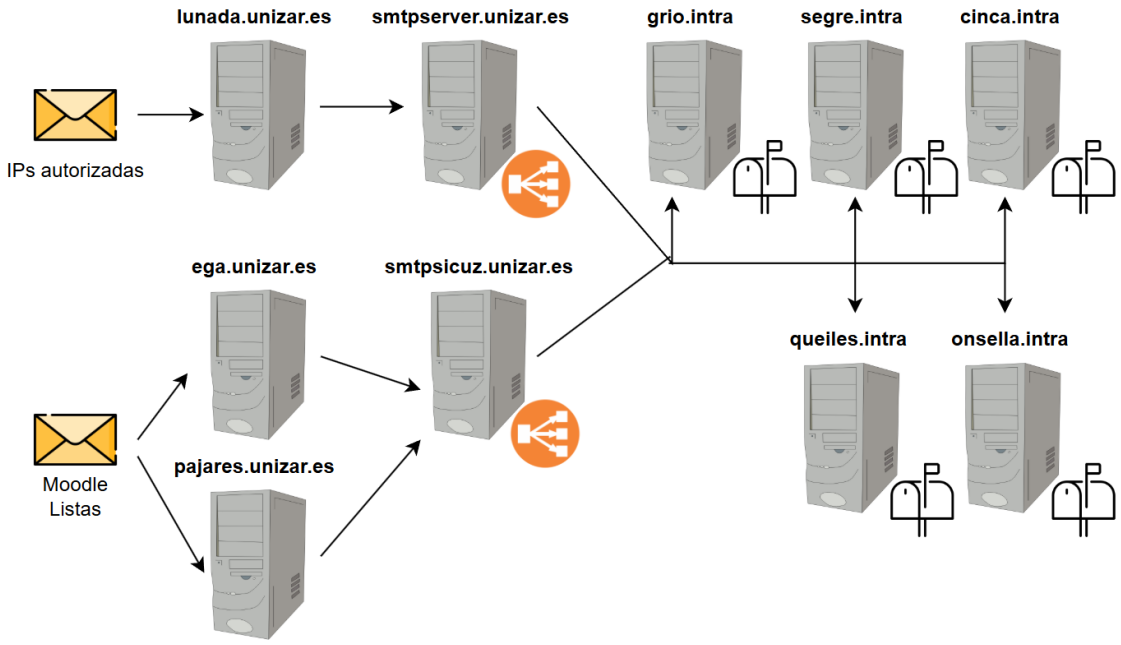

Figura 4: Arquitectura del SICUZ para mensajes de servicios de envío de correo masivo

2.1.5 syslog-ng

syslog-ng es un software de código abierto para la gestión y procesado de mensajes de *log* basados en el protocolo syslog [4]. Se encarga, entre otras funciones, de procesar y redireccionar *logs* de forma rápida y eficiente. Proporciona la capacidad de uso de filtros y personalización de cómo y dónde almacenar la información de múltiples tipos de registros incluyendo *logs* de correo electrónico.

En el contexto del proyecto, todos estos sistemas citados anteriormente están configurados para realizar el envío a tiempo real de los *logs* de actividad de exim4 a un repositorio central ubicado en `rsyslog.intra.unizar.es` utilizando syslog-ng [5].

2.2 Metodología anterior y herramientas de trabajo

Con anterioridad a la implementación de este proyecto, la red de la Universidad de Zaragoza cuenta con un nodo que integra Elasticsearch junto a su visualizador de datos Kibana, similar al de Opensearch, con el objetivo de visualizar todos los registros de la VPN de la institución necesarios para detectar anomalías en los intentos de acceso complementándose con otras herramientas. Sin embargo, en lo que respecta el servicio de correo no existe ninguna herramienta para su gestión. En consecuencia, la única manera de investigar un caso es consultar directamente los registros de los servidores en la máquina `rsyslog.intra.unizar.es` del SICUZ.

A continuación, se explican en detalle cada una de estas herramientas.

2.2.1 Opensearch

Opensearch es un software de código abierto diseñado para búsquedas y análisis. Surge derivado del código fuente de Elasticsearch, tras la decisión de Elastic de aplicar una licencia más restrictiva, mientras que Opensearch continuó desarrollándose como un proyecto de código abierto. Este software es usado para una gran variedad de contextos como el monitoreo de aplicaciones o análisis de registros entre otros. Destaca por su capacidad de manejar grandes volúmenes de datos con un rendimiento excepcional, facilitando así la ingesta y consulta de los datos [6].

Para este proyecto se ha usado la funcionalidad de análisis de registros de Opensearch, siendo estos almacenados en forma de "documentos", que representan un conjunto de componentes y sus valores asociados, los cuales conforman cada registro.

Cabe resaltar que Opensearch es totalmente compatible con herramientas como Logstash y Filebeat, las cuales simplifican enormemente el proceso de extracción, procesamiento e ingesta de los *logs* hacia Opensearch. Estas herramientas se describen brevemente a continuación.

2.3.2 Filebeat

Filebeat es un agente especializado en la recopilación de *logs* y archivos para posteriormente enviar sus datos a un destino específico, ya sea a un sistema final como Opensearch o a un componente intermedio como Logstash [7].

Entre sus múltiples funcionalidades, Filebeat permite configurar expresiones regulares para identificar qué líneas de un *log* deben ser capturadas (incluyendo el caso de que un *log* sea multilínea) e incluso discriminar registros que no sean relevantes por ser ya obsoletos.

Otra característica destacable de Filebeat es su capacidad de mantener un registro de los *logs* que ya ha enviado. Gracias a este mecanismo, en caso de que se detenga su ejecución y posteriormente sea reiniciado, no se envían de nuevo los datos previamente transmitidos, salvo que se elimine dicho registro de actividad.

2.3.4 Logstash

Logstash es una herramienta avanzada para el procesamiento de datos capaz de realizar la ingesta de información de multitud de fuentes y formatos para procesarlos y mandarlos a un destino específico [8].

Logstash recibe los datos en su estado bruto y se encarga de convertirlos en formato JSON, identificando y estructurando los diferentes campos y valores. Para llevar a cabo este proceso, Logstash emplea un filtro Grok, el cual permite modificar o añadir campos adicionales antes de que el resultado sea enviado a Opensearch, enriqueciendo así sus propiedades.

Los pares clave-valor generados tras el procesamiento son enviados al destino configurado. En el caso de este TFG, se envían al índice intermedio de Opensearch.

2.3.5 Filtro Grok

Un filtro Grok es una herramienta destinada a estructurar los datos no estructurados, proporcionando organización y sentido a la información que se le suministre [9]. Esto se consigue mediante una serie de patrones predefinidos compuestos por expresiones regulares diseñadas para identificar determinadas cadenas, las cuales serán asignadas a las claves que se le especifiquen. También permite la creación de patrones personalizados mediante una mezcla de los patrones existentes y nuevas expresiones regulares.

Este filtro se integra en el fichero de Logstash, junto con las definiciones de la entrada y salida de datos.

2.3.6 Dash y Plotly

Dash es un framework de Python que permite la creación de aplicaciones web sin necesidad de emplear lenguajes como JavaScrip [10]. A su vez, Plotly es una potente biblioteca capaz de construir una gran diversidad de gráficos interactivos. Esta herramienta complementa en todos los aspectos a Dash, permitiendo desarrollar una web que combine ambos [11].

En el contexto del proyecto, estas herramientas han sido empleadas para implementar la funcionalidad de búsqueda de documentos y visualización de trazas.

Capítulo 3

Diseño e implementación

En este capítulo se describe la arquitectura del sistema, el origen de los datos con los que se ha trabajado, el procesado de estos y el diseño e implementación de la herramienta.

En la Figura 5 se puede observar un diagrama de la arquitectura del sistema de procesamiento de *logs* desarrollado desde su origen en `rsyslog.intra.unizar.es` hasta su consulta mediante Python pasando por Filebeat, Logstash y Opensearch. Su funcionamiento se describe más adelante.

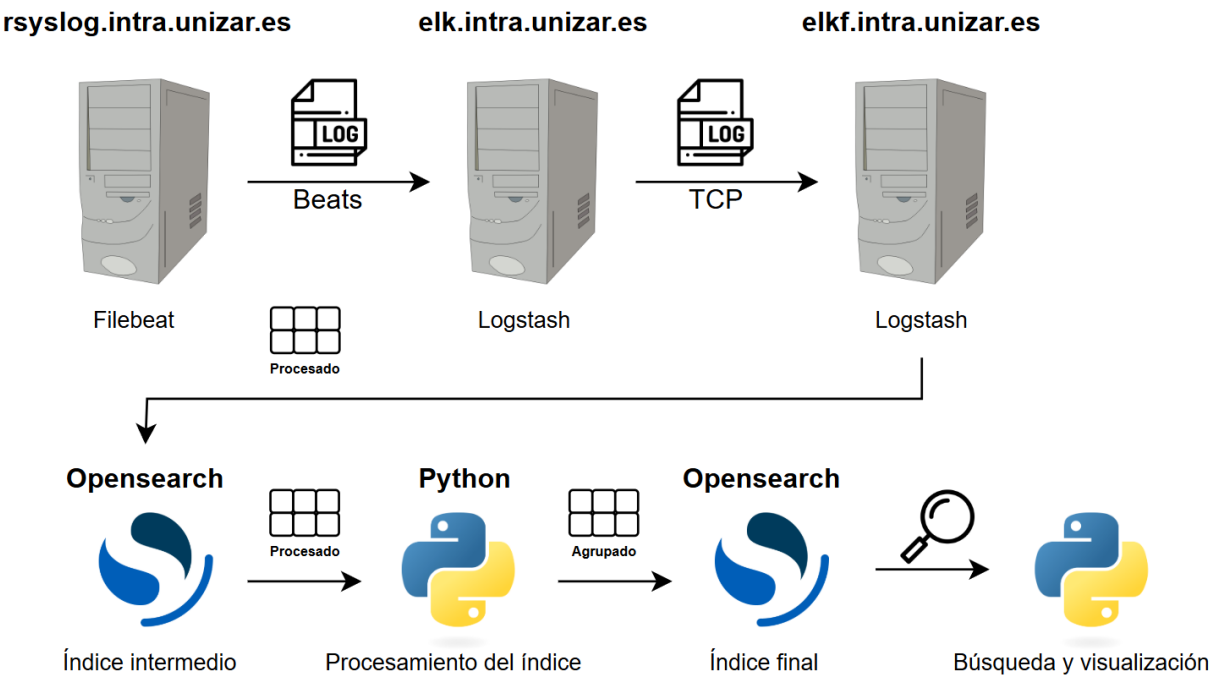

Figura 5: Diagrama del procesamiento de los registros

3.1 Datos, diseño e implementación

3.1.1 Configuración de Filebeat

Para iniciar el proceso de recolección de registros, Filebeat se encarga de leer los *logs* almacenados en el repositorio general de *logs* de la Universidad, `rsyslog.intra.unizar.es`, el cual alberga los registros generados por de exim4 de todos los servidores implicados en el sistema de correo electrónico. Para ello, es imprescindible especificar en su fichero de configuración las rutas de las cuales se extraerán los *logs*, además de una ruta de destino. En este caso, los registros se deben enviar a Logstash especificando la IP y el puerto en el que se encuentra operativo para que este puede enviarlos una vez procesados a Opensearch con una estructura coherente.

3.1.2 Implementación de Logstash y Grok

Una vez que los *logs* han sido procesados por Filebeat, son recibidos por Logstash, donde se les somete a un proceso de normalización antes de ser enviados a OpenSearch.

Para llevar a cabo este procesamiento, se deben definir una serie de filtros Grok. En este caso específico se definen 3 filtros principales: uno para los registros de envío, otro para los de recepción y un tercero para el resto de casos relacionados con el correo, los registros restantes que ayudan a la comprensión lectora de los registros son irrelevantes y se descartan.

Cada uno de estos filtros incorpora patrones creados mediante expresiones regulares y utilizando el lenguaje propio de los filtros Grok, adaptados a los campos específicos que se desean extraer de los registros, contemplando cada uno de los casos posibles [12] [13] [14] [15]. Para el desarrollo de estos patrones, se ha partido de un proyecto en GitHub, diseñado para procesar automáticamente los ficheros exim4 mediante Filebeat. Sin embargo, los patrones definidos en dicho proyecto no corresponden con el formato de los *logs* del SICUZ ya que no son exim4 puro, sino que han pasado por un proceso de enriquecimiento y personalización por parte del SICUZ. Por este motivo, ha sido necesario modificar patrones existentes y crear otros nuevos que faciliten el procesamiento futuro, además de adaptar los patrones base extraídos del proyecto al formato de los fitros Grok [16].

Una vez que los registros han sido procesados por los filtros, se extrae la hora registrada en ese mensaje y se establece como *timestamp*, utilizado por Opensearch para ordenar los registros cronológicamente. Una vez completado este proceso, los registros se envían a un índice en Opensearch. Este primer índice será considerado

como índice intermedio, ya que no es el índice del que se obtendrán los datos finales para la herramienta.

3.1.3 Procesamiento mediante Python

El índice intermedio generado contiene la información de los registros de forma fragmentada. No obstante, la estructura de estos no es plenamente funcional debido a que para un mismo correo, existen en el índice un documento para el emisor y otro por cada destinatario, lo que reduce considerablemente la eficiencia. Por esta razón, se plantea un segundo procesamiento de los datos con el propósito de unificar en un sólo documento todos los documentos de un mismo mensaje.

Para ello, en primer lugar se define un *mapping*, es decir, una especificación de la estructura y los tipos de datos que tendrá el índice final al que se subirán los nuevos documentos. Esta estructura se compone de los campos que aparecen en el documento del emisor, junto con los campos de todos los destinatarios.

Para alcanzar este propósito, se consulta el índice intermedio ya creado y se generan objetos que contienen todas las componentes de cada destinatario. Posteriormente, estos objetos se agregan a la información extraída del emisor, creando un único documento que engloba todos los datos del envío. Finalmente, este documento consolidado se carga en el índice final. Es importante mencionar que sólo se procesan los documentos cuyo mensaje enviado se ha clasificado como completado, ya que la integración de los datos requiere que tanto el envío como la recepción del mensaje haya concluido con éxito. Una vez procesado se eliminan todos los registros usados del índice intermedio, optimizando el uso de espacio [17].

Un ejemplo simplificado de este proceso es el siguiente:

Índice intermedio:

documento 1:

exim4.emisor: emisor@unizar.es

documento 2:

exim4.destinatario: d1@unizar.es

documento 3:

exim4.destinatario: d2@unizar.es

Índice final:

documento 1:

exim4.emisor: emisor@unizar.es

exim4.all_destinatarios.exim4.destinatario: d1@unizar.es, d2@unizar.es

3.1.4 Herramienta web de trazabilidad

Se ha desarrollado una herramienta capaz de rastrear y visualizar la traza completa de un correo electrónico que ha circulado por alguno de los servidores de la Universidad de Zaragoza. Con esta herramienta es posible observar y analizar el recorrido completo de un mensaje desde que su envío hasta que se deposita en un buzón.

Cada servidor por el que pasa un mensaje asigna dos elementos cruciales: un identificador único para el mensaje en ese servidor y el identificador que será utilizado por el siguiente servidor en la cadena. Con estos datos, es posible reconstruir el trayecto exacto del mensaje mediante la consulta de registros. Un registro cuyo identificador principal coincide con el identificador de destino de otro define la secuencia de movimiento, permitiendo determinar si un servidor es previo o posterior en la ruta.

Tras obtener en orden los identificadores principales de cada registro del mensaje en los servidores, se realiza una consulta para recuperar la información detallada de cada etapa. Es importante subrayar que no todos los registros del correo tienen asignado un identificador de destino asociado a uno de los servidores de la Universidad. Esto ocurre cuando el destino es externo, como puede ser una redirección a servicios de correo externo (por ejemplo, Google). En tal situación, la única información disponible proviene del registro previo al salto externo.

Una vez identificado el trayecto completo del mensaje, se puede comenzar con la preparación del grafo. En primer lugar, se calcula el número de nodos necesarios para representar cada una de las etapas del trayecto, cuáles serán las posiciones de dichos nodos en el grafo y además qué nodos están unidos entre sí facilitando la interpretación del recorrido y sus ramificaciones.

Para enriquecer la consulta de información, cada nodo incluye una etiqueta interactiva que se muestra al pasar el cursor sobre él, que incluye la siguiente información clave:

- Identificador del correo en ese servidor
- Servidor actual
- Correo del emisor
- Servidor de destino
- Correo de los destinatarios
- Fecha

En el grafo, el nodo más a la izquierda representa el primer registro del mensaje, mientras que el nodo final de cada una de las ramas representa si el correo ha sido destinado a un servidor externo, si se ha depositado en uno de los sistemas de buzones

de la Universidad o si se carece de información sobre el mensaje en dicho servidor debido a registros eliminados.

Los nodos se complementan con una paleta de colores. Aquellos que son azules indican que en dicho servidor se le ha asignado un identificador y hay constancia de su registro. Un detalle importante es que en caso de que un nodo indique que el servidor de destino es `smtp.unizar.es`, el siguiente servidor será uno de los sistemas de buzones anteriormente mencionados. Por tanto, en el siguiente nodo se puede consultar cuál de ellos es. Que el destino de un mensaje sea un servidor de buzón no significa que se entregue, ya que en caso de que el destinatario tenga habilitada la redirección a otro servicio como Google, no se realizará maildrop, es decir, el correo no habrá sido depositado en uno de los sistemas de buzones interno y por tanto el siguiente nodo no será gris sino verde. Un color verde en el nodo indica que la información de ese registro es inaccesible, ya sea por redirección a un servidor externo o no hay información disponible acerca de un determinado registro. Un nodo de color gris indica maildrop, y por tanto el mensaje ha sido depositado en un buzón. Dicho buzón es representado por el nodo anterior del que parte. Por último, un nodo de color amarillo indica el registro desde el que ha comenzado el proceso de búsqueda, es decir el documento a partir del cual se ha consultado realizar la traza.

La visualización del grafo incluye una lista con los identificadores representados en el él para facilitar la búsqueda de información acerca de ellos.

En la Figura 6 se presenta un ejemplo de traza de un correo electrónico, donde surgen todos los casos posibles. El nodo amarillo corresponde con el servidor `cinca`, cuyo identificador de mensaje coincide con el que se ha iniciado la consulta. Se observa que el mensaje tiene un registro anterior en `visuela` y uno posterior en `ega`. Además, se indica en el nodo gris que, para los destinatarios existentes en dicho nodo, el mensaje ha entregado en `cinca` y por otro lado se ha redireccionado a Google para el resto de los destinatarios especificados en el nodo verde.

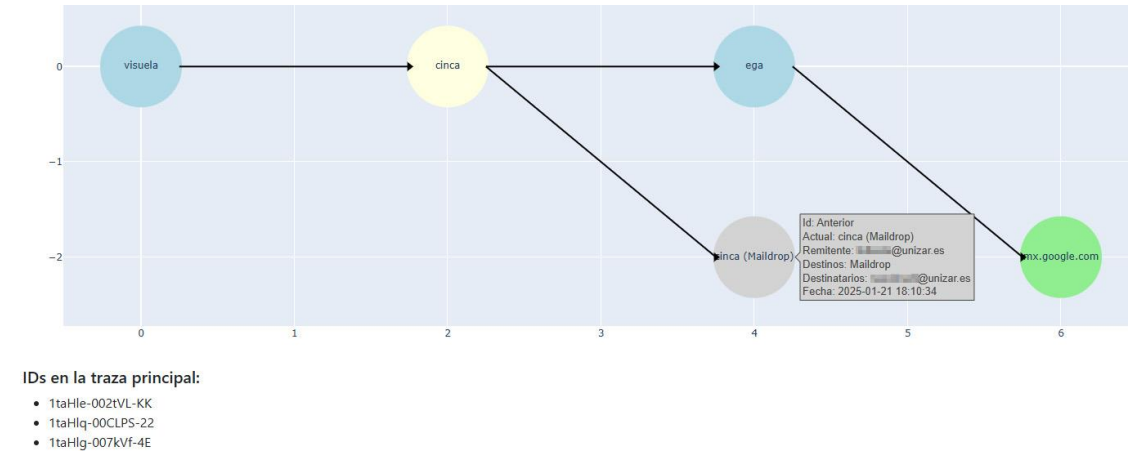

Figura 6: Traza completa de un mensaje a partir del identificador asignado en `cinca`

3.1.5 Herramienta web de búsqueda

Complementaria al grafo, esta herramienta ofrece la posibilidad de realizar búsquedas rápidas sobre los términos más relevantes: el identificador del mensaje, el emisor, el destinatario y un rango de fechas. Estas búsquedas se ejecutan mediante consultas al índice final alojado en Opensearch. Los resultados de la búsqueda se organizan en una disposición clara donde cada coincidencia se muestra en una caja con los siguientes elementos: identificador del mensaje, fecha, servidor actual, remitente, destinatarios, servidor de destino e identificador de destino.

Junto a los componentes de cada resultado se encuentran también dos botones. El primero de ellos permite abrir una nueva vista en la que se genera el grafo de la traza correspondiente al identificador del mensaje seleccionado. El segundo botón permite acceder al documento correspondiente a dicho mensaje en Opensearch. Esta opción es especialmente útil para quien desee examinar en detalle toda la información asociada al mensaje.

En la Figura 7 se ilustra la vista principal de la herramienta de búsqueda. Por su parte, la Figura 8 presenta el desplegable que contiene todas las opciones de búsqueda disponibles, mientras que en la Figura 9 se muestra un ejemplo concreto de los resultados obtenidos tras una búsqueda.

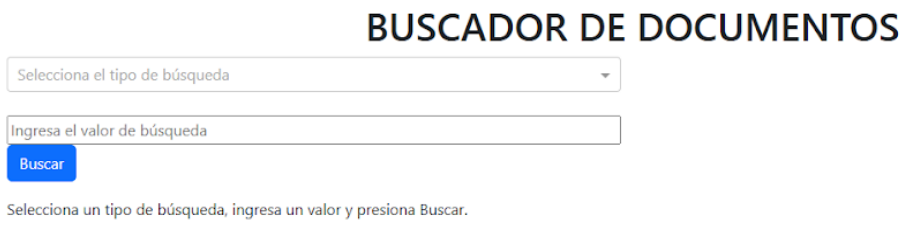

Figura 7: Página principal de la herramienta de búsqueda

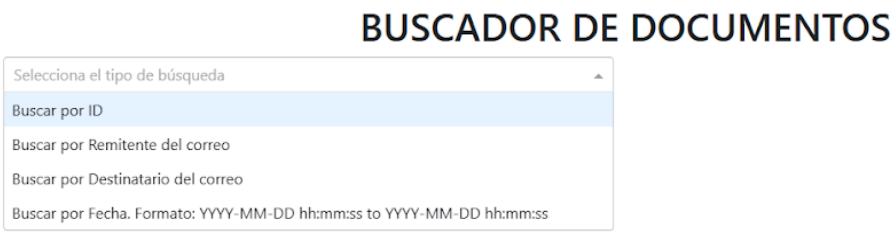

Figura 8: Desplegable con las opciones de búsqueda disponibles

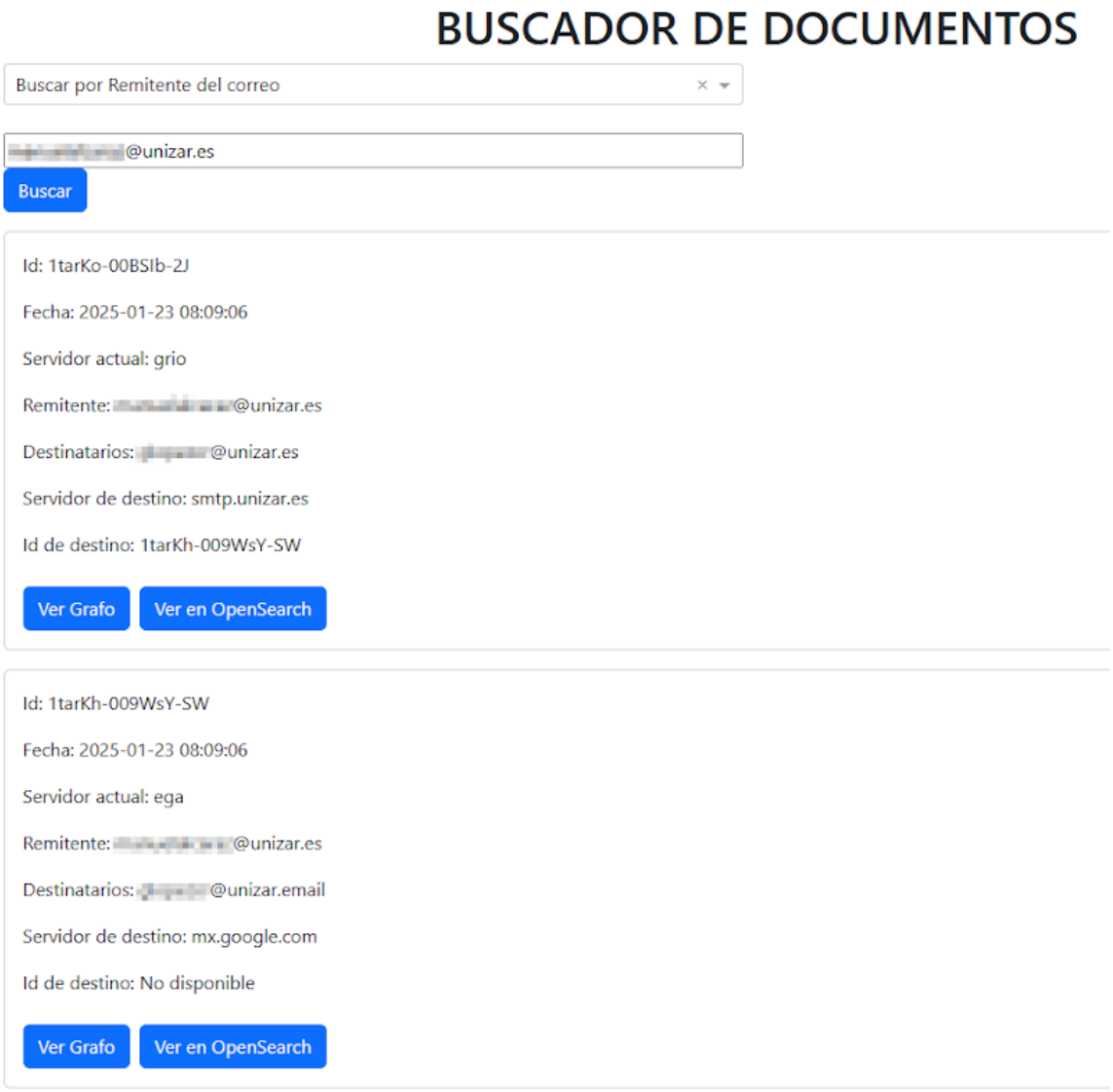

Figura 9: Resultados de una búsqueda por correo electrónico remitente

3.1.6 Aspectos de la puesta en producción

Para la implementación en producción de la herramienta, se han realizado algunos ajustes esenciales para garantizar su correcto funcionamiento.

En primer lugar, en la máquina `rsyslog.intra.unizar.es`, ya existe una instancia de Filebeat configurado y activo, que procesa *logs* de otros servicios como la VPN y los envía a `elk.intra.unizar.es`. Debido a que Logstash en este proyecto debe ser ejecutado en `elkf.intra.unizar`, ha sido necesario realizar una redirección. Para ello, teniendo en cuenta que Filebeat no permite redireccionar a máquinas diferentes simultáneamente, desde `elk.intra.uniar.es` se establece una la redirección de la información mediante Logstash a `elfk.intra.unizar.es` por TCP en formato JSON en lugar de *beats*.

Para el procesamiento entre el índice intermedio y el final mediante Python, se ha establecido en `elkf.intra.unizar.es` un Cron que realiza múltiples tareas. En primer lugar, cada 10 minutos se ejecuta el código en Python. Para prevenir la creación de múltiples instancias o fallos en caso de que la ejecución dure más de 10 minutos se usa Flock, una herramienta capaz de establecer un fichero que sirve de bloqueo (es decir, cuando termina la ejecución del programa en Python el fichero es eliminado, pero mientras se está ejecutando dicho fichero existe y Cron no será capaz de ejecutar el código). Además, Cron también se encarga de activar el entorno virtual requerido para la correcta ejecución del *script* en Python y las bibliotecas necesarias.

Las herramientas de búsqueda y traza de correos se han desplegado en `elkf.intra.unizar.es` y es posible acceder a ellas mediante el navegador especificando dicha máquina y el puerto correspondiente, únicamente si se pertenece al grupo de administración de la Universidad.

Por último, la Figura 10 muestra la arquitectura final para despliegue completo la herramienta en el entorno de producción.

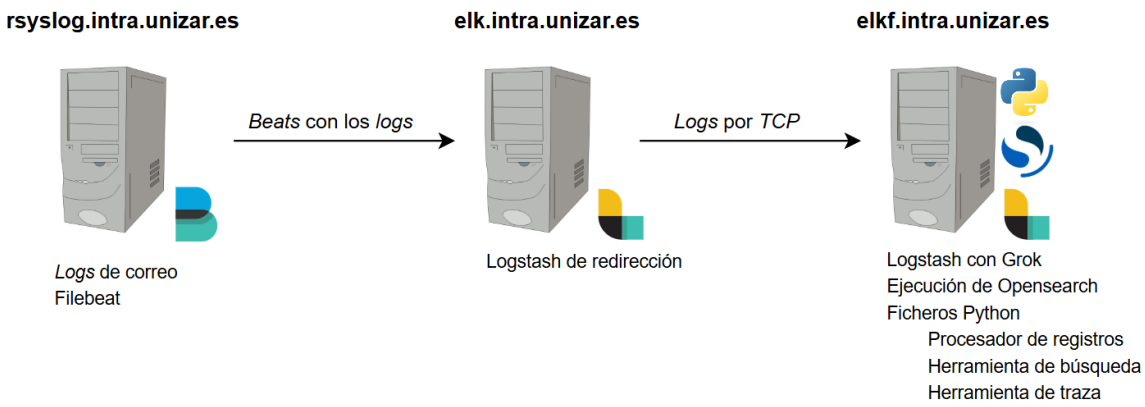

Figura 10: Arquitectura del sistema

3.2 Alternativas descartadas

3.2.1 Unir a los destinatarios mediante Logstash

Se llevaron a cabo diversas pruebas con el objetivo de unificar en un único documento la información del emisor y de los destinatarios de un mensaje únicamente desde Logstash. Sin embargo, esta herramienta no está diseñada para gestionar un nivel de complejidad tan elevado, ya que su funcionamiento se basa en el procesamiento secuencial de los registra a medida que estos se reciben.

Dada esta limitación, se determinó que la solución más adecuada para obtener una estructura de datos coherente y funcional, acorde a los requerimientos del TFG, consistía en implementar un procesamiento adicional que se apoyara en el uso de ambos índices.

3.2.2 Grafo en Opensearch

En un primer momento, se contempló realizar el grafo de la traza del correo dentro del entorno de Opensearch, aprovechando su capacidad para realizar diferentes visualizaciones de los datos, incluyendo vistas personalizadas mediante el lenguaje Vega. No obstante, lograr representar los datos de la forma deseada es muy complejo y con un nivel de personalización limitado respecto a lo que es posible realizar mediante el uso de bibliotecas de Python como Dash y Plotly.

3.2.3 Opensearch ante Elasticsearch

Al comienzo del proyecto, surgió la duda entre optar por Opensearch o Elasticsearch. Ambas presentan buenas soluciones, pero muchas del configuracioens disponibles de forma gatuita en Opensearch no lo son en Elasticsearch [18].

Considerando que herramientas como Logstash y Filebeat son totalmente compatibles con Opensearch y éste comparte la mayor parte de las funcionalidades de Elasticsearch, se decidió que Opensearch era la mejor opción.

3.2.4 Procesamiento de todo tipo de registros

El planteamiento inicial para procesar los datos era tratar cada uno de los registros almacenados, independientemente se aquellos obligatorios para el funcionamiento del sistema como aquellos considerados irrelevantes, entre los cuales predominan principalmente errores.

Finalmente, se decidió que lo mejor era no incluir estos registros en el índice intermedio ya que muchos de ellos carecen de identificador por lo que permanecen en

el índice hasta que sean eliminados manualmente. A pasar de esta decisión, en caso de que sea necesario consultar detalles relacionados con errores de envío y similares siempre es posible obtener los datos necesarios mediante la herramienta y realizar una consulta rápida entre los *logs* de los servidores mediante comandos de terminal.

3.2.5 Múltiples instancias de Filebeat

En `rsyslog.intra.unizar.es` ya existe una instancia de Filebeat activa y configurada para enviar registros a `elk.intra.unizar.es`. Sin embargo, para este proyecto es necesario crear una segunda instancia de Filebeat. Esto es un proceso largo y complejo que podría llegar a afectar a la instancia actual. Para evitar conflictos, se decidió emplear la misma metodología que sigue la instancia ya lanzada: usar `elk.intra.unizar.es` para redireccionar los *logs* a la máquina deseada mediante Logstash. Esto no afecta a la eficiencia y mantiene los sistemas ordenados.

Capítulo 4

Casos de uso

En este capítulo se presentan una serie de casos de uso para mostrar el funcionamiento de la herramienta desarrollada. Estos casos presentan:

1) un usuario ha reportado que un correo que no le ha llegado

2) un posible acceso al sistema de correo mediante un mensaje malicioso.

4.1 Correo que no llega a su destinatario

Este caso engloba diferentes escenarios que se tratarán de igual manera con la herramienta. Estos escenarios son: el correo no llega a un destinatario de la Universidad de Zaragoza, no llega a un destinatario externo o que no llega a ciertos suscriptores de una lista.

Una vez que un usuario ha reportado el problema, se le solicitan datos por los que comenzar la búsqueda. Estos datos pueden ser el remitente o la fecha en la que sucedió, y a partir de esta información se busca localizar las trazas de dicho envío que se encuentren en el sistema. Con los datos, se realiza una consulta en la herramienta. Una vez que se ha identificado el mensaje, se genera su traza verificando si el usuario que ha reportado el error se encuentra entre los destinatarios, ya sea de un buzón de la Universidad o uno externo, y se verifica si la entrega ha resultado exitosa o ha habido algún error por el camino. En caso de observar una anomalía, gracias al identificador asignado por el servidor en el que dicho envío se ha bloqueado, es posible acceder al documento de dicho mensaje en Opensearch donde se encuentran disponibles todos los detalles del mensaje y su posible error.

Finalmente, en caso de que sea necesario profundizar en la investigación, dado que se conoce el identificador del mensaje y el servidor del error, se puede realizar consulta rápida entre los *logs* almacenados en `rsyslog.intra.unizar.es` mediante comandos de terminal para ver al completo todos los posibles mensajes de error que pueden estar relacionados con dicho mensaje.

4.2 Correo malicioso

Dado un escenario en el que se ha detectado un envío de un correo fraudulento, se deben realizar varias comprobaciones para investigar el origen y causa de dicho envío.

El objetivo de la herramienta desarrollada es obtener el buzón de la Universidad de Zaragoza en el que ha sido entregado dicho correo malicioso, para a partir de ahí, obtener más información. Para lograr esto se debe consultar en la herramienta el remitente o destinatarios de dicho envío. Una vez que se ha identificado el mensaje de entre los resultados de la búsqueda, se procede a generar la traza. En ella es posible visualizar qué rama termina con nodo gris, es decir, obtener el servidor de buzón en el que se ha depositado el mensaje. Consultando la información de dicho nodo, se obtiene toda la información necesaria para proseguir con la investigación.

Capítulo 5

Conclusiones y trabajo futuro

5.1 Conclusiones

En este TFG se han desarrollado una serie de herramientas que, trabajando conjuntamente, son capaces de procesar *logs* de servidores de correo electrónico de manera automática, permitiendo además su consulta posterior. La base de este proceso se encuentra en el uso de las herramientas Filebeat, Logstash y Opensearch. Todo ello se complementa mediante el procesamiento implementado en Python que, con la ayuda de ciertas bibliotecas, facilita mucho el trabajo de personalización de los datos.

El procesamiento de los datos entre índices realizado con Python no es tan eficiente como lo son Logstash o Filebeat, pero es necesario para conseguir una fuente de datos organizada para que no sólo por el sistema de visualización de la herramienta sea capaz de trabajar sobre ella, sino para que un ser humano sea capaz de interpretar los datos mucho más rápido. Esta pequeña ralentización en el procesamiento no influye en el resultado, ya que el tiempo transcurrido desde que el mensaje se envía hasta que es procesado por Python es de apenas unos minutos.

Respecto a la visualización y consulta, se ha implementado una solución que respeta la accesibilidad y comprensión de los datos, agilizando el proceso de consulta mediante una interfaz. Esta funcionalidad, combinada con la representación de trazas, proporciona una visión completa de la información de un mensaje durante todo su recorrido para los diferentes destinatarios. Sin esta herramienta, el proceso de relacionar los registros correspondientes a un mismo mensaje y trazar su recorrido sería un proceso laborioso y difícil de manejar.

5.2 Trabajo futuro

A pesar de que esta herramienta es útil y eficiente, no es perfecta. Existen posibles mejoras futuras que harían de esta herramienta una, más robusta y completa:

Implementación de búsqueda de documentos mediante múltiples campos simultáneamente: Actualmente la herramienta sólo permite la búsqueda de un determinado campo de los registros. Sin embargo, para realizar una búsqueda más específica, sería interesante establecer un nuevo sistema de búsqueda con múltiples filtros que combinados a petición del usuario realicen una búsqueda más personalizada.

Detección de comportamientos sospechosos: Ya que para la realización de la herramienta se han procesado todos los registros, sería útil usar estos recursos para desarrollar una herramienta capaz de detectar comportamientos sospechosos, como pueden ser envíos masivos de correos, envíos fuera del horario habitual, acceso a buzones, envío de spam, entre otros.

Añadir nuevas fuentes de datos: Para complementar la información actual, se podrían procesar otras fuentes de datos que permitan realizar consultas para casos más complejos, normalmente relacionadas con peticiones de usuarios del sistema de correo. Por ejemplo, usuarios que reportan no tener un correo en su buzón, pero sin embargo en los registros del sistema aparece como entregado y depositado.

Bibliografía

[1] M. Cancemi, «Registros de Logs de TI. ¿Qué son y cómo funcionan?,» Libertia, 2024. [En línea]. Available: https://libertia.es/registros-de-logs/.

[2] «RedIRIS - Servicio LAVADORA - Plataforma Unificada AntiSpam de RedIRIS,» RedIRIS, 2024. [En línea]. Available: https://www.rediris.es/lavadora/.

[3] Redmine-SICUZ, «SUZ_073_040_Correo_electrónico,_listas,_transferencia_ficheros_Redmine,» [En línea]. Available: https://redmine.unizar.es/.

[4] R. Gerhards, «The Syslog Protocol,» 2009.

[5] «syslog-ng - Log Management Solutions,» syslog-ng, 2025. [En línea]. Available: https://www.syslog-ng.com/.

[6] «¿Qué es OpenSearch? - Explicación de Open Source Search Engine - AWS,» Amazon Web Services, Inc., 2024. [En línea]. Available: https://aws.amazon.com/es/what-is/opensearch/.

[7] «Filebeat: Análisis de logs ligero y Elasticsearch,» Elastic, 2025. [En línea]. Available: https://www.elastic.co/es/beats/filebeat.

[8] «Logstash: recopila, parsea y transforma logs | Elastic,» Elastic, 2025. [En línea]. Available: https://www.elastic.co/es/logstash.

[9] «Grok filter plugin | Logstash Reference [8.17] | Elastic,» Elastic, 2025. [En línea]. Available: https://www.elastic.co/guide/en/logstash/current/plugins-filters-grok.html.

[10] «Dash Documentation & User Guide | Plotly,» Plotly, 2025. [En línea]. Available: https://dash.plotly.com/.

[11] «Plotly,» Data Apps for Production | Plotly, 2025. [En línea]. Available: https://plotly.com/python/.

[12] «Test grok patterns,» Grok Constructor, [En línea]. Available: https://grokconstructor.appspot.com/do/match#result.

[13] «Grok Patterns,» Grok Constructor, [En línea]. Available: https://grokconstructor.appspot.com/groklib/grok-patterns.

[14] «regex101: build, test, and debug regex,» regex101, [En línea]. Available: https://regex101.com/.

[15] T. d. l. Fuente, «regexp,» [En línea]. Available: https://blyx.com/public/docs/curso-linux-principiantes/regexp.html.

[16] lbausch, «Github - filebeat-exim4,» GitHub, 2024. [En línea]. Available: https://github.com/lbausch/filebeat-exim4.

[17] «opensearch-py,» PyPI, 2025. [En línea]. Available: https://pypi.org/project/opensearch-py/.

[18] U. S. E. K. Sokratis Papadopoulos, «Architecting the OpenSearch service at CERN,» Switzerland, 2024.

Anexos A

Dedicación

La dedicación invertida en este proyecto se ven reflejados en el diagrama de Gantt de la Figura 11 y en la tabla de horas dedicadas de la Figura 12. El diagrama de Gantt muestra las diferentes fases por las que se ha pasado para el desarrollo del proyecto. En la tabla de horas dedicadas de la Figura 12, se concreta la cantidad de horas dedicadas a cada una de las fases.

	nov 2024	dic 2024	ene 2025
Comprensión de la base del SICUZ			
Familiarización con las herramientas			
Desarrollo del procesamiento de los datos hasta el índice intermedio			
Desarrollo del procesamiento de los datos hasta el índice final			
Desarrollo de la herramienta de trazas			
Desarrollo de la herramienta de búsqueda			
Mejoras a los filtros y procesado			
Implementación a nivel de producción			
Redacción de la memoria			

Figura 11: Diagrama de Gantt

	Horas
Comprensión de la base del SICUZ	5
Familiarización con las herramientas	3
Desarrollo del procesamiento de los datos hasta el índice intermedio	45
Desarrollo del procesamiento de los datos hasta el índice final	62
Desarrollo de la herramienta de trazas	104
Desarrollo de la herramienta de búsqueda	26
Mejoras a los filtros y procesado	17
Implementación a nivel de producción	24
Redacción de la memoria	82
Total	368

Figura 12: Horas invertidas