Universitat Oberta de Catalunya

uoc.edu

Detección de anomalías con Elastic Stack

Henry Patricio Farinango Endara
Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC)
Seguridad empresarial

Miguel Ángel Flores Terron
Víctor Garcia Font

29/12/2020

GNU Free Documentation License (GNU FDL)

Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Detección de anomalías con Elastic Stack*
Nombre del autor:	*Henry Farinango Endara*
Nombre del consultor/a:	*Miguel Flores Terron*
Nombre del PRA:	*Víctor Garcia Font*
Fecha de entrega (mm/aaaa):	29/12/2020
Titulación:	*Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC)*
Área del Trabajo Final:	*Seguridad empresarial*
Idioma del trabajo:	*Español*
Palabras clave	*Seguridad empresarial, Elastic Stack, Wazuh*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

La dependencia de la tecnología hace que las empresas en la actualidad implementen seguridad a varios niveles con la finalidad de evitar que la empresa tenga problemas de amenazas contra la seguridad de la información y afecte a sus recursos internos de forma crítica.

En este trabajo de fin de Máster se implementa la solución de seguridad basada en la arquitectura “todo en uno” de Wazuh y Elastic Stack a manera de laboratorio, con la finalidad de realizar pruebas de concepto para la detección de anomalías que se presentan en los dispositivos de una red LAN, en este caso en concreto del proyecto, para servidores que están en una DMZ, que compone al agente de Wazuh. De esta manera, el aporte de seguridad de forma proactiva con la recolección de logs en tiempo real, permite que este sistema en cuestión genere alertas en caso de intentos de ataques y ejecute Active Response, acción que permita mitigar el incidente detectado.

Este proyecto promueve las soluciones basadas en software libre, validando que es una solución completa de seguridad empresarial en el contexto de analisis de datos de registro para securizar host de la red interna empresarial.

Se llega a concluir que la solución es idónea para entornos empresariales de cualquier índole, más aún para entorno pequeños como lo simulado. Considerando que la forma de automatizar las respuestas contra incidentes de seguridad propone una gran alternativa en el campo de las tecnologías d ela información.

Abstract (in English, 250 words or less):

The dependence on technology makes companies nowday implement security through multiple levels in order to prevent the company from having problems of threats against information security and affecting its internal resources in a critical way.

In this Master's thesis, the security solution based on the “all-in-one” architecture of Wazuh and Elastic Stack is implemented as a laboratory, in order to carry out proofs of concept for the detection of anomalies that occur in the devices on a LAN network, in this case specifically for servers that are in a DMZ, which makes up the Wazuh agent. In this way, the security contribution proactively with the collection of logs in real time, allows this system in question to generate alerts in case of attempted attacks and execute Active Response, an action that allows mitigating the detected incident.

This project promotes the opensource software solutions, validating that it is a complete business security solution in the context of log data analysis to secure host of the internal business network.

It is concluded that the solution is ideal for business environments of any kind, even more for small environments such as ours simulated. Considering that the way to automate responses against security incidents proposes a great alternative in the field of information technology.

Índice

Lista de tablas

Lista de figuras

1. Introducción

1.1. Contexto y justificación del Trabajo

Hoy en día, el uso de Internet está extendido a nivel mundial y su fenómeno trae múltiples ventajas, pero también varios problemas. A medida que crece el número de usuarios, la frecuencia, variedad y malignidad de los ataques sufridos a través de la red se ha incrementado drásticamente en los últimos años.

Se precisa en [1] que los primeros 9 meses del 2019 se efectivizaron 5.183 ciberataques a nivel mundial para varias compañías, donde se expusieron 7.900 millones de registros que con respecto al año anterior se han incrementado a más del doble. Clara evidencia de que los ataques informáticos a las redes de datos van en aumento y para ello es necesario de alguna manera mitigarlos.

La dependencia de la tecnología hace que las empresas en la actualidad implementen seguridad a varios niveles para la protección de datos. Para evitar que la empresa tenga problemas de amenazas contra la seguridad de la información y afecte a sus recursos internos de forma desastrosa en las áreas más trascendentales como el sector productivo y financiero. Las organizaciones actualmente contratan personal experto en tecnologías de la información que sean capaces de utilizar herramientas para predecir dichas amenazas y proteger de los riesgos que se pueden presentar.

Por ello, las anticuadas protecciones centradas en el ordenador se han visto obligadas a adaptarse a la nueva situación para brindar una protección eficaz frente a cualquier tipo de amenaza. De tal manera que, el primer escudo para una red de datos es un Firewall en la seguridad perimetral, mientras que los IDS (Intrusion Detection System) se considera como la segunda barrera de seguridad en la protección del sistema de seguridad.

De esta manera, los sistemas de detección de intrusiones en la red (NIDS) y los sistemas de detección de intrusiones basado en host (HIDS) se han postulado como protección eficaz, ya que se centran exactamente en distinguir a los usuarios de red legítimos de los maliciosos, análisis de logs, detección de vulnerabilidades, entre otros. En el mismo sentido existen muchas más características que estos sistemas brindad a la seguridad enfocados para respuesta ante incidentes.

Para esto es importante la prevención de amenazas y riesgos para los sistemas de información internos, así evitando intenciones maliciosas de intrusos dentro de los sistemas privados de las empresas.

Este trabajo de fin de master se centra en la detección de anomalías para una red en el ámbito empresarial en donde se pretende realizar un análisis de Seguridad a los servicios de la misma mediante el Elastic Stack para hacer el trabajo de deteccion de anomalías más eficiente de una forma dinámica que en

concordancia con el Firewall se brinde un apoyo para mejor la capacidad de respuesta a los incidentes.

La Respuesta Activa o Active Response es una forma de hacerlo, para que se realice una respuesta a los ataques en tiempo real sin necesidad de asesoramiento humano. Se ejecuta cuando se activa una alerta, un nivel de alerta o un grupo de reglas específicos. Las respuestas activas son respuestas con estado o sin estado. Las respuestas con estado se configuran para deshacer la acción después de un período de tiempo específico, mientras que las respuestas sin estado se configuran como acciones únicas [2].

1.2. Objetivos del Trabajo

Los principales objetivos para este trabajo de fin de master son los siguientes:

Objetivos a nivel de investigación y estudios:

- Realizar una revisión bibliográfica sobre la detección de anomalías para redes LAN y sus componentes.
- Estudiar las bondades que ofrecen los IDS open-source para detección de anomalías.
- Describir las funcionalidades de IDS con respecto a Elastic Stack.
- Analizar el IDS más propicio para integración con Elastic Stack.
- Investigar la generación de Active Response desde los IDS.

Objetivos a nivel de implantación y desarrollo:

- Instalar y configurar un entorno de laboratorio para simulación de un escenario propicio para pruebas del Elastic Stack.
- Incorporar al entorno planteado un IDS para detección de anomalías en host de la red.
- Configurar los Active Response en el IDS en caso de detección de anomalías.
- Realizar pruebas de funcionamiento para la puesta en marcha de la simulación final.

Objetivos a nivel académico/entrega:

- Desarrollar las entregas parciales y enviarlas en tiempo y forma.
- Desarrollar la memoria final del TFM.
- Generar un PPT y video que sintetice todo el proyecto.

1.3. Enfoque y método seguido

El presente proyecto se enfoca en el ámbito empresarial, el cual consiste en implementar el Elastic Stack con IDS, cuya finalidad es la de detectar anomalías

y generar alertas, de manera que permita crear Active Response para que el Firewall situado en la red cambie sus reglas en función de las advertencias. Por lo tanto, se divide el proyecto en dos partes, una parte teórica y otra práctica.

- La primera parte teórica es referente a la investigación sobre las herramientas a utilizar concerniente a la detección de anomalías. Además, se profundiza en conceptos de seguridad para establecer una base teórica fundamentada.
- La segunda parte consiste en seleccionar las herramientas más propicias analizadas en el estudio previo y realizar con ellas un análisis de la funcionalidad del IDS para detección de anomalías. Posteriormente, se debe identificar las alertas las cuales se adaptan para crear un Active Response y modificar las reglas de Firewall.

Se considera como la mejor estrategia para llegar a cumplir los objetivos planteados dividir la parte teórica y posterior la práctica, de esta manera se empieza por los objetivos relacionados con la investigación y estudio. Es propicio la adquisición de los conocimientos necesarios para la ejecución de la parte práctica con la integración de los distintos componentes en la red planteada en el laboratorio.

En cuanto a la metodología que permita llegar a cumplir con los objetivos [3], es la generalmente manejada para resolver problemas de ingeniería que consta de las siguientes fases:

- Identificación de problema
- Recopilación de Información
- Generar soluciones
- Analizar y seleccionar solución
- Prueba e implementación de solución

Consecuentemente se puede describir las actividades y resultados que se producen de inicio y fin en el proyecto.

1.4. Planificación del Trabajo

	Actividad	Categoría	Progreso	Inicio	Días
1	**Planificación**	Milestone	16/9/2020	29/9/2020	
1.1	Establecer problema a resolver	Goal	100%	16/9/2020	5
1.2	Definición de objetivos	Goal	100%	21/9/2020	3

1.3	Definir propuesta metodológica	Goal	100%	24/9/2020	2
1.4	Elaborar cronograma de trabajo	Goal	100%	26/9/2020	2
	Entrega de plan de trabajo	Goal	100%	28/9/2020	2
2.	**Investigación y Análisis**	Milestone	30/9/2020	27/10/2020	
2.1.	**Estudio de Elastic Stack**	Goal	100%	30/9/2020	2
2.1.1.	Elasticsearch	Goal	100%	2/10/2020	1
2.1.2.	Logstash	Goal	100%	3/10/2020	1
2.1.3.	Beats	Goal	100%	4/10/2020	1
2.1.4.	Kibana	Goal	100%	5/10/2020	1
2.2.	**Definición de IDS**	Goal	100%	6/10/2020	2
2.2.1.	HIDS	Goal	100%	8/10/2020	1
2.2.2.	NIDS	Goal	100%	9/10/2020	1
2.3.	**Estudio de dispositivos necesario para entorno de laboratorio**	Goal	100%	10/10/2020	2
2.3.1.	Dispositivos de red	Goal	100%	12/10/2020	1
2.3.2.	Servidores	Goal	100%	13/10/2020	1
2.4.	**Requisitos para detección de anomalías**	Goal	100%	14/10/2020	2
2.4.1.	Comparación de gestores de la seguridad	Goal	100%	16/10/2020	1
2.4.2.	Elección de gestor de la seguridad	Goal	100%	17/10/2020	1
2.5.	**Arquitectura de Elastic Stack integrado Wazuh**	Goal	100%	18/10/2020	2
2.5.1.	Servidor Wazuh	Goal	100%	20/10/2020	1
2.5.2.	Uso de Wazuh	Goal	100%	21/10/2020	1
2.5.3.	Mánager de Wazuh	Goal	100%	22/10/2020	1

2.5.4.	Agente de Wazuh	Goal	100%	23/10/2020	1
2.5.5.	Active Response	Goal	100%	24/10/2020	1
2.6.	**Telegram**	Goal	100%	25/10/2020	1
2.6.1.	Bots	Goal	100%	26/10/2020	1
	Entrega de plan de trabajo	Goal	100%	27/10/2020	1
3.	**Implementación y puesta en marcha**	Goal	28/10/2020	24/11/2020	
3.1.	**Requerimientos**	Goal	100%	28/10/2020	2
3.1.1.	Implementación y configuración de entorno de simulación	Goal	100%	30/10/2020	1
3.1.2.	Instalación y configuración de dispositivos de red	Goal	100%	31/10/2020	1
3.1.3.	Instalación y configuración de servicios en la red	Goal	100%	1/11/2020	1
3.2.	**Diseño/Adaptación**	Goal	100%	2/11/2020	3
3.2.1.	Instalación del Mánager de Wazuh	Goal	100%	5/11/2020	2
3.2.2.	Instalación del agente de Wazuh	Goal	100%	7/11/2020	2
3.2.3.	Instalación de Elastic Stack	Goal	100%	9/11/2020	2
3.2.4.	Configuración de Active Response para deteccion de ataques	Goal	100%	11/11/2020	2
3.2.5.	Configuración de Alertas por Telegram	Goal	100%	13/11/2020	2
3.3.	**Pruebas de Funcionamiento**	Goal	100%	15/11/2020	3
3.3.1.	Ejecución de ataque SQL injection	Goal	100%	18/11/2020	2
3.3.2.	Ejecución de ataque DoS	Goal	100%	20/11/2020	2
3.3.3.	Ejecucion de alertas	Goal	100%	22/11/2020	2
	Entrega de plan de trabajo	Goal	100%	24/11/2020	1

4.	**Presentación**	Goal	25/11/2020	29/12/2020	
4.1.	**Desarrollo de la memoria final del TFM**	Goal	100%	25/11/2020	25
4.2.	**Concluciones sobre el trabajo realizado**	Goal	100%	20/12/2020	4
4.3.	**Generación de PPT y video que sintetice todo el proyecto**	Goal	100%	24/12/2020	5
	Entrega de plan de trabajo	Goal	100%	29/12/2020	1

Tabla 1: Planificación temporal del proyecto

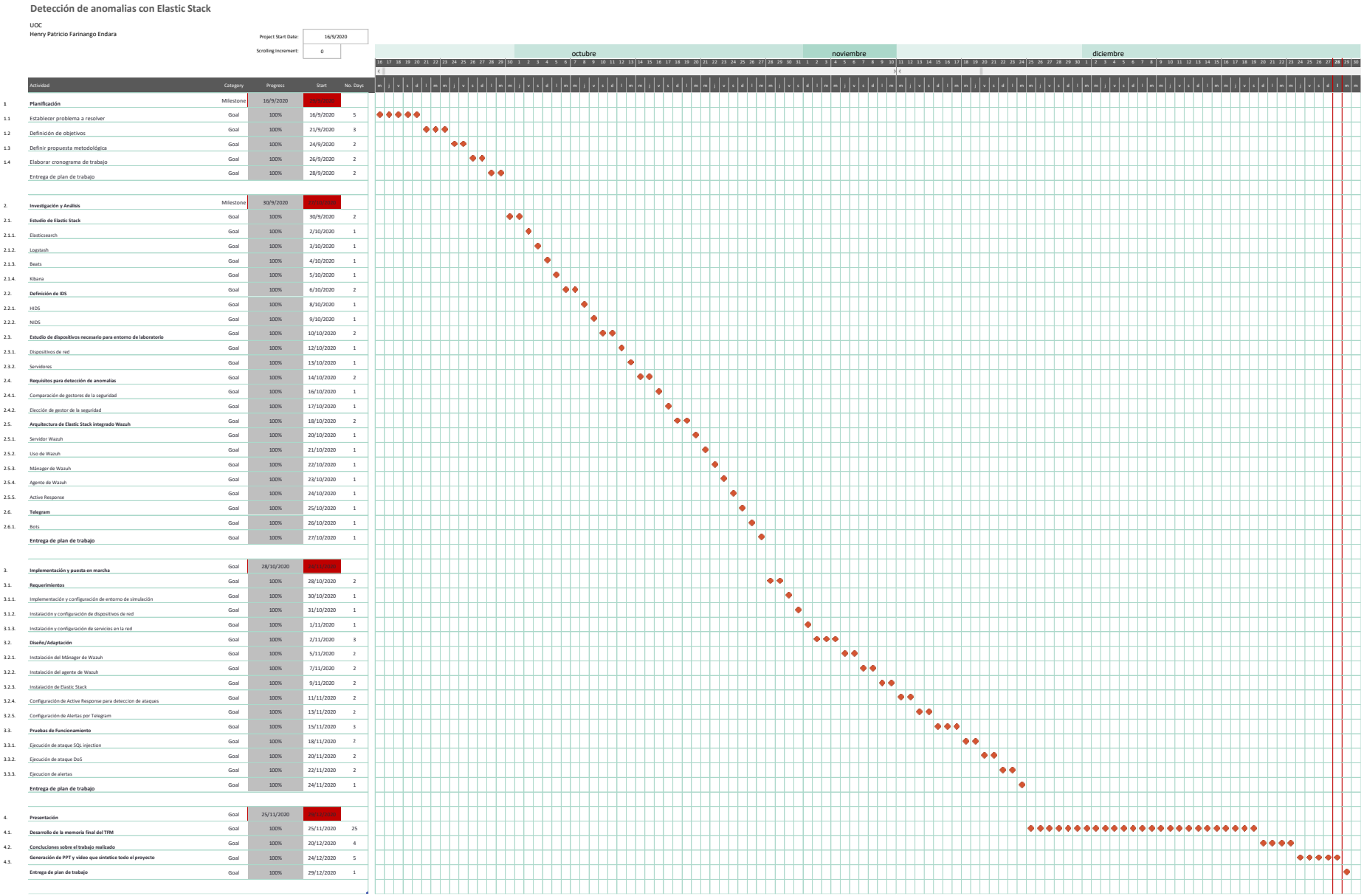

Tabla 2: Diagrama de Gantt con la planificación temporal

1.5. Recursos de software y hardware

Para este proyecto se hace una distinción entre recursos de software y hardware:

1.5.1. Recursos de software

Se utilizan los siguientes recursos de software:

1) Sistemas Operativos
a) Microsoft Windows 10 Home
b) GNU/Linux: CentOS 7
2) Virtualización: VirtualBox, GNS3
3) Edición y maquetación de memoria: Microsoft Office
4) Gestor de referencias bibliográficas: Mendeley 1.19.4

1.5.2. Recursos de hardware

Se utiliza los siguientes recursos de hardware, tomando en cuenta que se plantea un ambiente simulado:

Host: Asus ROG G53
Procesador: Intel(R) Core™ i7-9750h CPU @ 2.60GHz
RAM: 32,0 GB

1.6. Estado del Arte

La seguridad y privacidad de los servicios en línea tienen muchos desafíos y problemas que aún quedan por investigar. Los problemas de seguridad y privacidad siguen siendo un desafío importante. Las anomalías laborales pueden ocasionar graves problemas de privacidad y seguridad al proporcionar un servicio. En la era del rápido desarrollo tecnológico relacionado con la tecnología de la información, el número y la complejidad de las amenazas a la seguridad de los sistemas informáticos o las redes también están aumentando. Prevenir todas las amenazas a la seguridad se vuelve imposible debido a la constante adaptación de los piratas informáticos a las tecnologías emergentes de protección de la información. Las organizaciones que cuentan con sistemas de tecnología de la información (TI) toman las medidas necesarias para contrarrestar cualquier amenaza a los recursos informáticos y de red de manera eficiente [4].

Se puede destacar que para entrar en más profundidad en un análisis de eventos en cuanto a la seguridad se tiene herramientas como los HIDS (Sistemas de Deteccion de Intrusiones de Host) y los NIDS (Sistemas de Detección de

Intrusiones en la Red). Estas se caracterizan por el tipo de servicio que dan, así también el tipo de análisis de seguridad que brindan, alcance, etc [5]. Por lo tanto, estos se definen de la siguiente manera:

- HIDS: Se basa en la detección en un host únicamente, analizando los registros, es decir, la deteccion de intrusos a nivel de equipo.

- NIDS: Se destaca por la detección del tráfico de la red en la que están conectados los hosts; de esta manera se puede decir que realiza la deteccion de intrusos a nivel de red.

Se puede diferenciar en que los HIDS pueden detectar en el host los eventos y generar alertas al igual que los NIDS pero en adicional es capaz de examinar el flujo de datos entre host y servidor, por ello no se aplican técnicas para evadir NIDS y así mismo esta comunicación en caso de ser encriptada aún puede ser monitorizada por los HIDS ya que estos inspeccionan el tráfico antes de que sea encriptado [6].

1.6.1. Sistema de detección de intrusos (IDS)

A continuación se presentan algunas de las soluciones más conocidas de IDS:

1.6.1.1. Snort

Es un NIDS de código abierto que Puede realizar análisis de tráfico en tiempo real y registro de paquetes en redes IP. Puede analizar los protocolos y buscar contenido coincidente. También, se puede utilizar para detectar varios ataques, como desbordamientos de búfer, escaneos de puertos sigilosos, etc. En el caso de [7] se toma como solución para análisis para deteccion de intrusiones en la red y genera una alerta por detección. Snort consta principalmente de cuatro componentes: rastreadores de datos, preprocesador, motor de detección y sistema de registro y alarma [8].

1.6.1.2. Suricata

Es un IDS de los más recientes y está diseñado para funcionar con los conjuntos de reglas de Snort. Suricata fue desarrollado para ser un "motor IDS de próxima generación" con capacidades IPS (Sistema de Prevención de Intrusiones) diseñadas para ser retrocompatible con los conjuntos de reglas de Snort. Suricata fue diseñado como un sistema multiproceso, lo que le permite aprovechar múltiples núcleos. En una máquina de un solo núcleo, se ha demostrado que Snort supera a Suricata [9].

1.6.1.3. Wazuh

Wazuh es una plataforma gratuita y de código abierto para la detección de amenazas, el control de la seguridad, la respuesta a incidentes y el cumplimiento normativo. Se puede utilizar para monitorear puntos finales, servicios en la nube y contenedores, y para agregar y analizar datos de fuentes externas [10].

1.6.1.4. OSSEC

Es una herramienta de código abierto que permite la detección de intrusos basado en host. Los operadores pueden introducirse en una variedad de framework; Servidores web, servidores de correo, servidores VMWare [11]. OSSEC ofrece la capacidad de realizar análisis de registros de forma centralizada, agregar o eliminar agentes y grupos, borrar la caché y obtener información sobre decodificadores y reglas. Por esta razón en [12] se hace su implantación para una compañía que requiere aplicar un análisis de los eventos para sus servidores y poder administrar la seguridad.

1.6.2. Gestión de seguridad de la información

A continuación se presentan algunas de las soluciones más conocidas y utilizadas para gestión de la información de seguridad:

1.6.2.1. Nagios Log Server

Nagios Log Server simplifica enormemente el proceso de búsqueda de sus datos de registro. Configure alertas para notificarle cuando surjan amenazas potenciales, o simplemente consulte sus datos de registro para auditar rápidamente cualquier sistema. Con Nagios Log Server, obtiene todos sus datos de registro en una ubicación, con alta disponibilidad y conmutación por error incorporada [13]. La supervisión del servidor se realiza mediante Nagios, que es una herramienta de código abierto es utilizada en [14], donde se agregan varios nodos a la red y se monitorea el rendimiento y el estado de la red. Se recopilan los datos y se proporcionan estadísticas en tiempo real y se analiza el rendimiento de la red.

1.6.2.2. Graylog

Graylog captura, almacena y habilita de forma centralizada la búsqueda en tiempo real y el análisis de registros contra terabytes de datos de máquina de cualquier componente en la infraestructura de TI y aplicaciones. El software utiliza una arquitectura de tres niveles y almacenamiento escalable basado en Elasticsearch y MongoDB [15]. En esta investigación [16] se centraliza y analiza

eventos de seguridad con esta herramienta y proporciona una interfaz de usuario para consulta de eventos de host.

1.6.2.3. Elastic Stack

La pila ELK consta de varios componentes de código abierto (ElasticSearch, Logstash y Kibana) y se puede ejecutar en un entorno de hardware virtual [5]. ElasticSearch es un motor de búsqueda basado en Apache Lucene. Logstash es un componente de canalización de recopilación de datos dinámicos para recopilar los registros necesarios y enviarlos al componente ElasticSearch después de la transformación al formato JSON. Kibana es un componente para visualización en varios tipos, como gráficos, tablas y mapas, etc [17]. En [18] se evalúa el rendimiento del Elastic Stack frente a soluciones de comerciales.

2. Investigación y Análisis

En este apartado se realiza una revisión bibliográfica sobre soluciones que permiten las deteccion de anomalías en redes empresariales de area local (LAN).

2.1. Estudio de Elastic Stack

Elastic Stack es un rico ecosistema de componentes que actúa como una pila completa de búsqueda y análisis. Los componentes principales de Elastic Stack son Kibana, Logstash, Beats, X-Pack y Elasticsearch. Elasticsearch está en el corazón de Elastic Stack, proporcionando capacidades de almacenamiento, búsqueda y análisis. Kibana, que también se denomina ventana de Elastic Stack, es una excelente visualización e interfaz de usuario para Elastic Stack. Logstash y Beats ayudan a introducir los datos en Elastic Stack. X-Pack ofrece potentes funciones que incluyen supervisión, alertas y seguridad para que su sistema esté listo para la producción. Dado que Elasticsearch está en el corazón de Elastic Stack, cubriremos la pila de adentro hacia afuera, comenzando desde el corazón y pasando a los componentes circundantes [19].

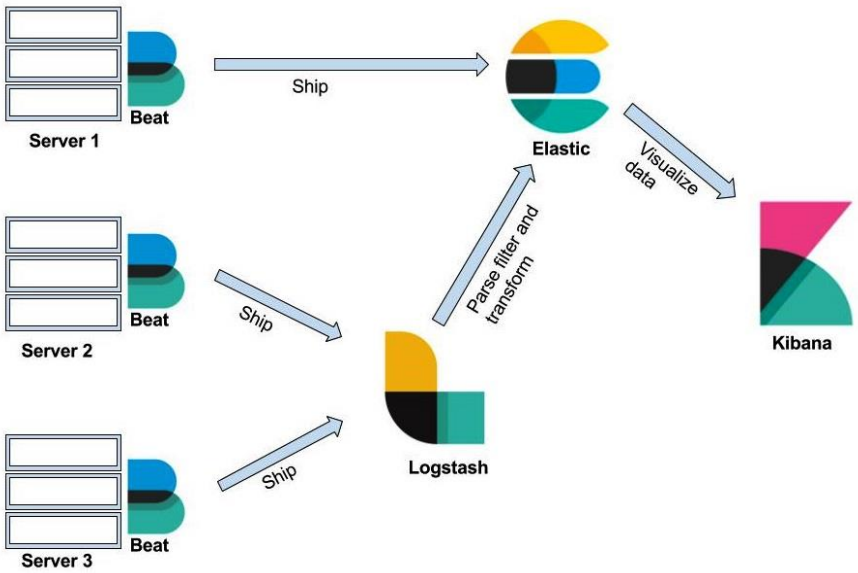

Figure 1. Gestión de registros y métricas del sistema con Elastic Stack
Fuente: [20]

2.1.1. Elasticsearch

Elasticsearch es un motor de búsqueda y análisis y se basa en documentos, lo que significa que las entradas se almacenan como documentos JSON y todos los campos se pueden indexar y utilizar en una sola consulta. ES también permite la búsqueda de texto completo en datos no estructurados, por lo que se puede utilizar para crear muchas soluciones para Big Data porque no depende del tipo de fuente de datos: estructurada, semiestructurada o no estructurada. Aunque generalmente se usa para registros y monitoreo, ES no está completamente explotado y se puede usar para muchas otras cosas, como: interactuar con cualquier API RESTful usando JSON sobre http [21].

2.1.2. Logstash

Logstash ayuda a centralizar datos de eventos como registros, métricas o cualquier otro dato en cualquier formato. Puede realizar una serie de transformaciones antes de enviarlo a un escondite de su elección. Es un componente clave de Elastic Stack, que se utiliza para centralizar los procesos de recopilación y transformación en su canalización de datos [19].

En [20] define las siguientes características clave :

- **Procesamiento de datos centralizado:** Logstash ayuda a construir una canalización de datos que puede centralizar el procesamiento de datos. Con el uso de una variedad de complementos para entrada y salida, puede convertir muchas fuentes de entrada diferentes a un solo formato común.

- **Soporte para formatos de registro personalizados:** Los registros escritos por diferentes aplicaciones a menudo tienen formatos específicos de la

aplicación. Logstash ayuda a analizar y procesar formatos personalizados a gran escala. Proporciona soporte para escribir sus propios filtros para la tokenización y también proporciona filtros listos para usar.

- **Desarrollo de complementos:** se pueden desarrollar y publicar complementos personalizados, y ya hay una gran variedad de complementos desarrollados a medida disponibles.

2.1.3. Beats

Beats son remitentes de datos livianos y de un solo propósito que usamos para obtener datos de diferentes servidores. Beats se puede instalar en los servidores como un agente ligero para enviar métricas del sistema o procesar o archivar datos a Logstash o Elasticsearch. Recopilan datos de la máquina en la que están instalados y luego los envían a Logstash, que usamos para analizar o transformar los datos antes de enviarlos a Elasticsearch, o podemos enviar los datos de Beats directamente a Elasticsearch [19].

Hay muchos componentes Beat que ya han sido creados por el equipo de Elastic y la comunidad de código abierto. El equipo de Elastic ha creado Beats, incluidos Packetbeat, Filebeat, Metricbeat, Winlogbeat, Audiobeat y Heartbeat [22].

2.1.4. Kibana

Kibana es una herramienta de panel que es fácil de usar y trabaja en estrecha colaboración con Elasticsearch. Podemos usar Kibana para diferentes casos de uso, como el monitoreo del sistema y el monitoreo de aplicaciones. Kibana no es solo una herramienta de visualización, también crea un ecosistema de monitoreo completo cuando aprovechamos el poder de Elastic Stack [22].

En [20] se precisan algunas de las características clave de Kibana son las siguientes:

- Proporciona análisis flexible y una plataforma de visualización para inteligencia empresarial.
- Proporciona capacidades de análisis, resumen, creación de gráficos y depuración en tiempo real.
- Proporciona una interfaz intuitiva y fácil de usar, que es altamente personalizable mediante algunas funciones de arrastrar y soltar y alineaciones cuando sea necesario.
- Permite guardar el tablero y administrar más de un tablero. Los paneles se pueden compartir e incrustar fácilmente en diferentes sistemas.
- Permite compartir instantáneas de registros en los que ya ha buscado y aísla múltiples transacciones problemáticas.

2.2. Definición de IDS

Un sistema de detección de intrusiones (IDS) es una combinación de hardware/software o una combinación de hardware y software que detecta las intrusiones en un sistema o red. IDS complementa un firewall al proporcionar una inspección exhaustiva tanto del encabezado de los paquetes como de su contenido, protegiendo así contra ataques, que de otra manera serían percibidos por un firewall como tráfico de red aparentemente benigno.

2.2.1. HIDS

El sistema de detección de intrusiones basado en host se refiere a la detección de intrusiones en un solo sistema. Normalmente, se trata de una implementación basada en software en la que un agente, como se muestra en la Figure 2. IDS basado en host, se instala en el host local que monitorea e informa la actividad de la aplicación. HIDS monitorea el acceso al sistema y su aplicación y envía alertas de cualquier actividad inusual. Supervisa constantemente los registros de eventos, registros del sistema, registros de aplicaciones, aplicación de políticas de usuario, detección de rootkits, integridad de archivos y otras intrusiones en el sistema. Supervisa constantemente estos registros y crea una línea de base. Si aparecen nuevas entradas de registro, HIDS compara los datos con la línea de base y si se encuentran entradas fuera de esta línea de base, HIDS activa una alerta. Si se detecta alguna actividad no autorizada, HIDS puede alertar al usuario o bloquear la actividad o tomar cualquier otra decisión basada en la política configurada en el sistema [23].

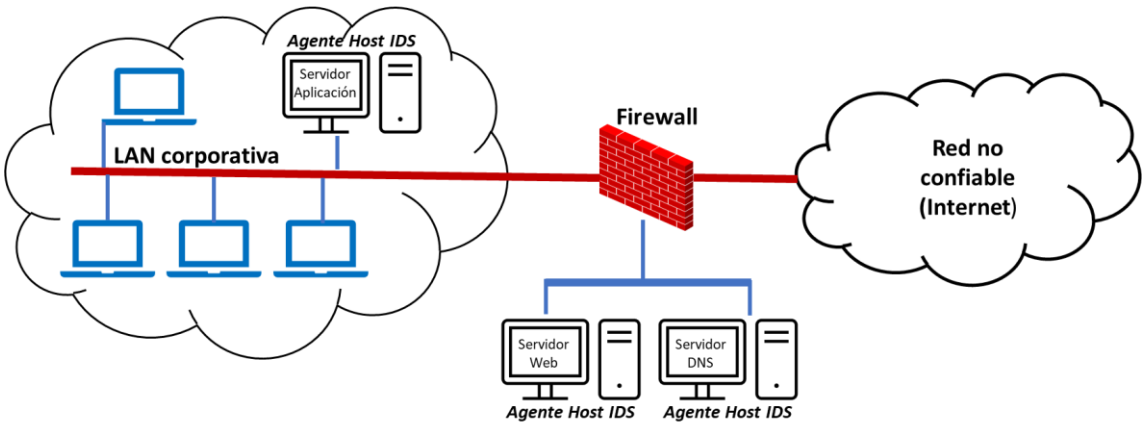

Figure 2. IDS basado en host
Fuente: Autor

HIDS generalmente se instala en servidores o dispositivos de punto final para proteger el sistema de intrusiones. La función de HIDS depende únicamente de las pistas de auditoría generadas por el sistema. Si los piratas informáticos logran desactivar estos registros, incluso si tiene un agente HIDS en ejecución, es posible que no active ninguna alerta. Ésta es la mayor desventaja de HIDS.

Las ventajas de HIDS son:

- Protección a nivel del sistema. Protege de ataques dirigidos al sistema.
- Cualquier actividad no autorizada en el sistema (cambios de configuración, cambios de archivo, cambios de registro, etc.) se detecta y se genera una alerta para acciones adicionales.

También hay desventajas:

- La funcionalidad HIDS funciona solo si los sistemas generan registros y coinciden contra las políticas predefinidas. Si por alguna razón, los sistemas no generan registros, es posible que HIDS no funcione correctamente.
- Si los piratas informáticos desactivan el servidor HIDS, entonces HIDS no sirve de nada. Esto es cierto para cualquier software de protección contra vulnerabilidades.

2.2.2. NIDS

Un sistema de detección de intrusiones basado en red (NIDS) 1 monitorea (y detecta) cualquier actividad sospechosa en una red. Verifica todos y cada uno de los paquetes que ingresan a la red para asegurarse de que no contengan ningún contenido malicioso que pueda dañar la red o el sistema final. El sistema de detección de intrusiones en la red detecta el tráfico de la red de forma continua. El tráfico se compara con perfiles de firmas conocidas y, si se encuentran anomalías en el tráfico, un NIDS activa una alarma en la consola de administración. Un solo sensor, como se muestra en la Figure 3, implementado en modo promiscuo o en modo en línea puede monitorear / proteger varios hosts en la red [23].

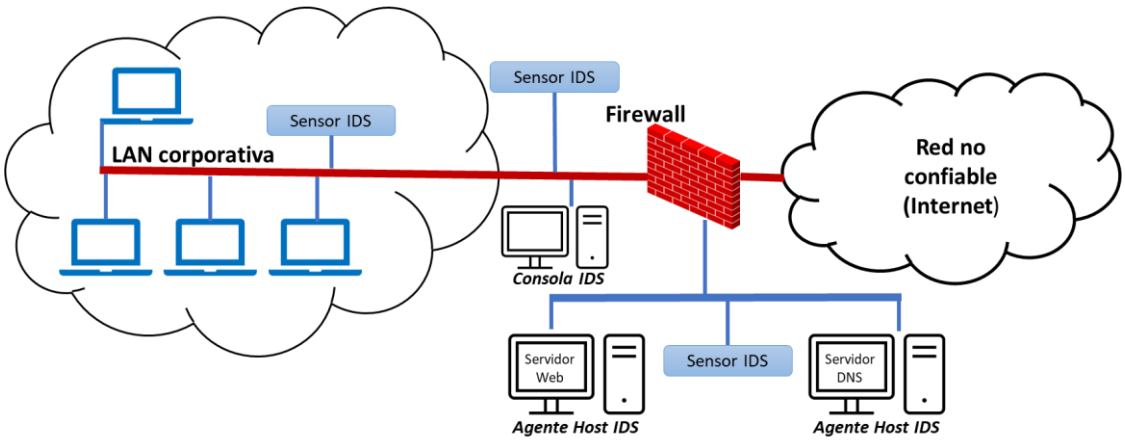

Figure 3. IDS basado en redes
Fuente: Autor

Los NIDS protegen la red y sus recursos desde la perspectiva de la red. Pueden detectar ataques de reconocimiento, ataques de denegación de servicio directamente a nivel de red. NIDS genera alertas tan pronto como descubre estos ataques. NIDS es una solución de hardware / software ubicada cerca del firewall

como un dispositivo independiente (sensor) y tiene un sistema operativo de red (pila TCP / IP). Los sensores tienen interfaces para monitorear la red (interfaz de monitoreo) y una interfaz de administración que se utiliza para controlar y recibir alertas y para enviar estas alertas al controlador de administración central [23].

2.3. Estudio de dispositivos necesario para entorno de laboratorio

Para desarrollo de este proyecto se plantea un escenario de laboratorio el cual permita la simulación de un entorno empresarial y que se pueda implementar el Elastic Stack para detección de anomalías.

Para poder llevar a cabo lo planteado es necesario hacer un estudio de los dispositivos para la simulación y por esta razón se platea un entorno empresarial genérico con implementación de Elastic Stack en la Figure 4.

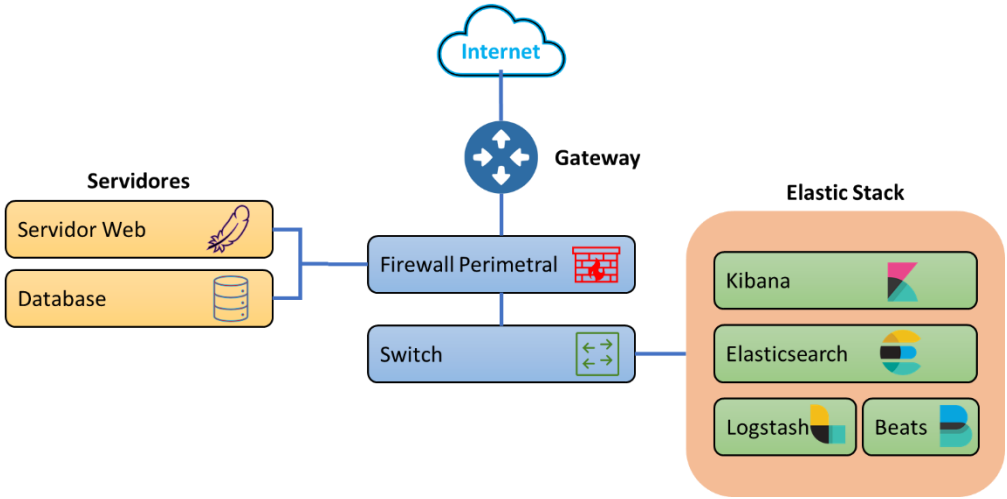

Figure 4. Red empresarial y Elastic Stack
Fuente: Autor

2.3.1. Dispositivos de red

Las redes informáticas son la base a muchas tareas diarias de negocios, comercio electrónico, gobierno electrónico, educación, entre otros. La complejidad y el despliegue de las redes informáticas sigue creciendo, ya que constantemente surgen nuevos protocolos, arquitecturas y aplicaciones, y las redes informáticas están en continuo proceso de adaptación. Por esta razón es importante conocer los dispositivos que mantienen en funcionamientos las compañías.

2.3.1.1. Router

La solución para resolver los problemas de comunicación de la capa 3 (o superior a la capa 3) dependerá de un enrutador de red (Router). Un enrutador de red une diferentes tipos de redes. Cuando dirige el tráfico de red, se basa en

direcciones IP o lógicas de paquetes en lugar de direcciones MAC o físicas. Como resultado, un enrutador puede diferenciar diferentes tipos de red.

El principio de un enrutador es leer todas las direcciones lógicas de paquetes de datos entrantes y luego, basándose en su propia tabla de enrutamiento, reenviar estos paquetes de datos entrantes a sus destinos. Debido a que el enrutador no solo puede leer direcciones IP, sino que también tiene la capacidad informática para admitir el enrutamiento de paquetes entrantes, es capaz de encontrar la ruta más corta en la red hacia el destino del paquete. En este sentido, un enrutador puede ser una computadora estándar o convencional que está especialmente configurada para propósitos de enrutamiento. Normalmente, un enrutador sería más lento que un Switch de capa 2 (según la ID del puerto) o un puente (según la MAC) porque no solo se necesita tiempo para leer la dirección IP de un paquete, sino que también la interpreta [24].

2.3.1.2. Switch

Un conmutador o Switch de red es uno de los componentes clave para conectar muchos dispositivos de red diferentes, como clientes, servidores host, impresoras, matrices de almacenamiento y bibliotecas de cintas. Forma una red de área extensa y una red de área extensa (LAN/WAN) para compartir información y recursos informáticos.

La función de un conmutador de red es similar a un concentrador; une todos los dispositivos informáticos para que puedan comunicarse entre sí. Sin embargo, un Switch hace el trabajo de una manera diferente. En lugar de inundar todos los paquetes en cada puerto, un conmutador puede leer la dirección de una trama de datos desde un puerto de origen y dedicarla directamente al puerto de destino conectado con el dispositivo de destino. El conmutador establecerá un canal lógico para la transmisión del flujo de datos entre esta fuente y los puertos de destino. En otras palabras, esta transmisión no interferirá con otros puertos. Si otras computadoras están intentando transmitir sus paquetes, pueden hacerlo a través de puertos no utilizados. Un conmutador permite que muchas computadoras conectadas transmitan muchos flujos de datos al mismo tiempo [24].

2.3.2. Servidores

Cuando se usa en referencia al hardware, un servidor es cualquier computadora que ejecuta un programa de servidor, que puede incluir, todas las configuraciones y sistemas operativos. Los servidores y clientes utilizan protocolos de comunicación para intercambiar información para realizar tareas, necesitan funcionar durante largos períodos de tiempo sin interrupciones, deben ser duraderos, fiables y tener fuentes de alimentación ininterrumpidas. Existen variaciones de servidor a servidor y cliente-servidor [25].

La mayoría de la gente está familiarizada con el uso de un navegador web. Incluso cuando utilizan un navegador nuevo por primera vez, son capaces de visitar una página web y navegar a otros sitios inmediatamente. Esto se debe a que los navegadores suelen incluir las mismas funciones prácticas. Los servidores web, por otro lado, rara vez se parecen entre sí. Sin embargo, los servidores comparten algunos puntos en común [26].

2.3.2.1. Apache Web Server

El servidor web Apache es uno de los servidores web más populares en la actualidad y ampliamente utilizado durante la última década lo confirma [27]. Es utilizado por aproximadamente el 50% de todos los sitios web. Apache es multiplataforma, ligero, robusto y se utiliza tanto en pequeñas empresas como en grandes corporaciones. Apache también es gratuito y de código abierto. Esto se debe a la facilidad de uso de las funciones administrativas y a su flexibilidad basada en su diseño modular [26].

2.3.2.2. MySQL Database

Con PHP puede permitir que el servidor web interactúe con cualquier otro programa. Una de las aplicaciones más utilizadas para interactuar con un servidor web es un servidor de base de datos. Con una conexión de base de datos al servidor web, su sitio puede buscar los datos del sistema de base de datos y así ofrecer contenido dinámico. Existe una gran lista de sistemas de bases de datos para elegir. Para propósitos de este proyecto se usa uno de los más comunes, MySQL, que es una base de datos relacional que utiliza el lenguaje de consulta estructurado (SQL). SQL es el conjunto de instrucciones más extendido que se utiliza para configurar y consultar un sistema de base de datos.

MySQL puede ejecutarse en hardware muy modesto y ejerce muy poca presión sobre los recursos del sistema; muchos usuarios pequeños brindan información a sus organizaciones ejecutando MySQL en modestos sistemas de escritorio. La velocidad con la que puede recuperar información la ha convertido en una de las favoritas de los administradores web.

Se puede instalar sin mucha configuración complicada y sofisticada. Ahora que muchas distribuciones de Linux incluyen MySQL, la instalación puede ser casi automática.

MySQL tiene un enfoque de licencia dual: si desea construir su propio producto a partir de él, paga a MySQL AB una tarifa de licencia. Si solo desea utilizar MySQL para proporcionar sus propios datos, no tiene que pagar la tarifa de licencia. MySQL también ofrece soporte técnico, al igual que muchas otras empresas y consultores, algunos de ellos probablemente cerca de usted [28].

2.4. Requisitos para detección de anomalías

Para el proceso de deteccion de anomalías en este proyecto, se necesita realizar una simulación de un entorno empresarial en base a su deteccion, donde se pueda tener una respuesta activa a los casos estudiados en el apartado 3.2, de esta manera se consideran los siguientes requisitos:

- Permita su implementación sin limitantes y no tenga costo por ello.
- Pueda implantarse en un entorno de práctica desde un ordenador, soporte virtualización y funcional para implantar en contenedores en caso de requerirlo.
- Permita monitorizar la mayor parte de dispositivos de red y sistemas operativos.
- Permita aspectos de seguridad, evite los accesos anónimos y cuente con características de cifrado de datos.
- Permita alta disponibilidad con procesos automáticos.
- Que permita deteccion de ataques y uso de plugins para mejorar su rendimiento en cuanto a seguridad.
- Proporcione alertas y una rápida respuesta.

En base a estos requerimientos se hace una comparación de gestores de la seguridad expuestos en el apartado 1.6.2.

2.4.1. Comparación de gestores de la seguridad

En este espacio se realiza una comparación en cuanto a las principales características que se consideran en este proyecto para tomar en cuenta un gestor de la seguridad. Como parte fundamental hay que destacar que este proyecto busca una herramienta que es adecuada en cuanto a: Sistema Operativo, licencia, alertas, recolección de datos, filtrado, reportes. Por lo tanto se desarrolla una tabla comparativa Tabla 3:

	Elastic Stack	Nagios Log Server	Graylog
Sistemas Operativo	Windows, Red Hat y Todos Linux	Red Hat Enterprise Linux CentOS	Windows, Linux, Ubuntu y CentOS
Licencia	Open source Comercial: Cloud, Emterprise y Free Trial	Comercial: Free Trial Desde $1,995.00	Open Source Enterprise: Free for Under 5GB/día
Alertas	Compatible a través de X-pack / se pueden crear configuraciones de observador Integraciones disponibles para enviar alertas a herramientas AIOps (Inteligencia artificial para operaciones de TI)	Los usuarios pueden configurar umbrales según el tipo de registro y luego realizar algunas tareas diferentes en consecuencia. Existe una opción para enviar un correo electrónico a los usuarios cuando se activa una alerta, enviar una trampa SNMP, ejecutar scripts personalizados o reenviar la alerta a otro software de Nagios.	Característica incorporada Alertas generadas según la transmisión Integraciones disponibles para enviar alertas a herramientas AIOps
Recolección de datos	Nodos de ingesta de Elasticsearch, Beats, Logstash	Nagios Log Server está diseñado para recibir registros generados desde diferentes sistemas operativos y dispositivos como Windows, Linux, servidores de correo, servidores web, aplicaciones, servidores SQL y más.	Entradas de mensajes y paquetes de contenido (entradas, extractores, flujos de salida, configuración del tablero) Carro lateral Graylog
Filtro de registro Filtrado de búsqueda	Complemento de filtro Grok. Filtro de búsqueda con niveles de campos, búsquedas guardadas, gráficos, etc.	Sus datos se almacenan en una ubicación y se pueden buscar con múltiples consultas y filtros, lo que le permite profundizar rápidamente en el problema exacto que está buscando.	Utiliza reglas de Drools, extractores, filtros de campos, listas negras
Reporte Gestión de datos históricos	Disponible en componente X-Pack Genere rápidamente informes de cualquier visualización o panel de Kibana y cualquier dato sin procesar. Cada informe está optimizado para impresión, personalizable y en formato PDF.	Genere una multitud de informes, incluidos gráficos circulares, histogramas, listas y más. Los tipos de informes se pueden personalizar por completo mediante las opciones de filtrado anteriores y se pueden organizar según las preferencias del usuario en un panel. Estos informes se pueden configurar para que se actualicen automáticamente, creando un informe de estado en vivo.	Sin capacidad de generación de informes incorporada La API REST se puede utilizar para generar informes propios sobre los datos históricos y los datos de transmisión.

Tabla 3. Comparativa de gestores de la seguridad
Fuente: [29] [30] [19][14] [16]

2.4.2. Elección de gestor de la seguridad

En la Tabla 3 se exponen características importantes en donde se determina que Elastic Stack es el más idóneo por su amplia compatibilidad con plataformas/software y uso de plugin que lo mejoran en capacidad de monitoreo y análisis. De esta manera se presentan las ventajas y desventajas de Elastic Stack:

Elastic Stack	
Ventajas	**Desventajas**
Tanto el análisis como el seguimiento mediante visualización se pueden realizar con la misma herramienta. Auto personalización y alertas según la gravedad. Función de aprendizaje automático incorporada. Puede revisar a diario y almacenar para uso futuro. No ocupa mucho espacio durante la instalación a pesar de que son tres herramientas diferentes combinadas.	La instalación puede requerir mucho tiempo, con muchos paquetes adicionales para instalar La sangría en los archivos de configuración debe hacerse con cuidado, cualquier error en el monitor no crea un índice El firewall y los permisos de la red deben estar bien definidos o, de lo contrario, el panel de Kibana no funcionará. Actualizar si se debe realizar manualmente El mantenimiento de tres módulos diferentes es realmente difícil, si ocurre algún error para averiguar qué módulo ha causado, puede ser bastante tedioso.

Tabla 4. Ventajas y desventajas de Elastic Stack
Fuente: [31]

La razón por la que se elige Elastic Stack es por:

- Es un gestor de seguridad open source que únicamente requiere de pago para soporte y monitoreo en la nube.
- No está limitada en su versión y posee todas las características para implantar soluciones on-primise y por lo tanto para escenarios prácticos.
- Documentación online actualizada y muy completa.
- Posibilita la integración de IDS como Wazuh, Suricata, entre otros.
- Sobre todo, cumple todos los requisitos indicados en la sección 2.4.

2.5. Arquitectura de Elastic Stack integrado Wazuh

Se plantea la arquitectura de red con la integración de Wazuh, ya que al hacer la elección de Elastic Stack como el gestor idóneo para este proyecto, se complementa con Wazuh para detección de anomalías como un HIDS que mediante un plugin se integra a la pila [32]. A continuación en la se expone una arquitectura donde se integra a Elastic Stack y Wazuh.

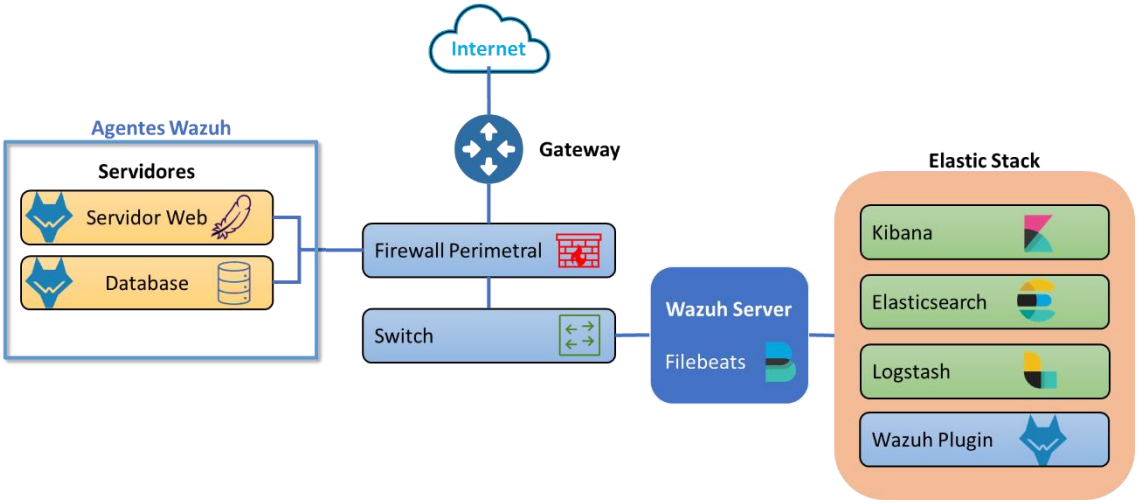

Figure 5. Arquitectura de Wazuh centralizada
Fuente: Adaptado de [32]

2.5.1. Servidor Wazuh

En este proyecto se considera un HIDS que permite obtener alertas para dar respuesta inmediata con Active Response, comentado en el apartado 2.4.2 se considera a Wazuh como herramienta para detección de anomalías. Wazuh es una solución de monitoreo de seguridad gratuita, de código abierto y lista para la empresa para detección de amenazas, monitoreo de integridad, respuesta a incidente es posible integrar a Elastic Stack [33].

2.5.2. Uso de Wazuh

Se utiliza Wazuh como un sistema de detección de intrusiones de host (HIDS) en cada uno de los nodos de la red de simulación.

Una breve presentación de algunos de los casos de uso más comunes de la solución Wazuh [5][34]:

- Detección de intrusión

    - Los agentes de Wazuh escanean los sistemas monitoreados en busca de malware, rootkits y anomalías sospechosas. Pueden detectar archivos ocultos, procesos encubiertos o escuchas de red no registrados, así como inconsistencias en las respuestas a las llamadas del sistema.

- Además de las capacidades del agente, el componente del servidor utiliza un enfoque basado en firmas para la detección de intrusiones, utilizando su motor de expresión regular para analizar los datos de registro recopilados y buscar indicadores de compromiso.

- Análisis de datos de registro

  - Los agentes de Wazuh leen los registros del sistema operativo y de las aplicaciones y los reenvían de forma segura a un administrador central para el análisis y el almacenamiento basados en reglas. Cuando no se implementa ningún agente, el servidor también puede recibir datos a través de syslog desde dispositivos o aplicaciones de red.

  - Las reglas de Wazuh le ayudan a conocer los errores del sistema o de la aplicación, las configuraciones incorrectas, las actividades maliciosas intentadas y / o exitosas, las violaciones de políticas y una variedad de otros problemas operativos y de seguridad.

- Supervisión de la integridad de los archivos

  - Wazuh monitorea el sistema de archivos e identifica cambios en el contenido, los permisos, la propiedad y los atributos de los archivos que debe vigilar. Además, identifica de forma nativa a los usuarios y las aplicaciones que se utilizan para crear o modificar archivos.

  - Las capacidades de monitoreo de la integridad de los archivos se pueden usar en combinación con la inteligencia de amenazas para identificar amenazas o hosts comprometidos.

- Detección de vulnerabilidades

  - Los agentes de Wazuh extraen datos del inventario de software y envían esta información al servidor, donde se correlaciona con las bases de datos CVE (Common Vulnerabilities and Exposure) continuamente actualizadas, para identificar software vulnerable conocido.

  - La evaluación automatizada de vulnerabilidades lo ayuda a encontrar los puntos débiles de activos críticos y tomar medidas correctivas antes de que los atacantes los exploten para sabotear negocios o robar datos confidenciales.

- Evaluación de la configuración

  - Wazuh supervisa los ajustes de configuración del sistema y las aplicaciones para asegurarse de que cumplan con sus políticas de seguridad, estándares y / o guías de refuerzo. Los agentes realizan exploraciones periódicas para detectar aplicaciones que se sabe que son vulnerables, no están parcheadas o configuradas de forma insegura.

- Además, las comprobaciones de configuración se pueden personalizar, adaptándolas para alinearlas correctamente con la organización. Las alertas incluyen recomendaciones para una mejor configuración, referencias y mapeo con cumplimiento normativo.

- Respuesta al incidente

  - Wazuh proporciona respuestas activas listas para usar para realizar varias contramedidas para abordar las amenazas activas, como bloquear el acceso a un sistema desde la fuente de la amenaza cuando se cumplen ciertos criterios.

  - Además, Wazuh se puede utilizar para ejecutar de forma remota comandos o consultas del sistema, identificando indicadores de compromiso (IOC) y ayudando a realizar otras tareas forenses en vivo o de respuesta a incidentes.

2.5.3. Mánager de Wazuh

El administrador de Wazuh es el sistema que analiza los datos recibidos de todos los agentes registrados que activan alertas cuando un evento coincide con una regla. También funciona como agente en la máquina local, por lo que tiene todas las características que tiene un agente. Además, el mánager puede reenviar las alertas que desencadena a través de syslog, correos electrónicos o API externas integradas.

El mánager de Wazuh utiliza fuentes de inteligencia de amenazas para mejorar sus capacidades de detección. También hace uso de los requisitos de cumplimiento normativo (por ejemplo, PCI DSS, HIPAA, NIST 800-53...) y el marco Mitre ATT&CK para enriquecer los datos de alertas, proporcionando contexto útil alrededor de ellos. Además, el servidor Wazuh se puede integrar con software externo y plataformas de mensajería instantánea [35].

2.5.3.1. Arquitectura del mánager

El servidor Wazuh ejecuta el motor de análisis, la API RESTful de Wazuh, el servicio de registro de agentes, el servicio de conexión de agentes, el demonio de clúster de Wazuh y Filebeat. El diagrama siguiente representa la arquitectura y los componentes del servidor:

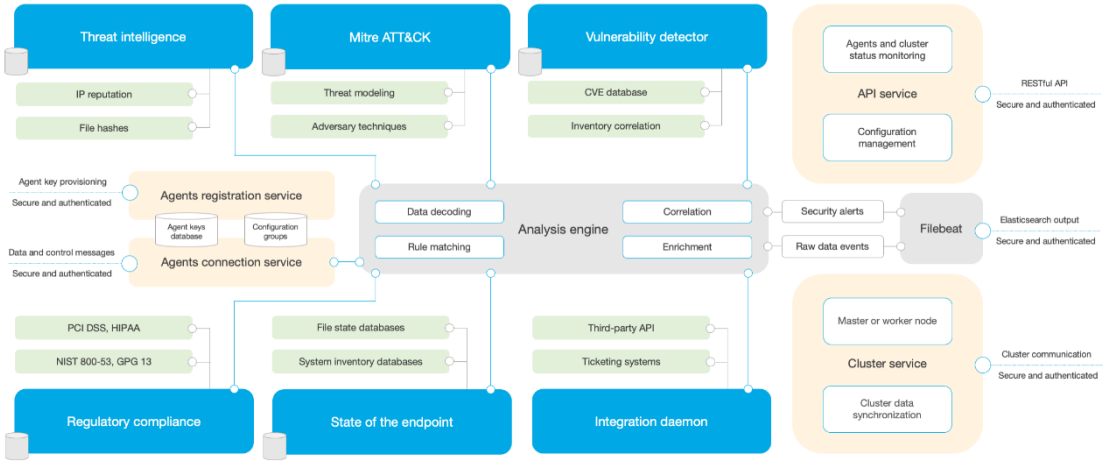

Figure 6. Arquitectura del mánager de Wazuh.
Fuente: [35]

2.5.4. Agente de Wazuh

El agente de Wazuh se ejecuta en Linux, Windows, macOS, Solaris, AIX y otros sistemas operativos. Se puede implementar en equipos portátiles, escritorios, servidores, instancias en la nube, contenedores o máquinas virtuales. Proporciona capacidades de prevención, detección y respuesta de amenazas. También se utiliza para recopilar diferentes tipos de datos del sistema y de la aplicación, que se remite al mánager de Wazuh a través de un canal cifrado y autenticado [35].

2.5.4.1. Arquitectura de agentes

El agente Wazuh tiene una arquitectura modular, donde diferentes componentes se encargan de sus propias tareas: monitoreo del sistema de archivos, lectura de mensajes de registro, recopilación de datos de inventario, configuración del sistema de escaneo, búsqueda de malware, etc. Los usuarios pueden habilitar o deshabilitar los módulos del agente a través de los ajustes de configuración, adaptando la solución a sus casos de uso particulares [35].

El diagrama siguiente representa la arquitectura y los componentes del agente:

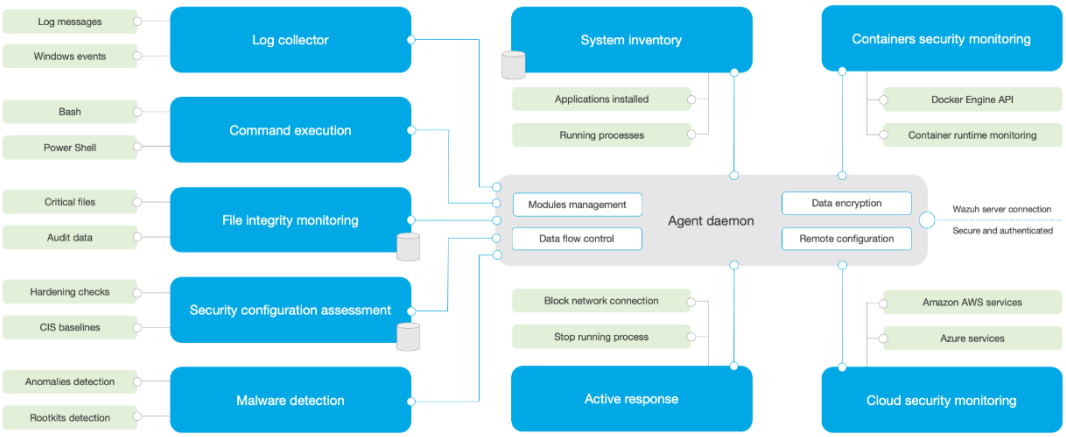

Figure 7. Arquitectura del agente de Wazuh.
Fuente: [35]

2.5.5. Active Response

Las Active Response realizan varias contramedidas para abordar las amenazas activas, como bloquear el acceso a un agente desde la fuente de la amenaza cuando se cumplen ciertos criterios.

Se ejecuta un script en respuesta a la activación de alertas específicas según el nivel de alerta o el grupo de reglas. Se puede iniciar cualquier número de scripts en respuesta a un desencadenante, sin embargo, estas respuestas deben considerarse detenidamente. La mala implementación de reglas y respuestas puede aumentar la vulnerabilidad del sistema [34].

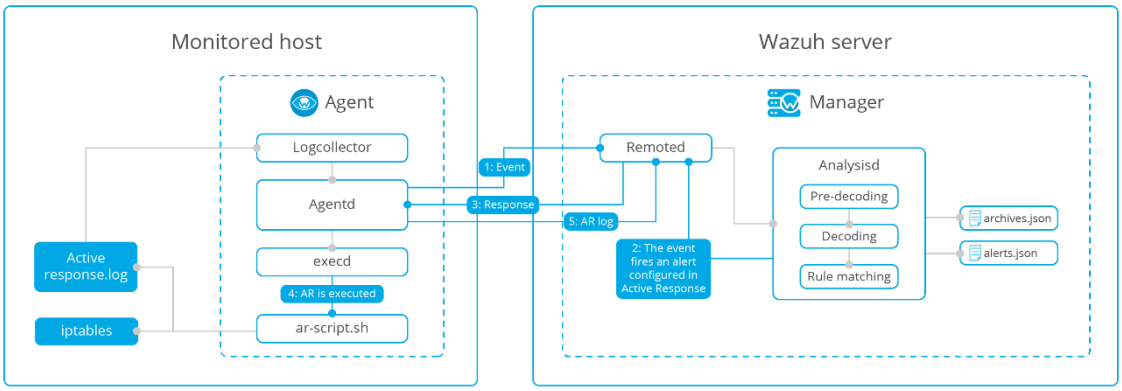

Figure 8. Ciclo de una Active Response con Wazuh
Fuente: [34]

El proceso en cuanto se detecta una anomalía en la seguridad para generar una Active Response es:

1) Evento
2) Evento dispara una alerta configurada.
3) Respuesta

4) Se ejecuta Active Response
5) Log Active Response

2.6. Telegram

Telegram es una app de mensajería en la nube para móviles y computadoras con foco en la seguridad y la velocidad [36]. Telegram ofrece llamadas cifradas de extremo a extremo [37] y chats "secretos" cifrados de extremo a extremo opcionales entre dos usuarios en línea en clientes de teléfonos inteligentes, mientras que los chats en la nube utilizan cifrado cliente-servidor / servidor-cliente.

2.6.1. Bots

Telegram lanzó una plataforma para que desarrolladores externos creen bots. Los bots son cuentas de Telegram operadas por programas. Pueden responder a mensajes o menciones, pueden ser invitados a grupos y pueden integrarse en otros programas [38]. Además, hay bots en línea, que se pueden usar desde cualquier pantalla de chat. Para activar un bot en línea, el usuario debe escribir en el campo del mensaje el nombre de usuario y la consulta de un bot. El bot luego ofrecerá su contenido. El usuario puede elegir entre ese contenido y enviarlo dentro de un chat [39].

3. Implementación y puesta en marcha

La implantación que se propone en este proyecto es un entorno de laboratorio donde se considera a Wazuh y Elastic Stack como herramientas para detección de anomalías, mismas que son detectadas para llevar a cabo respuestas ante estos incidentes con las Active Response.

3.1. Requerimientos

Los requerimientos para la implementación de la propuesta refieren a los componentes en la arquitectura que van a permitir realizar las pruebas de funcionamiento. Por lo tanto, se destacan los principales:

- Software GNS3 todo en uno (GUI)
- Elastic Stack
- Mánager de Wazuh
- Agente de Wazuh (Servidores)

Una vez que se plantean los requerimientos necesarios para poder llevar a cabo la implantación se procede a las instalación y configuración de estos.

3.1.1. Implementación y configuración de entorno de simulación

Para poder realizar un correcto laboratorio para pruebas de funcionamiento se consolida una arquitectura de red en GNS3, herramienta que permite crear, diseñar y probar redes en un entorno virtual. Este software de simulacion ofrece una manera fácil de diseñar y construir redes de cualquier tamaño sin necesidad de hardware adicional al del ordenador y esto se puede realizar de forma gratuita.

Para poder descargarlo se debe dirigir al siguiente enlace:

- https://www.gns3.com/software/download

A continuación se exponen los requerimientos mínimos del software:

OS	Windows 7 (64 bits) y versiones posteriores, Mavericks (10.9) y versiones posteriores, Any Linux Distro - Debian/Ubuntu son proporcionados y soportados
Procesador	2 o más núcleos lógicos - Serie AMD-V / RVI o Intel VT-X / EPT - extensiones de virtualización presentes y habilitadas en el BIOS. Más recursos permiten una simulación más grande
Memoria	4 GB de RAM
Almacenamiento	1 GB de espacio disponible (la instalación de Windows es inferior a 200 MB
Notas adicionales	Se necesita más almacenamiento para el sistema operativo y las imágenes del dispositivo.

Tabla 5. GNS3 - Requisitos mínimos
Fuente: [40]

GNS3 todo en uno es la parte cliente de GNS3 y es la interfaz gráfica de usuario (GUI). Se instala el software todo en uno en su PC local (Windows, MAC, Linux) y permite crear topologías con este software de forma personalizada.

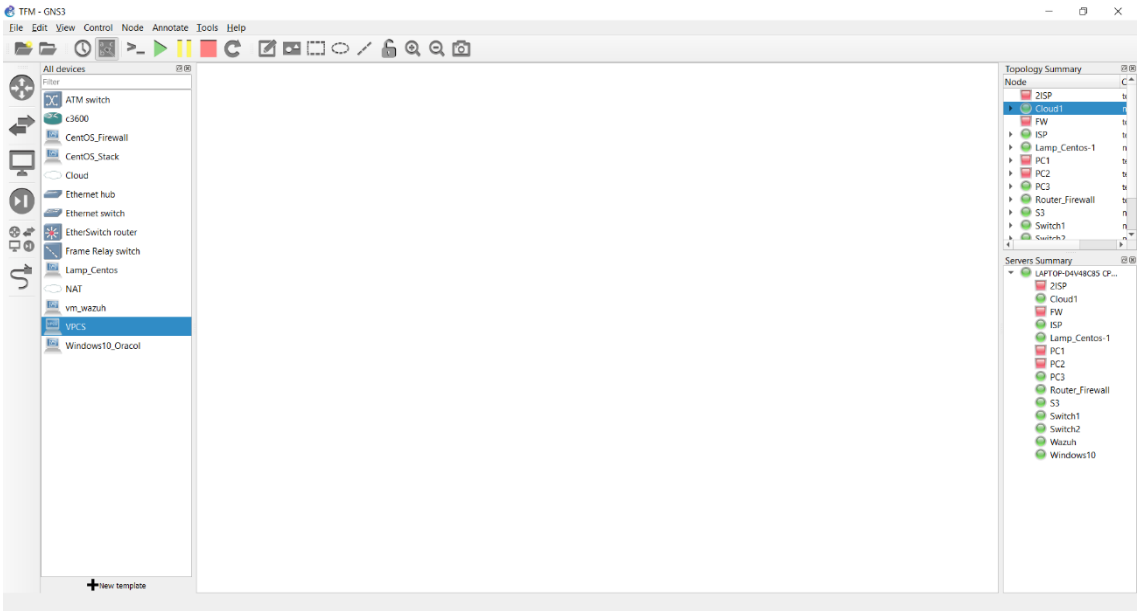

Figure 9. Interfaz gráfica de GNS3
Fuente: Autor

GNS3 es compatible con dispositivos emulados y simulados.

Emulación: GNS3 imita o emula el hardware de un dispositivo y ejecuta imágenes reales en el dispositivo virtual, es decir, se puede copiar el Cisco IOS de un router Cisco real, físico y ejecutar eso en un router Cisco virtual, emulado en GNS3.

Simulación: GNS3 simula las características y la funcionalidad de un dispositivo, como un Switch, de manera que no está ejecutando los sistemas operativos reales (tales como Cisco IOS), sino más bien, un dispositivo simulado desarrollado por el software propietario, como el Switch incorporado de la capa 2.

Para instalarlo se debe seguir los pasos que se brinda en la documentación en:

- https://docs.gns3.com/docs/

Una vez que el simulador está instalado en el ordenador, se debe agregar los dispositivos de red.

3.1.2. Instalación y configuración de dispositivos de red

Para agregar los dispositivos de red se toman en cuenta la emulación de router con las imágenes y la simulación de switches.

Como primera parte se descargan las imágenes desde:

- https://docs.gns3.com/docs/emulators/cisco-ios-images-for-dynamips

En este caso se utiliza los router Cisco de la serie C3640 con las siguientes características:

Nombre de archivo: c3640-a3js-mz.124-25d.bin
MD5: db9f63ca1b46d18fb835496bfffe608a
RAM mínima: 128MB

Para poder agregar la imagen descargada se debe dirigir a *Edit* > *Preferencias* > *IOS routes* y agregar una nueva imagen.

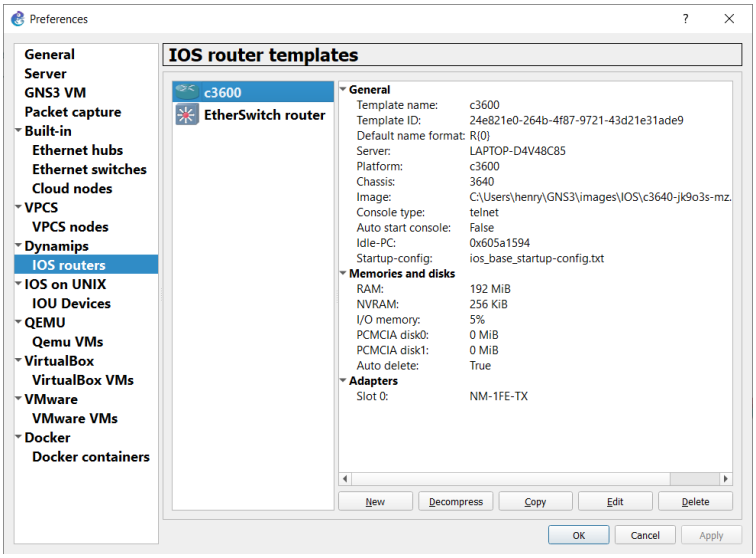

Figure 10. Agregar Router IOS en GNS3
Fuente: Autor

Con los Switch es diferente ya que estos vienen preinstalados para poder simularlos, donde su funcion en la arquitectura de red planteada es únicamente la de conmutar los paquetes.

Finalmente debemos arrastrar al área de trabajo los componentes para conectarlos y diseñar la arquitectura requerida. Pero antes de ello se agrega las VM de los servicios y stack requeridos.

Los routers realizan la fusión de convergencia de la red y así permite la simulacion de redes remotas y directamente conectadas, las configuraciones de cada router se exponen en el Anexo1.

3.1.3. Instalación y configuración de servicios en la red

Los servicios que se plantearon en la arquitectura son Apache Web Server y MySQL que permiten en este proyecto detectar anomalías poniendo a estos como objetivos de un atacante. En general y lo mas recomendado en las redes empresariales es situarlos de un DMZ (Zona Desmilitarizada), la cual está separada del resto de la red LAN.

Los servicios que se consideran para este proyecto son los más comunes para aplicativos web, esto se amplía en el apartado 2.3.2 en cuanto a la contribución del uso de estos.

3.1.3.1. Instalación de Apache

Apache es un servidor web multiplataforma de código abierto. Proporciona una gama completa de características de servidor web, incluyendo CGI, SSL y dominios virtuales.

Los siguientes comandos deben ejecutarse con privilegios de usuario root. Para instalar Apache, se ingresa los siguientes comandos en el terminal:

```
# yum install httpd -y
```

Se inicia el servicio Apache y se establece que se inicie automáticamente en cada reinicio:

```
# systemctl start httpd
# systemctl enable httpd
```

Para que el servidor Apache pueda ser alcanzado de forma remota se debe agregar una regla de firewall. Para ello, se ingresan los siguientes comandos desde el Terminal:

```
# firewall-cmd --permanent --add-service-http
# systemctl restart firewalld
```

Para probar el funcionamiento se lo hace desde un navegador web y colocamos http://172.17.1.10 que como resultado se obtiene lo mostrado en la Figure 11.

Figure 11. Prueba de funcionamiento de Apache Web Server.
Fuente: Autor

3.1.3.2. Instalación de MySQL

Se descarga el paquete rpm, que creará un archivo de repositorio yum para la instalación de MySQL Server. Esto se debe realizar como usuario root.

```
# yum install wget
# wget http://dev.mysql.com/get/mysql57-community-release-el7-7.noarch.rpm
```

Luego, se debe instala el paquete rpm descargado.

```
# rpm -ivh mysql157-community-release-el7-7.noarch.rpm
```

Para verificar que se haya instalado correctamente se verifica con el comando:

```
# ls -1 /etc/yum.repos.d/mysql-community*
```

Esta muestra los directorios donde se guardaron los repositorios:

```
[root@localhost ~]# ls -1 /etc/yum.repos.d/mysql-community*
/etc/yum.repos.d/mysql-community.repo
/etc/yum.repos.d/mysql-community-source.repo
[root@localhost ~]#
```

Figure 12. Validacion de instalación de MySQL.
Fuente: Autor

Finalmente se procede a la instalación de MySQL Server, con el comando:

```
# yum install mysql-server
```

Para poder iniciar el servicio y también lo haga desde el arranque, se debe ejecutar los comandos:

```
# systemctl enable mysqld
# systemctl start mysqld
```

En la instalación nueva de MySQL Server. La contraseña de usuario raíz de MySQL está en blanco.

Para una buena práctica de seguridad, debemos restablecer la contraseña MySQL usuario raíz ejecutando lo siguiente en el Terminal:

```
# mysql -u root
```

El comando permite ingresar por consola al servidor donde se ingresa a la consola, mostrando “mysql>”.

Se utilizan los siguientes comandos para restablecer la contraseña de root.

```
mysql> use mysql;
mysql> update user set password=PASSWORD("claveroot") where user='root';
mysql> flush privileges;
mysql> quit
```

Con la finalidad de hacer más administrable a MySQL con entorno gráfico se instala phpMyAdmin.

3.1.3.3. Instalación phpMyAdmin

Antes se debe instalar PHP con el siguiente comando:

```
# yum install php-mysql php-gd php-pear -y
```

La prueba de verificación del servicio se lo realiza mediante la creación de un archivo de prueba y se obtiene lo mostrado en la Figure 13.

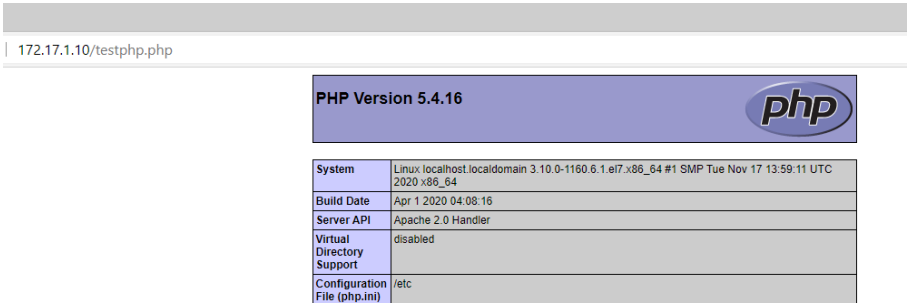

Figure 13. validación de funcionamiento de PHP.
Fuente: Autor

Por defecto phpMyAdmin no se encuentra en los repositorios oficiales de CentOS/RHEL/Scientific Linux. Así que se instala desde el repositorio EPEL.

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

Ahora, se instala phpMyAdmin:

```
# yum instalar phpmyadmin -y
```

Para la configuración, se edita el archivo phpmyadmin.conf:

```
# vi /etc/httpd/conf.d/phpMyAdmin.conf
```

En este archivo se comenta la sección <Directory> y se agregan las siguientes líneas como se muestra en la Figure 14.

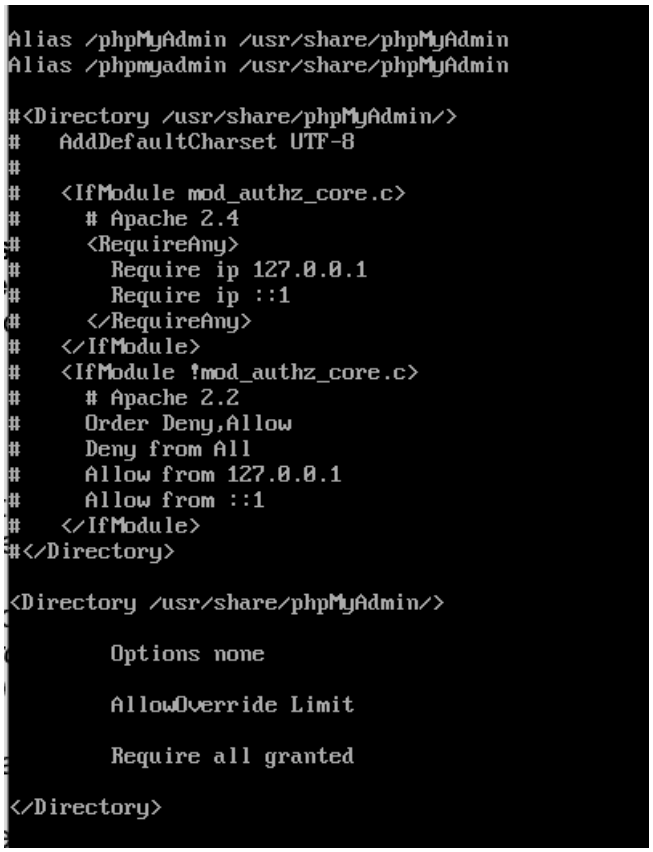

Figure 14. Configuración phpMyAdmin.
Fuente: Autor

Se edita el archivo "config.inc.php" y cambia de "cookie" a "http" para cambiar la autenticación en phpMyAdmin:

```
# vi /etc/phpMyAdmin/config.inc.php
```

El cambio se debe realizar en la sección que se muestra a continuación:

```
[...]

/* Tipo de autenticación */

$cfg['Servidores'][$i]['auth_type'] á 'http'; ¿Método de autenticación
(config, http o basado en cookies)?

[...]
```

Finalmente, se reinicia el servicio Apache:

```
# systemctl restart httpd
```

Ahora es posible acceder al entorno gráfico desde el navegador ingresando un usuario de base de datos.

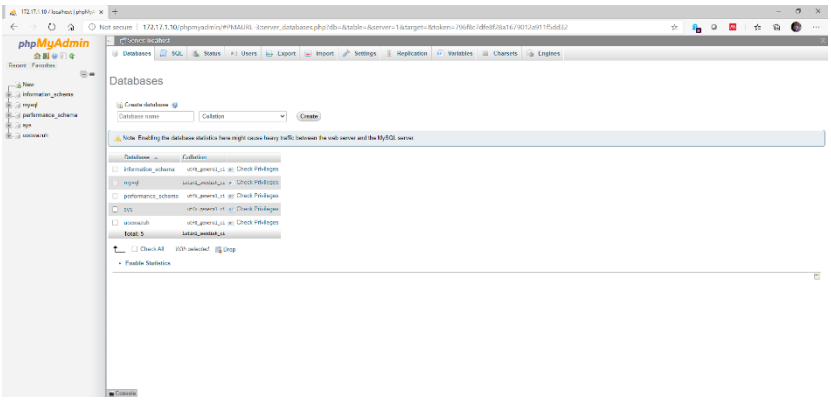

Figure 15. phpMyAdmin
Fuente: Autor

Se debe introducir usuario y contraseña de MySQL que se proporcionaron en 3.1.3.2. Luego se redirigirá a la interfaz web principal de PhpMyAdmin.

Ahora es posible gestionar las bases de datos MySQL desde la interfaz web phpMyAdmin.

3.2. Diseño/Adaptación

Hay dos alternativas diferentes para la implementación que se debe considerar en el diseño en el apartado :

- Todo en uno: Todos los componentes de Wazuh están instalados en el mismo host (Elastic Stack, Mánager Wazuh y Wazuh Kibana pluing), este tipo de implementación es adecuada para pruebas y pequeños entornos de trabajo según la documentación de Wazuh.

- Distribuido: Cada componente se instala en un host independiente como un clúster de un solo nodo o de varios nodos. Este tipo de implementación permite la alta disponibilidad y escalabilidad del producto y es adecuado para entornos de trabajo grandes.

En el caso de este proyecto se plantea una implementación todo en uno, ya que el entorno de laboratorio planteado corresponde a las recomendaciones de la documentación. Por ello en este apartado se indica la forma de instalación, implementación y pruebas de funcionamiento.

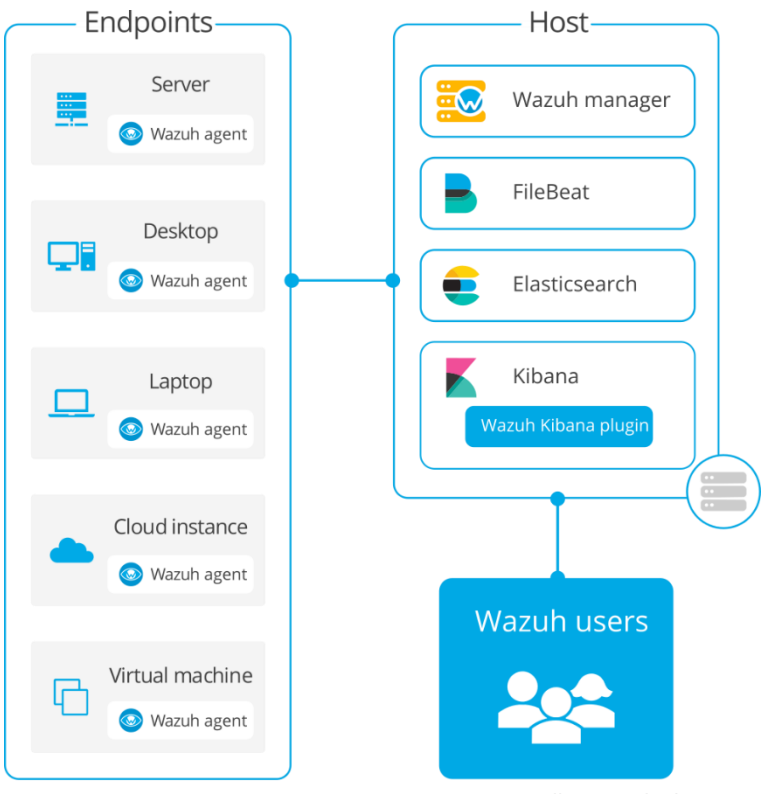

Figure 16. Diagrama de implementación de Wazuh todo en uno.
Fuente: [41]

Se expone la arquitectura de red con los componentes implementados, de manera que se agrega seguridad a la red empresarial en cuanto a deteccion de amenazas. Esto implica la convergencia de una red simulada en GNS3 con equipos de red, Elastic Stack, Mánager de Wazuh, Apache Server y MySQL, estos ultimo son los agentes de Wazuh que están instalados y configurados en un mismo host.

Para poder hacer la detección de anomalías se considera los datos de registros de los servidores donde los agentes de Wazuh leen los registros del sistema operativo y de las aplicaciones, y los reenvían de forma segura al mánager de Wazuh para el análisis y el almacenamiento basados en reglas.

Las reglas de Wazuh le ayudan a tener en cuenta los errores de aplicación o del sistema, las configuraciones erróneas, los intentos y/o las actividades maliciosas exitosas, las violaciones de políticas y una variedad de otros problemas operativos y de seguridad [41].

A continuación se muestra la tabla de direccionamiento para la arquitectura de red del laboratorio:

Dispositivo	Interfaz	IP
ISP	e0/0	192.168.2.2/24
	e0/1	172.17.0.1/24
	e0/2	172.17.3.1/24

Cloud	Loopback	192.168.2.1/24
Router_Firewall	e0/0	172.17.2.1/24
	e0/1	172.17.0.2/24
	e0/2	172.17.1.1/24
Lamp_Centos-1		172.17.1.10/24
Windows10		172.17.2.10/24
WAZUH4		172.17.2.20/24
KaliLinux-1		172.17.3.10/24

Tabla 6. Tabla de direccionamiento.
Fuente: Autor

Dada la tabla de direccionamiento se establece la arquitectura de red con la siguiente topología:

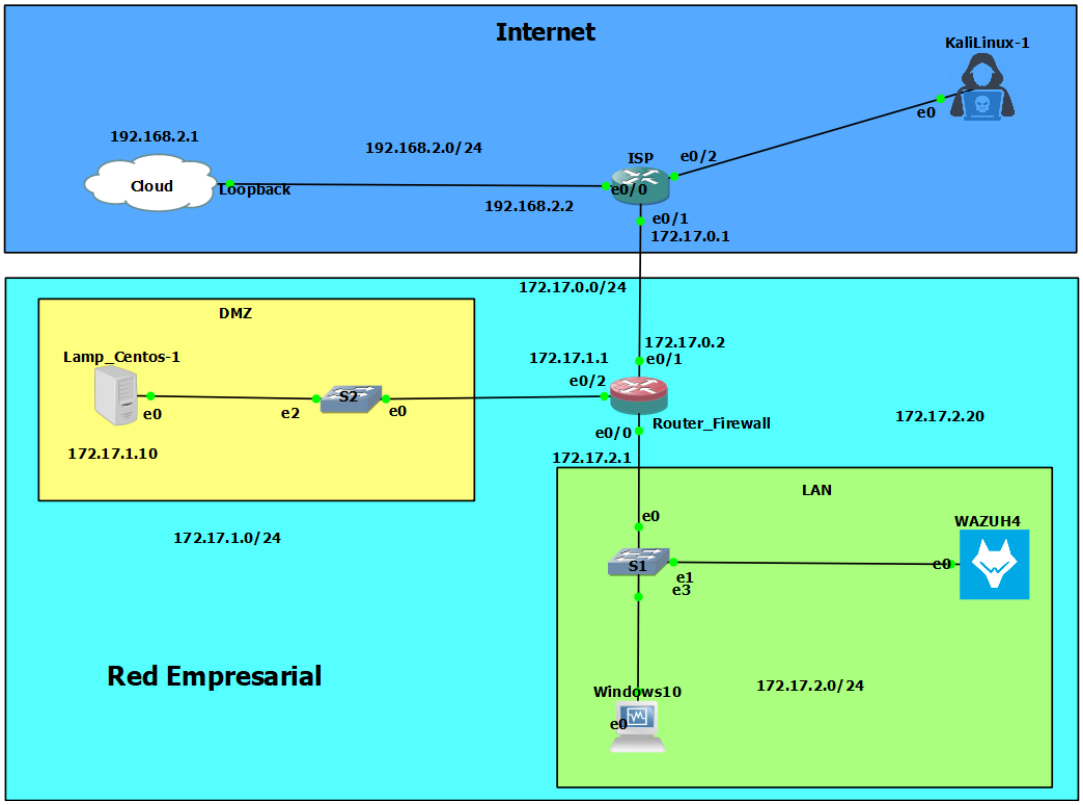

Figure 17. Arquitectura de red implementada
Fuente: Autor

Es importante para la instalación y puesta en marcha de esta solución exponer en esta memoria las versiones las cuales han sido consideradas, recalcando que son las ultimas hasta la actualidad de realizado este trabajo de fin de máster. A continuación en se expone las versiones utilizadas:

Componentes	Versión
FileBeat	7.9.1
Elasticsearch	7.9.1
Kibana	7.9.1

Mánager de Wazuh	4.0.2
Agente de Wazuh	4.0.2

Tabla 7. Versiones consideradas.
Fuente: Autor

3.2.1. Instalación del Mánager de Wazuh

Antes de la instalación del Mánager de Wazuh se debe instalar un prerrequisito para Elasticsearch que requiere el Kit de desarrollo de Java y la instalación de otros paquetes, incluyendo, y que se utilizará en otros pasos como wget, curl, unzip y libcap.

Para instalar los paquetes necesarios se ingresa:

```
# export JAVA_HOME=/usr/ && yum install curl unzip wget libcap && yum install
java-11-openjdk-devel
```

El primer paso para configurar Wazuh es agregar el repositorio al servidor.

Se debe importar la clave GPG[1] con:

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

Para configurar el repositorio, se ejecuta el comando:

```
# cat > /etc/yum.repos.d/wazuh.repo << EOF
[wazuh]
gpgcheck=1
gpgkey=https://packages.wazuh.com/key/GPG-KEY-WAZUH
enabled=1
name=EL-$releasever - Wazuh
baseurl=https://packages.wazuh.com/4.x/yum/
protect=1
EOF
```

El siguiente paso es instalar el manage Wazuh en el sistema con el comando:

```
# yum install wazuh-manager
```

Después se debe habilitar e iniciar:

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

[1] GNU Privacy Guard

Una vez completado el proceso, se comprueba el estado del servicio con:

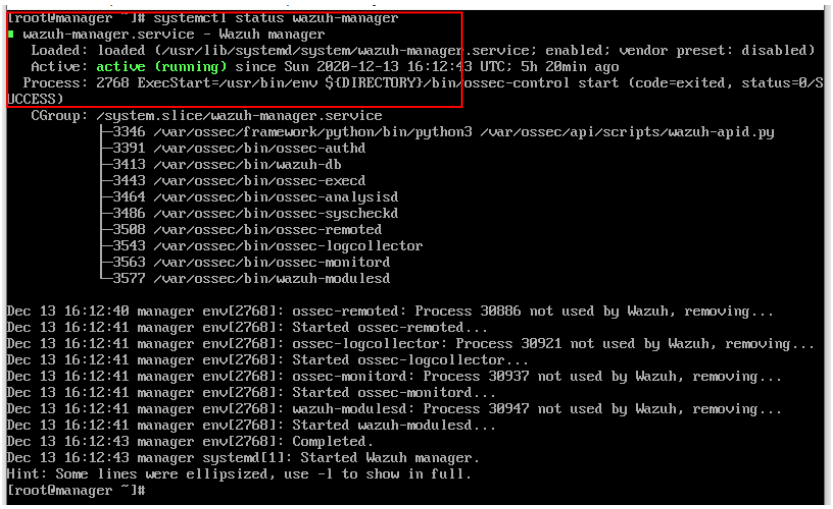

Figure 18. Estado del mánager de Wazuh
Fuente: Autor

3.2.2. Instalación del agente de Wazuh

Para la instalación del agente se debe considerar el sistema operativo y la versión del mánager. Dado que en el laboratorio planteado se utiliza únicamente Centos 7, a continuación se muestra los pasos solo para esta distribución:

Se debe agregar el repositorio de Wazuh con los siguientes comandos:

- Se importa la clave GPG:

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

- Se agrega el repositorio de Wazuh.

```
# cat > /etc/yum.repos.d/wazuh.repo << EOF
[wazuh]
gpgcheck=1
gpgkey=https://packages.wazuh.com/key/GPG-KEY-WAZUH
enabled=1
name=EL-$releasever - Wazuh
baseurl=https://packages.wazuh.com/4.x/yum/
protect=1
EOF
```

En el Terminal se agrega el comando de instalación:

```
# yum install wazuh-agent
```

Ahora, se configura la IP de mánager en el archivo de configuración */var/ossec/etc/ossec.conf*.

```
<!--
  Wazuh - Agent - Default configuration for centos 7.9
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <client>
    <server>
      <address>172.17.2.20</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>centos, centos7, centos7.9</config-profile>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <crypto_method>aes</crypto_method>
  </client>
```

Figure 19. Archivo de configuración del mánager de Wazuh
Fuente: Autor

Ahora que el agente está instalado, el siguiente paso es registrarlo y configurarlo para comunicarse con el mánager.

Para esto se debe ingresar en el Terminal el comando:

```
/var/ossec/bin/manage_agents
```

Este permite ejecutar por consola un menú de opciones que se pueden aplicar.

```
[root@manager ~]# /var/ossec/bin/manage_agents


****************************************
* Wazuh v4.0.2 Agent manager.          *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: L

Available agents:
   ID: 001, Name: localhost.localdomain, IP: any
```

Figure 20. Opciones del mánager de Wazuh
Fuente: Autor

Se debe elegir “A” para agregar un nuevo agente completando los datos que se requiere como un nombre, IP y un ID.

Es necesario extraer la llave del agente ingresado “E”, ya que permite vincular al servidor o host que se requiere. A continuación se muestra la llave generada posteriormente al ingreso del agente:

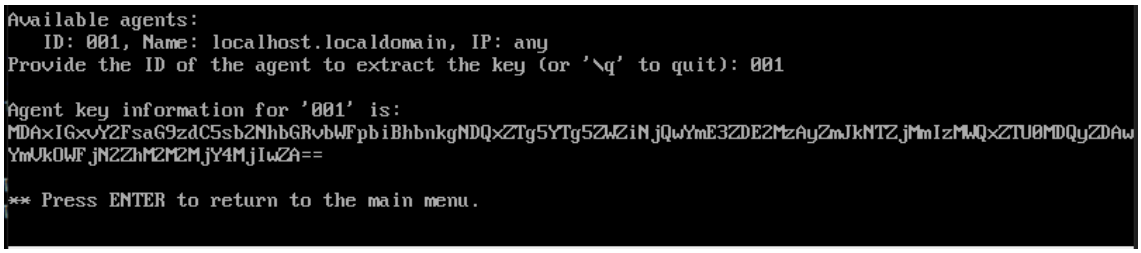

Figure 21. Generación de llave para agente de Wazuh
Fuente: Autor

Finalmente se debe agregar la llave generada en el agente, ejecutando el mismo comando y eligiendo la opción “I”.

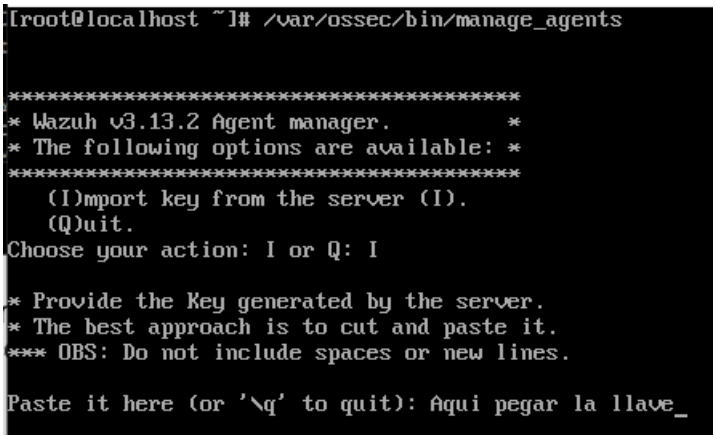

Figure 22. Ingreso de llave en agente de Wazuh
Fuente: Autor

Se muestra a continuación que el agente es agregado con éxito y mostrado en la interfaz Kibana en estado activo.

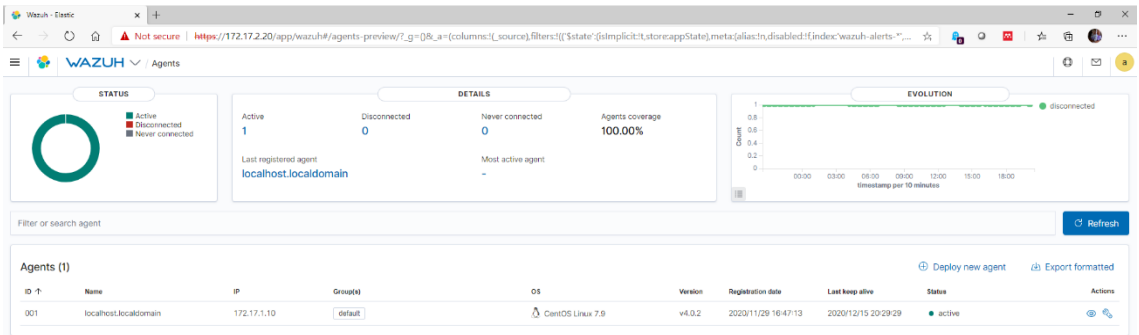

Figure 23. Estado activo de agente
Fuente: Autor

3.2.3. Instalación de Elastic Stack

En este apartado se detalla la instalación de Elastic Stack y se expone las consideraciones que se debe tomar en cuenta para vincular con el mánager de Wazuh como pluging en la interfaz de Kibana.

3.2.3.1. Instalación de Elasticsearch

Elasticsearch es un motor de búsqueda y análisis de texto completo escalable, se instala el paquete con el comando:

```
# yum install opendistroforelasticsearch
```

Para poder utilizar el plugin de Wazuh de forma correcta, es necesario añadir usuarios y roles adicionales. Para esto se ejecutaron los siguientes comandos:

```
# curl -so
/usr/share/elasticsearch/plugins/opendistro_security/securityconfig/roles.y
ml https://raw.githubusercontent.com/wazuh/wazuh-
documentation/4.0/resources/open-distro/elasticsearch/roles/roles.yml

# curl -so
/usr/share/elasticsearch/plugins/opendistro_security/securityconfig/roles_m
apping.yml https://raw.githubusercontent.com/wazuh/wazuh-
documentation/4.0/resources/open-
distro/elasticsearch/roles/roles_mapping.yml

# curl -so
/usr/share/elasticsearch/plugins/opendistro_security/securityconfig/interna
l_users.yml https://raw.githubusercontent.com/wazuh/wazuh-
documentation/4.0/resources/open-
distro/elasticsearch/roles/internal_users.yml
```

Los comandos que se ingresan crean los siguientes usuarios:

- wazuh_user: Creado para accesos de usuario de solo lectura.
- wazuh_admin: Creado para accesos de usuario con privilegios de administrador.
- wazuh_ui_user: Rol que permite leer los índices de Wazuh para wuazuh_user.
- wazuh_ui_admin: Realiza tareas de lectura, escritura, gestión e indexación en los índices de Wazuh para wazuh_admin.

Seguidamente se debe retirar los certificados de prueba:

```
# rm /etc/elasticsearch/esnode-key.pem /etc/elasticsearch/esnode.pem
/etc/elasticsearch/kirk-key.pem /etc/elasticsearch/kirk.pem
/etc/elasticsearch/root-ca.pem -f
```

Luego, se debe generar e implementar los certificados:

- Se crea el directorio para almacenar los certificados, para esto se debe dirigir a la ubicación de instalación.

```
# mkdir /etc/elasticsearch/certs
# cd /etc/elasticsearch/certs
```

- Por consiguiente, para generar los certificados se utiliza Search Guard y para descargarlo se ejecuta:

```
# curl -so ~/search-guard-tlstool-1.8.zip https://maven.search-
guard.com/search-guard-tlstool/1.8/search-guard-tlstool-1.8.zip
```

- Para extraer los archivos:

```
# unzip ~/search-guard-tlstool-1.8.zip -d ~/searchguard
```

- Luego, se descargar un archivo preconfigurado “search-guard.yml” para generar todos los certificados necesarios:

```
# curl -so ~/searchguard/search-guard.yml
https://raw.githubusercontent.com/wazuh/wazuh-documentation/4.0/resources/open-
distro/searchguard/search-guard-aio.yml
```

- Se ejecuta el script para crear los certificados:

```
# ~/searchguard/tools/sgtlstool.sh -c ~/searchguard/search-guard.yml -ca -
crt -t /etc/elasticsearch/certs/
```

- Se eliminan los archivos innecesarios:

```
# rm /etc/elasticsearch/certs/client-certificates.readme
/etc/elasticsearch/certs/elasticsearch_elasticsearch_config_snippet.yml
~/search-guard-tlstool-1.8.zip ~/searchguard -rf
```

- Se inicia y habilita el servicio Elasticsearch:

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

- Finalmente, se ejecuta el script de Elasticsearch para gargar nueva información de certificados e iniciar el clúster:

```
# /usr/share/elasticsearch/plugins/opendistro_security/tools/securityadmin.sh -cd
/usr/share/elasticsearch/plugins/opendistro_security/securityconfig/ -nhnv -cacert
/etc/elasticsearch/certs/root-ca.pem -cert /etc/elasticsearch/certs/admin.pem -key
/etc/elasticsearch/certs/admin.key
```

- Se verificar que la instalación se haya realizado correctamente:

Figure 24. Comprobación de Elasticsearch
Fuente: Autor

En la Figure 24 se muestra que la consulta realizada es correcta, mostrando la respuesta, la cual se la hace por una conexión segura.

3.2.3.2. Instalación de Filebeat

Para continuar con la configuración es necesario instalar Filebeat, herramienta del servidor Wazuh que reenvía de forma segura alertas y eventos archivados a Elasticsearch. Para instalarlo:

Instalación de paquetes Filebeat con:

```
# yum install filebeat
```

Se procede a descargar el archivo de configuración Filebeat preconfigurado utilizado para reenviar las alertas de Wazuh a Elasticsearch:

```
# curl -so /etc/filebeat/filebeat.yml
https://raw.githubusercontent.com/wazuh/wazuh-
documentation/4.0/resources/open-
distro/filebeat/7.x/filebeat_all_in_one.yml
```

Luego se debe descargar la plantilla de alertas para Elasticsearch:

```
# curl -so /etc/filebeat/wazuh-template.json
https://raw.githubusercontent.com/wazuh/wazuh/4.0/extensions/elasticsearch/
7.x/wazuh-template.json

# chmod go+r /etc/filebeat/wazuh-template.json
```

Posteriormente se debe descargar el módulo de Wazuh para Filebeats:

```
# curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.1.tar.gz
| tar -xvz -C /usr/share/filebeat/module
```

Se debe copiar los certificados de Elasticsearch en el directorio /etc/filebeat/certs.

```
# mkdir /etc/filebeat/certs
# cp /etc/elasticsearch/certs/root-ca.pem /etc/filebeat/certs/
# mv /etc/elasticsearch/certs/filebeat* /etc/filebeat/certs/
```

Finalmente se habilita e inician el servicio:

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

Para asegurar que Filebeat se ha instalado correctamente, debemos ejecutar el siguiente comando:

```
# filebeat test output
```

Se puede apreciar en Figure 25 que la configuración es correcta:

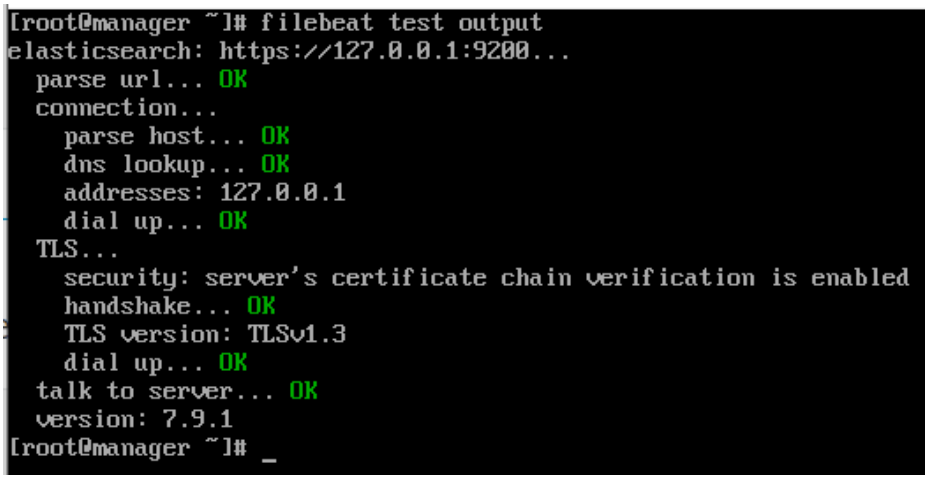

Figure 25. Verificación de instalación de Filebeart
Fuente: Autor

Se habilita e inicia el servicio Filebeat:

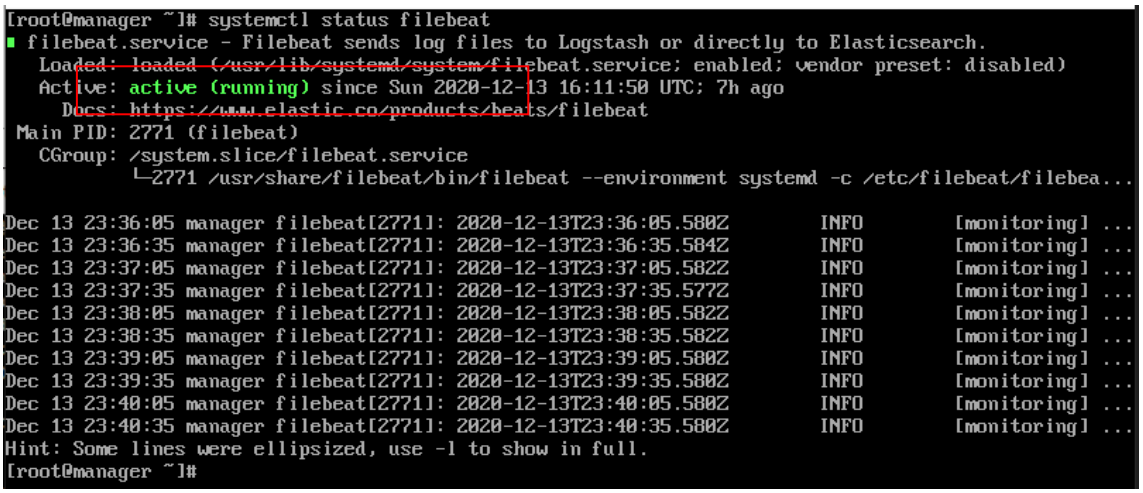

Figure 26. Estado de Filebeat
Fuente: Autor

3.2.3.3. Kibana

Kibana es una interfaz web flexible e intuitiva para minería y visualización de eventos y archivos almacenados en Elasticsearch. Para comenzar con la instalación se ejecuta el comando:

```
# yum install opendistroforelasticsearch-kibana
```

Se descargar el archivo de configuración de Kibana, donde están las configuraciones para acceder desde fuera y aceptar las IPs de host permitidos y puede ser cambiando en /etc/kibana/kibana.yml

Se debe actualizar los permisos y directorios:

```
# chown -R kibana:kibana /usr/share/kibana/optimize
# chown -R kibana:kibana /usr/share/kibana/plugins
```

Para instalar el plugin de Wazuh para Kibana se ejecuta:

```
# cd /usr/share/kibana
# sudo -u kibana /usr/share/kibana/bin/kibana-plugin install
https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.0.3_7.9.1-1.zip
```

Se copian los certificados de Elasticsearch en el directorio */etc/kibana/certs*:

```
# mkdir /etc/kibana/certs
# cp /etc/elasticsearch/certs/root-ca.pem /etc/kibana/certs/
# mv /etc/elasticsearch/certs/kibana_http.key /etc/kibana/certs/kibana.key
# mv /etc/elasticsearch/certs/kibana_http.pem /etc/kibana/certs/kibana.pem
```

Se configura el enlace de socket Kibana al puerto seguro 443:

```
# setcap 'cap_net_bind_service=+ep' /usr/share/kibana/node/bin/node
```

Finalmente se habilita e inician el servicio:

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

Kibana sólo escuchará en la interfaz de loopback (localhost) de forma predeterminada, lo que significa que sólo se puede acceder desde la misma máquina. Para acceder a Kibana de forma remota, se cambia el archivo /etc/kibana/kibana.yml.

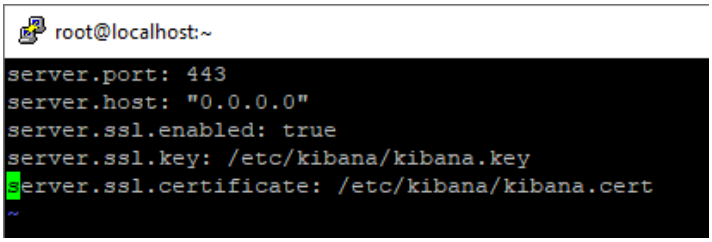

Figure 27. Archivo de configuración de Kibana
Fuente: Autor

A continuación se valida el funcionamiento del Kibana en el navegador:

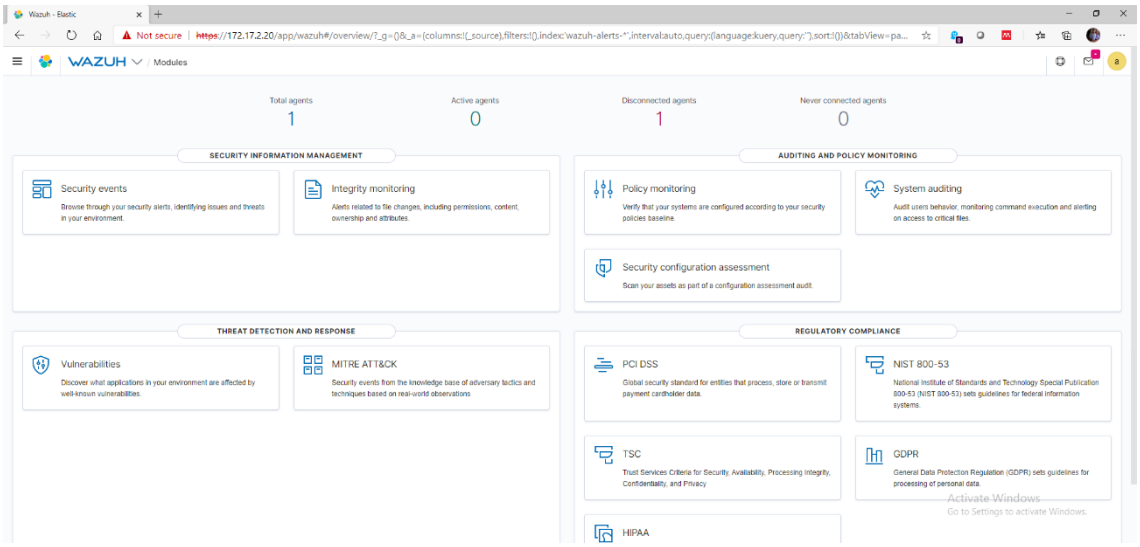

Figure 28. Interfaz de Kibana
Fuente: Autor

3.2.4. Configuración de Active Response para deteccion de ataques

En la sección 2.5.5 se explica su funcionamiento, por lo tanto en este apartado se muestra cómo bloquear ataques mediante la característica de Active Response en OSSEC. La respuesta activa permite que OSSEC ejecute

comandos en un agente en respuesta a determinados desencadenadores. Consecuentemente se simula un ataque SQL injection y DoS en el apartado 3.3.

3.2.4.1. Detectar el ataque

Lo primero que se debe realizar es determinar cuándo se va a ejecutar la active response. Para esto se tienen las siguientes opciones:

- Identificador de regla: la respuesta se ejecutará en cualquier evento con el identificador definido.
- Grupo de reglas: la respuesta se ejecutará en cualquier evento del grupo definido.
- Nivel: La respuesta se ejecutará en cualquier evento con este nivel o superior.

En el caso de este laboratorio se requiere evitar ataques del tipo DoS y SQL Injetion por lo que cuando se activa la regla 31164 y 31151 por motivo de “Intento SQL injection” y “Múltiples códigos de error del servidor web 400 de la misma IP de origen” respectivamente se ejecuta la respuesta activa correspondiente para bloquear la IP del atacante.

3.2.4.2. Definición del comando

Una vez que se comprende el momento en que se va a ejecutar la respuesta activa, ahora se define lo que hará. Si bien se puede crear un propio script para poder ejecutar, en este caso OSSEC brinda un conjunto de scripts preconfigurados para utilizar que se pueden encontrar en var/ossec/active-response/bin/firewall-drop.sh, y para realizar lo requerido se utiliza “firewall-drop.sh”, este funciona para sistemas operativos Linux/Unix comunes y permite bloquear una IP maliciosa mediante el firewall local.

Luego se define el comando en el mánager de Wazuh en el fichero “ossec.conf”.

```
<command>
    <name>firewall-drop</name>
    <executable>firewall-drop.sh</executable>
    <expect>srcip</expect>
    <timeout_allowed>yes</timeout_allowed>
</command>
```

Se establece el nombre del comando para posterior usa en la respuesta activa y la ruta de acceso del script que se va a ejecutar. Si el script necesita argumentos, debe especificarse en el campo “expect”. En este caso, el script necesita saber qué IP bloquear. Los decodificadores deben extraer ese campo para funcionar. Por último, es posible establecer un tiempo de espera definido para desbloquear la IP.

3.2.4.3. Definición de la respuesta activa

Ahora, se configura OSSEC para ejecutar la respuesta activa. Los campos principales son:

- command: el comando previamente definido (firewall-drop).
- location: donde se debe ejecutar el comando. Queremos ejecutar el comando en el agente que informó del evento. Por lo tanto, usamos local.
- rules_id: el comando se ejecuta si se activa la regla 5712.
- timeout: Bloquee la IP durante 1800 segundos en el firewall (iptables, ipfilter, etc.).

La respuesta activa al igual que el comando debe ser definida en el mánager en el fichero “ossec.conf”.

```
<active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>31164</rules_id>
    <timeout>1800</timeout>
</active-response>
```

Nota: el campo *timeout* se configura con la finalidad de facilitar las pruebas de concepto que se realizan en el siguiente apartado.

3.2.5. Configuración de Alertas por Telegram

Las alertas son un punto fundamental a la ahora de las respuestas frente a los ataques que puede enfrentar nuestros servidores y red, por lo tanto en este apartado se realizaron la pruebas de concepto para la ejecución de alertas enviadas por mensajería instantánea utilizando la API (Una interfaz de programación de aplicaciones) de Telegram, más información del funcionamiento en el apartado 2.6.

A continuacion se precenta uno de los bots más utilizados en Telegram para poder interactuar. En el caso de este proyecto se lo utiza para establecer un canal de comunicación entre el agente y la aplicación de mensajeria instantanea.

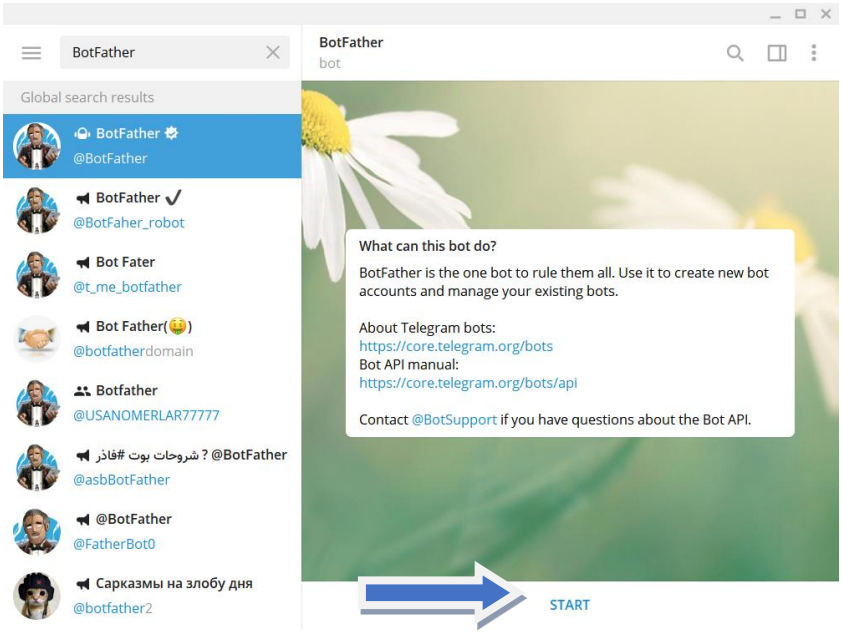

Figure 29. Interfaz de Telegram - BotFather.
Fuente: Autor

De la multiples funciones que se permiten una vez iniciada la comunicación, se crea un nuevo bot ingresando el texto:

```
/newbot
```

Luego se debe asignar un nombre para el bot. En este caso de los nombra *Wazuh*.

Finalmente se debe asignar un nombre de usuario el cual termine en “bot”, de esta manera se coloca *Wazuh_alerts_bot*.

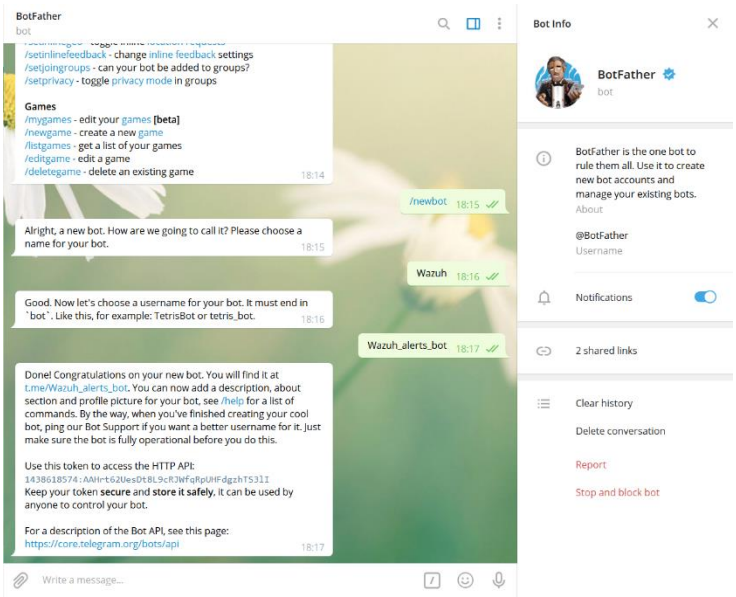

Figure 30. Creación de bot de Telegram.
Fuente: Autor

A continuación se recibe el token que permite la comunicación por HTTP.

```
1438618574:AAHrt62UesDt8L9cRJWfqRpUHFdgzhTS3lI
```

Para verificar la funcionalidad se debe ingresar como la URL (Localizador Uniforme de Recursos) en el navegador como se muestra:

```
https://api.telegram.org/bot1438618574:AAHrt62UesDt8L9cRJWfqRpUHFdgzhTS3lI/
getupdates
```

La respuesta de la consulta realizada se muestra en la Figure 31, donde se puede identificar el chat ID que es 765515016:

Figure 31. Respuesta de API de Telegram.
Fuente: Autor

Para poder validar la recepción de los mensajes se hace la siguiente consulta en el navegador:

```
https://api.telegram.org/bot1438618574:AAHrt62UesDt8L9cRJWfqRpUHFdgzhTS3lI/
sendmessage?chat_id=765515016&text=Hola Henry
```

La validación de la recepción en la aplicación se valida en la Figure 32:

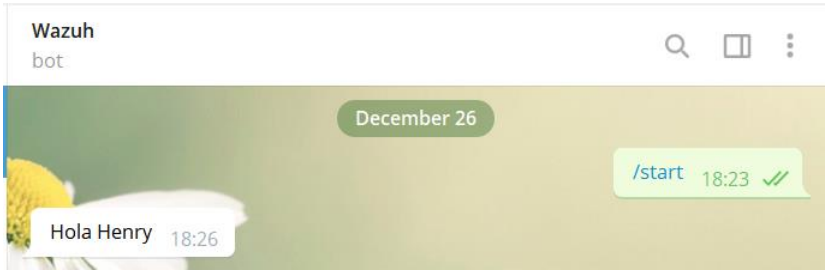

Figure 32. Recepción de mensaje en aplicación de Telegram.
Fuente: Autor

Para realizar las pruebas de funcionamiento de alertas por mensajería instantánea, en este proyecto se crea un archivo el cual hace la revisión de las respuestas activas generadas en los registros y posteriormente envía la información por mensajería instantánea. Este script se lo puede apreciar en el Anexo 3.

El script es agregado en /var/ossec/active-response/bin del agente, de esta manera se agrega los permisos de ejecución y propietario.

```
#chmod 750 /var/ossec/active-response/bin/alert-telegram.sh
#chown root:ossec /var/ossec/active-response/bin/alert-telegram.sh
```

Finalmente se debe realizar las configuraciones en *ossec.conf* para establecer la active response que se ha creado. Por consecuencia se establecen las siguientes etiquetas:

```
<command>
    <name>alert-telegram</name>
    <executable>alert-telegram.sh</executable>
    <expect></expect>
  </command>

<active-response>
     <command>alert-telegram</command>
     <location>local</location>
     <level>6</level>
 </active-response>
```

3.3. Pruebas de Funcionamiento

Las pruebas de conceto para este apartado consisten en realizar diferentes ataques al servidor web y la base de datos MySQL. Para esto se desarrolla una aplicación web la cual se conecta a la BDD y realiza la consulta de la existencia del usuario una vez que realiza un login, el codigo de la aplicación se puede apreciar en el Anexo2. Cabe aclarar que es una aplicación para pruebas de concepto básica para este proyecto.

3.3.1. Ejecución de ataque SQL injection

El ataque consiste en interferir en las consultas que realiza la aplicación a la base de datos.

Se propone un entorno desarrollado en PHP el cual cuenta con una sección de noticias. Las noticias mostradas varían según el “id” que identifica a cada noticia que se muestra por lo tanto este sitio es vulnerable.

El esquema de la base de datos SQL está configurada como se muestra a continuación en el entorno grafico con phpMyAdmin.

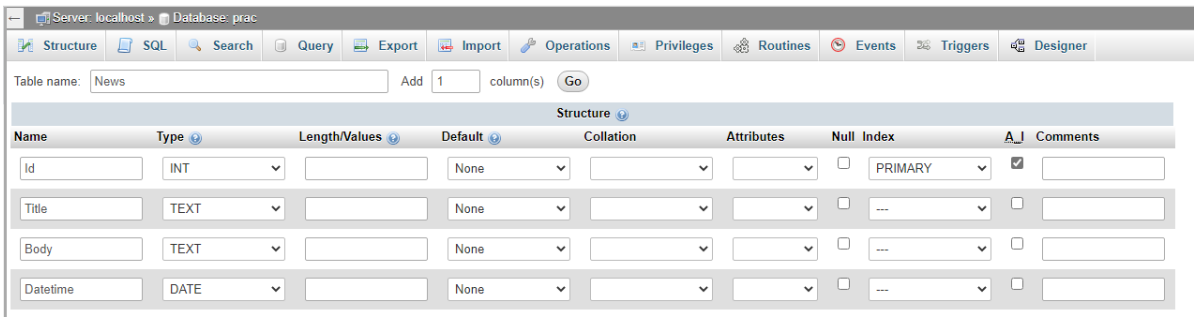

Figure 33. Configuración de la tabla News
Fuente: Autor

En este caso para crear las noticias del sitio web, se debe ingresar la información que esta almacenada en la base de datos SQL y por lo tanto esto es mostrado en el front-end, una vez que la consulta se realiza desde la misma.

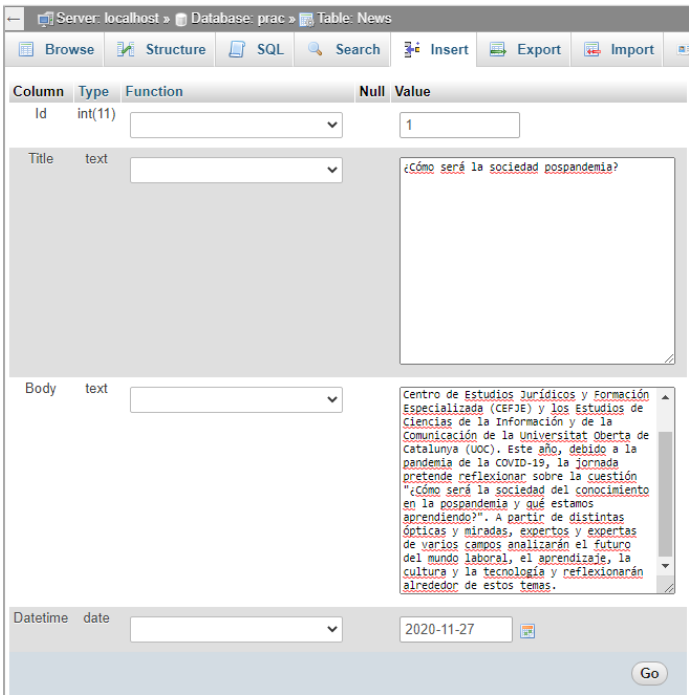

Figure 34. Creación de contenido para sitio web de prueba.
Fuente: Autor

Se crea un sitio de noticias sobre un servidor Apache, este está conectado a la base de datos MySQL. Como resultado se muestra en la información que contienen la base de datos.

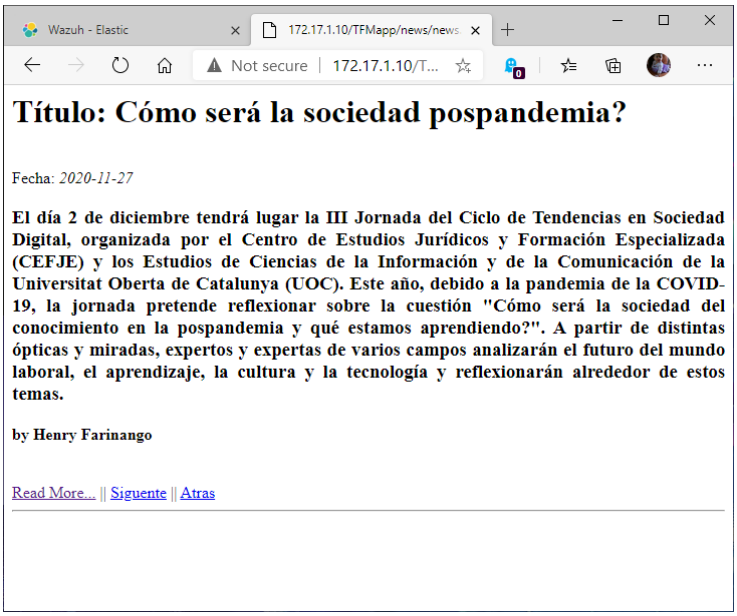

Figure 35. Sitio web de prueba.
Fuente: Autor

Para validar que el parametro *id* sea vulnerable a inyección SQL se relaiza lo siguiente:

Se realiza un ataque SQL injection del tipo UNION para ver la informacion de la BDD, cabe aclarar que al ya conocer información de la estrcutura de base de datos unicamente se hace la ejcucion del ataque, esto se lo realiza con lo siguiente que es ejecutado en la seccion de URL.

```
http://172.17.1.10/TFMapp/news/news.php?id=-3 union select * from Users
```

Como resultado se obtine:

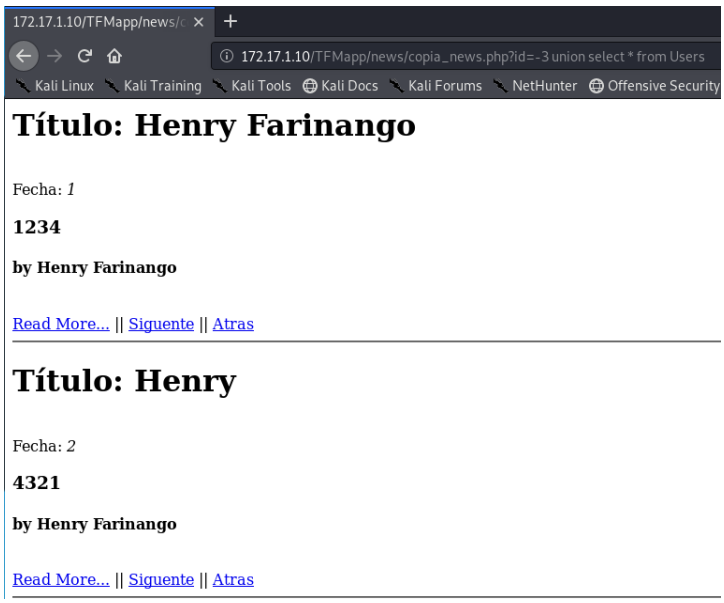

Figure 36. Explotación de vulnerabilidad por SQL injection.
Fuente: Autor

Se puede apreciar que mediante este ataque se deja expuesta la información de usuarios, identificando un exponencial robo de identidad digital.

Posterior al ataque, se puede identificar en la interfaz la alerta del ataque ejecutado, por lo tanto se debe proceder a generar una respuesta activa para este tipo de ataques.

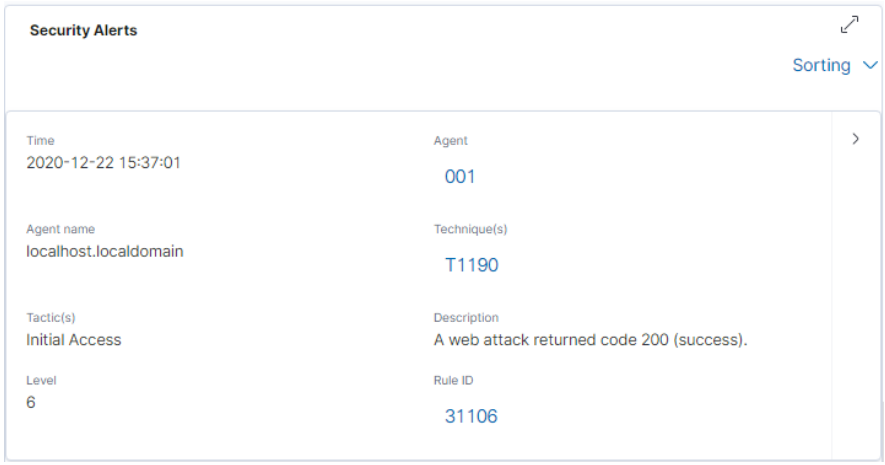

Figure 37. Alerta de seguridad - ataque SQL injection del tipo UNION.
Fuente: Autor

Ahora se debe agregar la respuesta activa en base al número identificador de la regla, en este caso 31106.

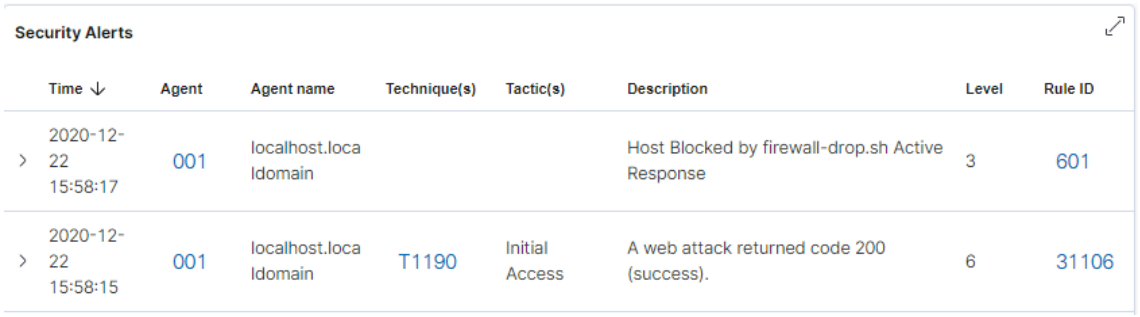

Figure 38. Validación de respuesta activa posterior al ataque SQL injection.
Fuente: Autor

De igual manera se realiza la verificación del funcionamiento de esta solución de seguridad para ataques SQL injection básicos, de esta manera se tiene una aplicación de login, el cual se conecta a una base de datos MySQL donde se encuentran los usuarios. Se valida que se puede hacer login con los usuarios creados:

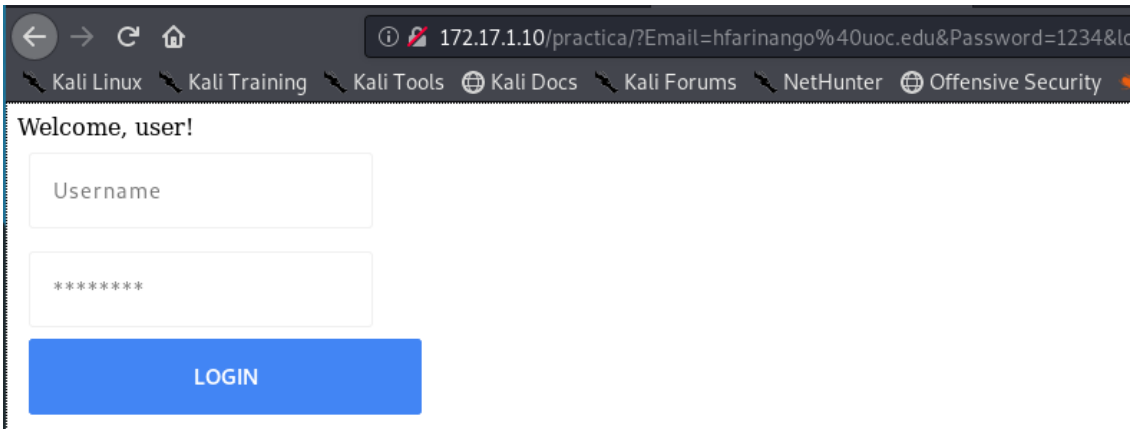

Figure 39. Acceso correcto de aplicación.
Fuente: Autor

Una vez que la aplicación es funcional se precede a ejecutar la inyección de codigo a la aplicación que por fines de prueba está desarrollada con vulnerabilidad para el ingreso de espacios. Por lo tanto se espera que sea identificado por Wazuh al memento de intentar hacer la inyección de codigo explotando la vulnerabilidad.

Para explotar la vulnerabilidad se ingresa la consulta:

‘ or 1=1#

Donde al momento de ejecutar la consulta desde el login automáticamente se ejecuta la respuesta activa.

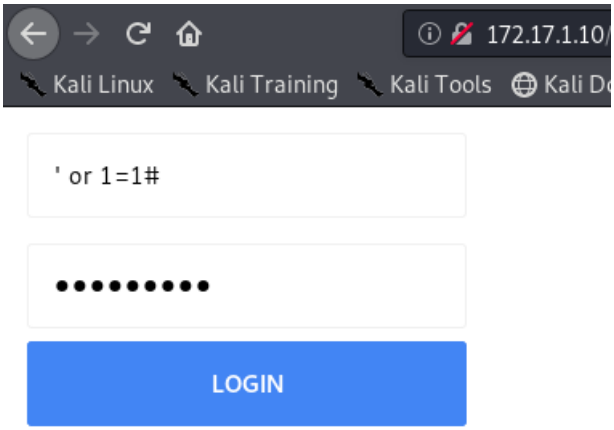

Figure 40. Ataque de SQL inyection.
Fuente: Autor

Como resultado del ataque se tiene la generación de una alerta de seguridad y por lo tanto la ejecución automática de la respuesta activa configurada.

De primera instancia se puede verificar en la interfaz de Kibana las alertas de seguridad:

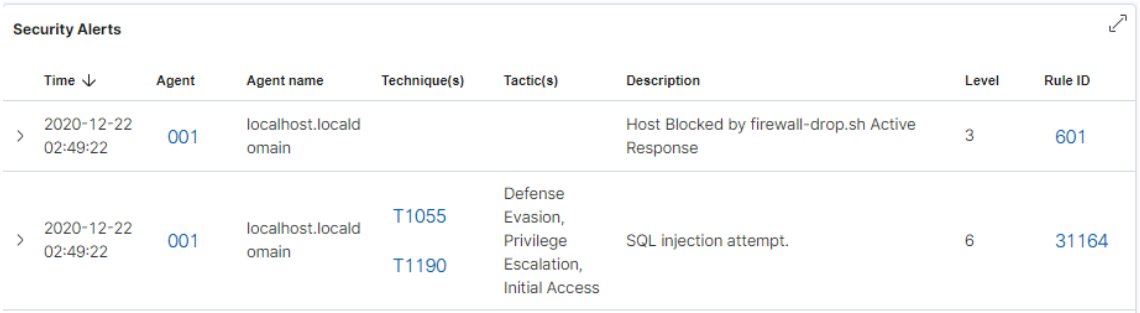

Figure 41. Alertas de seguridad en Wazuh – SQL injection.
Fuente: Autor

Por consecuencia la IP del atacante se ha bloqueado en el firewall del servidor de forma automática e inmediata ante la detección de intento de ataque por SQL injection en base al número identificador de la regla configurada para ejecutarse como respuesta activa establecida.

3.3.2. Ejecución de ataque DoS

El ataque de denegación de servicio DoS genera peticiones HTTP para saturar la respuesta del servidor web, este ataque se genera mediante Slowloris, un script basado en codigo Python a mayor detalle se explica en [42] donde facilita de igual manera su código.

Para este ataque se requiere la ejecución del script dirigido al servidor web, de manera que sea detectado por la solución de seguridad propuesta e inmediatamente genere la respuesta activa.

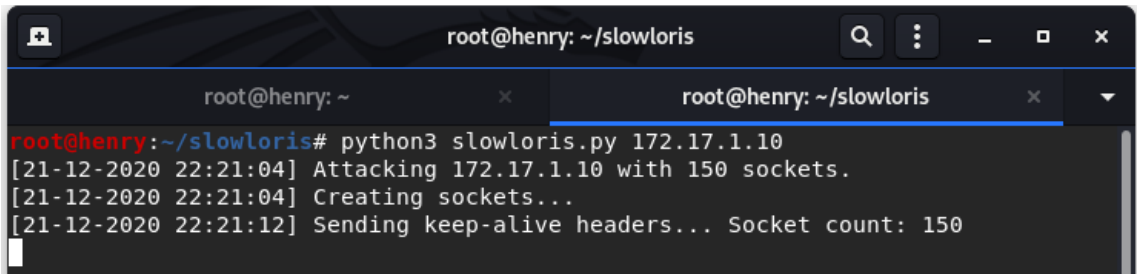

Figure 42. Ejecución de ataque DoS con Slowloris.
Fuente: Autor

Se validad en la interfaz de Kibana que la actividad generada al momento del ataque muestra como resultado lo siguiente:

2020-12-22 03:21:34	Host Blocked by firewall-drop.sh Active Response	3	601
2020-12-22 03:21:34	Web server 400 error code.	5	31101
2020-12-22 03:21:34	Web server 400 error code.	5	31101
2020-12-22 03:21:34	Web server 400 error code.	5	31101
2020-12-22 03:21:34	Web server 400 error code.	5	31101
2020-12-22 03:21:34	Web server 400 error code.	5	31101
2020-12-22 03:21:34	Web server 400 error code.	5	31101
2020-12-22 03:21:34	Web server 400 error code.	5	31101
2020-12-22 03:21:34	Multiple web server 400 error codes from same source ip.	10	31151

Figure 43. Alertas de seguridad en Wazuh – DoS.
Fuente: Autor

Donde se identifican los múltiples errores que generan las consultas parciales HTTP de los cuales provienen de una misma dirección IP.

En Figure 44 se muestra que el ataque es persistente a intervalos regulares, manteniendo los sockets y al volverlo a ejecutar este no tiene efecto al igual que la afectación que este tendría con el servidor web.

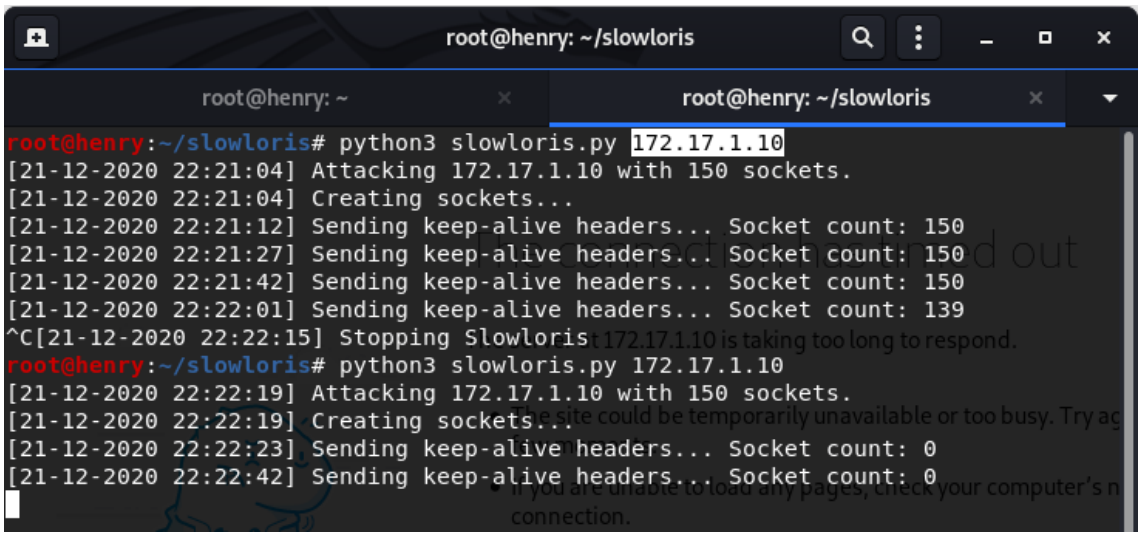

Figure 44. Ataque de DoS fallida.
Fuente: Autor

Si bien el ataque puede ser ejecutado por un momento hasta que sea detectado y mitigado de forma automática, gracias a la respuesta activa, el envio de cabeceras subLasecuentes en intervalos regulares ya no mantiene los sockets ya que se pierde conexión.

3.3.3. Ejecución de alertas

La prueba de concepto que se realiza trata sobre ejecutar uno de los ataques y verificar el funcionamiento de la respuesta activa.

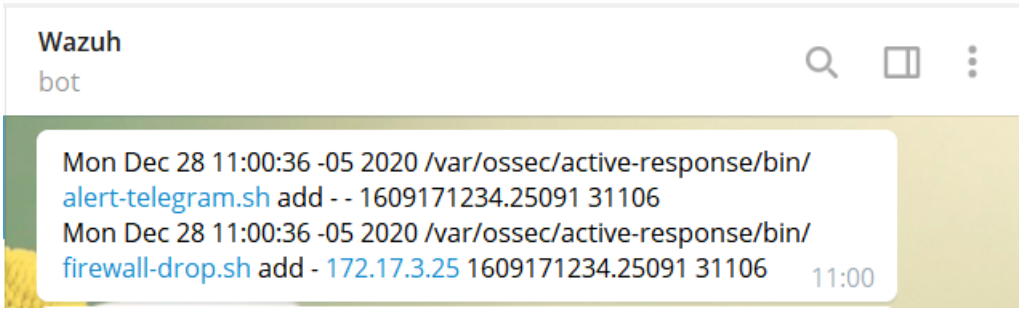

Figure 45. Alerta en Telegram
Fuente: Autor

En la Figure 45 se puede apreciar que la alerta corresponde a la ID de regla 31106 con la fecha y hora en la que se generó y ejecutando las respuestas activas, en este caso son dos, alert-telegram.sh y firewall-drop.sh. Proporcionando la alerta por Telegram y bloqueando la IP de host que está realizando el ataque respectivamente.

4. Conclusiones y Trabajos futuros

4.1. Conclusiones

En este trabajo se realizó una fundamentación teórica que establezca la propuesta de solución de seguridad basado en la detección de anomalías para host de una red como los son los servidores.

Las bondades que brinda Wazuh como IDS al momento de detectar anomalías es amplio, no solo por manejar un conjunto de reglas que detecta e identifica ataques, sino la forma en cómo se puede dar respuesta frente a los incidentes de seguridad que se pueden presentar. Por ello considero que las respuestas activas son una herramienta imprescindible a la hora de proponer una solución de seguridad.

La combinación de Elastic Stack y Wazuh con fines de proporcionar seguridad al ambiente empresarial se realizó exitosamente debido a las múltiples funciones y bondades de estas herramientas, gracias a la implantación de un ambiente de laboratorio se pudo validar el correcto funcionamiento de estos, donde la

implementación de Wazuh como pluging es una forma acertada para explotar todas las funciones que este brinda como IDS/HIDS.

La instalación que se realiza en esta solución es la más actual de momento hasta realizado este trabajo de fin de máster, por ello se considera que las soluciones a las seguridades incrementan cada día y de igual manera las técnicas para vulnerar y atacar sistemas críticos como son las BDD, es importante mantener la actualización de Wazuh sin antes considerar que esto puede afectar al cómo este trabaja en conjunto con Elastic Stack.

El ataque de DoS al realizar peticiones HTTP de forma parcial genera múltiples errores que son detectados por Wazuh por venir desde una misma fuente, de manera que mediante su detección la respuesta activa configurada, se ejecuta y agrega la regla de firewall del servidor web para la denegación por IP de origen del ataque detectado.

Los ataques de SQL injection simulados en este trabajo son basados en las más utilizados y simples en la práctica, pero sin embargo son su simpleza la que los hacen más peligrosos. Por ello es que se demuestra que la solución de Wazuh como HIDS y Elastic Stack corresponden a cualquier tipo de ataque mal intencionado.

Las respuestas activas que tienen la solución de seguridad Wazuh más Elastic Stack es bastante eficiente en cuanto al tiempo de respuesta con la que es detectado un ataque, es decir, que un administrador de una red realizando un monitoreo convencional de identificar ataques y bloquear las IPs del origen del ataque le tomará varios minutos, considerando que de cientos de posibles alertas que se pueden generar, la persona encargada las detecte y lo realice de forma manual. De manera que con esta solución es inmediata.

4.2. Trabajos futuros

Una vez que se han cumplido los objetivos planteados en este trabajo, se consideran a continuación algunas líneas de trabajo futuro para mejorar el proyecto:

- Proporcionar una mejor instalación de Wazuh enfocada únicamente en las necesidades de seguridad que son de interés, más no contar con alertas y información que se muestra la instalación por defecto con la finalidad de no sobrecargar el sistema, y eliminar posibles falsos positivos que se pueden considerar alertas, cuando en realidad son casos que no generan inseguridad en los sistemas.
- Considerar como parte del análisis para la detección de anomalías los equipos de red y proporcionar una respuesta activa frente a los posibles ataques que sean filtrados mediante listas de control de acceso (ACLs).
- En un futuro poder explotar al máximo las características de Wazuh y Elastic Stack en cuanto al análisis de una red de datos empresarial implementada.

Trabajos citados

[1] E. María, “The Top 12 Data Breaches of 2019,” 2019. https://www.securitymagazine.com/articles/91366-the-top-12-data-breaches-of-2019 (accessed Sep. 20, 2020).

[2] J. L. Thames, R. Abler, and D. Keeling, “A distributed active response architecture for preventing SSH dictionary attacks,” in *Conference Proceedings - IEEE SOUTHEASTCON*, 2008, pp. 84–89, doi: 10.1109/SECON.2008.4494264.

[3] S. Khandani, “ENGINEERING DESIGN PROCESS Education Transfer Plan Prepared by,” 2005.

[4] P. Dymora and M. Mazurek, “An Innovative Approach to Anomaly Detection in Communication Networks Using Multifractal Analysis,” *Appl. Sci.*, vol. 10, no. 9, p. 3277, May 2020, doi: 10.3390/app10093277.

[5] J. Tomás *et al.*, “TFM: Monitorización de Seguridad con Wazuh,” Universitat Oberta de Catalunya (UOC), Dec. 2019. Accessed: Sep. 29, 2020. [Online]. Available: http://openaccess.uoc.edu/webapps/o2/handle/10609/107166.

[6] V. Polo Cózar, Javier; Canto Rodrigo, Pau del; García Font, “Implementación de Wazuh en una organización pública,” Universitat Oberta de Catalunya (UOC), 2020. Accessed: Sep. 29, 2020. [Online]. Available: http://openaccess.uoc.edu/webapps/o2/handle/10609/117787.

[7] Y. K. Penya and P. G. Bringas, “Experiences on designing an integral Intrusion Detection System,” in *Proceedings - International Workshop on Database and Expert Systems Applications, DEXA*, 2008, pp. 675–679, doi: 10.1109/DEXA.2008.54.

[8] R. T. Gaddam and M. Nandhini, “An analysis of various snort based techniques to detect and prevent intrusions in networks: Proposal with code refactoring snort tool in Kali Linux environment,” in *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2017*, Jul. 2017, pp. 10–15, doi: 10.1109/ICICCT.2017.7975177.

[9] K. Wong, C. Dillabaugh, N. Seddigh, and B. Nandy, “Enhancing Suricata intrusion detection system for cyber security in SCADA networks,” Jun. 2017, doi: 10.1109/CCECE.2017.7946818.

[10] “Wazuh · The Open Source Security Platform.” https://wazuh.com/ (accessed Sep. 28, 2020).

[11] R. Venkatesan, D. Rubidha Devi, R. Keerthana, and A. Arjun Kumar, “A novel approach for detecting DDOS attack in h-IDS using association rule,” Nov. 2018, doi: 10.1109/ICSCAN.2018.8541174.

[12] D. Teixeira, L. Assunção, T. Pereira, S. Malta, and P. Pinto, “OSSEC IDS Extension to Improve Log Analysis and Override False Positive or Negative Detections,” *J. Sens. Actuator Networks*, vol. 8, no. 3, p. 46, Sep. 2019, doi: 10.3390/jsan8030046.

[13] “Nagios - The Industry Standard In IT Infrastructure Monitoring.” https://www.nagios.org/ (accessed Sep. 28, 2020).

[14] J. Renita and N. E. Elizabeth, “Network’s server monitoring and analysis using Nagios,” in *Proceedings of the 2017 International Conference on*

Wireless Communications, Signal Processing and Networking, WiSPNET 2017, Feb. 2018, vol. 2018-January, pp. 1904–1909, doi: 10.1109/WiSPNET.2017.8300092.

[15] "Industry Leading Log Management | Graylog." https://www.graylog.org/ (accessed Sep. 28, 2020).

[16] A. Ruiz, A. Tutor, : Pau, D. Canto, R. Profesor, and H. R. Pous, "Centralización y análisis de eventos de seguridad con Graylog," Universitat Oberta de Catalunya (UOC), 2019. Accessed: Sep. 28, 2020. [Online]. Available: http://openaccess.uoc.edu/webapps/o2/handle/10609/107506.

[17] "Búsqueda open source: los creadores de Elasticsearch, el ELK Stack y Kibana | Elastic." https://www.elastic.co/es/ (accessed Sep. 28, 2020).

[18] S. J. Son and Y. Kwon, "Performance of ELK stack and commercial system in security log analysis," in *2017 IEEE 13th Malaysia International Conference on Communications, MICC 2017*, Mar. 2018, vol. 2017-November, pp. 187–190, doi: 10.1109/MICC.2017.8311756.

[19] P. Shukla, S. Kumar, S. Chhajed, and M. Ochoa, *Learning Elastic Stack 6.0 : a beginner's guide to distributed search, analytics, and visualization using Elasticsearch, Logstash and Kibana.* Packt Publishing, 2017.

[20] B. Dixit and R. Kuć, *Elasticsearch : a complete guide : end-to-end search and analytics : a course in three modules*. .

[21] A. Talaş, F. Pop, and G. Neagu, "Elastic stack in action for smart cities: Making sense of big data," in *Proceedings - 2017 IEEE 13th International Conference on Intelligent Computer Communication and Processing, ICCP 2017*, Nov. 2017, pp. 469–476, doi: 10.1109/ICCP.2017.8117049.

[22] A. Srivastava, Anurag, and author, *Kibana 7 quick start guide : visualize your Elasticsearch data with ease*. .

[23] U. H. Rao and U. Nayak, "Intrusion Detection and Prevention Systems," in *The InfoSec Handbook*, Berkeley, CA: Apress, 2014, pp. 225–243.

[24] C. Wu and R. Buyya, "Data Center Networks," in *Cloud Data Centers and Cost Modeling*, Elsevier, 2015, pp. 497–576.

[25] B. Kte' pi, "Servers.," *Salem Press Encycl. Sci.*, 2019, [Online]. Available: https://ezproxy.biblioteca-uoc.idm.oclc.org/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=ers&AN=94982046&site=eds-live&scope=site.

[26] C. Karayiannis, "The Apache Web Server," in *Web-Based Projects that Rock the Class*, Berkeley, CA: Apress, 2019, pp. 1–37.

[27] Netcraft, "September 2020 Web Server Survey," 2020. https://news.netcraft.com/archives/category/web-server-survey/.

[28] "Learning MySQL."

[29] R. BHARGAVA, "Log Monitoring and Analysis: Comparing ELK, Splunk and Graylog," 2018. https://devops.com/log-monitoring-and-analysis-comparing-elk-splunk-and-graylog/ (accessed Oct. 20, 2020).

[30] P. Anderson, "Nagios Syslog Server Review," 2018. https://www.netadmintools.com/syslog-server/nagios/ (accessed Oct. 20, 2020).

[31] H. S. Kumar, N. Kumar, and M. Devi, "Intrusion Detection System using ELK Stack," 2019, doi: 10.22214/ijraset.2019.6115.

[32] "Mejorar la analítica de Seguridad con el Elastic Stack, Wazuh e IDS | Elastic Blog." https://www.elastic.co/es/blog/improve-security-analytics-

with-the-elastic-stack-wazuh-and-ids (accessed Oct. 21, 2020).
[33] “Security Onion Documentation Release 2.3,” 2020.
[34] “Welcome to Wazuh · Wazuh 3.10 documentation.” https://documentation.wazuh.com/3.10/index.html (accessed Oct. 22, 2020).
[35] “Welcome to Wazuh · Wazuh 4.0 documentation.” https://documentation.wazuh.com/4.0/index.html (accessed Dec. 22, 2020).
[36] “Telegram Messenger.” https://telegram.org/ (accessed Dec. 28, 2020).
[37] “FAQ for the Technically Inclined.” https://core.telegram.org/techfaq#q-how-are-voice-calls-authenticated (accessed Dec. 28, 2020).
[38] “Telegram Bot Platform.” https://telegram.org/blog/bot-revolution (accessed Dec. 28, 2020).
[39] “Introducing Inline Bots.” https://telegram.org/blog/inline-bots (accessed Dec. 28, 2020).
[40] “Software | GNS3.” https://www.gns3.com/software (accessed Nov. 23, 2020).
[41] “Installation guide · Wazuh 4.0 documentation.” https://documentation.wazuh.com/4.0/installation-guide/index.html#installation-guide (accessed Dec. 20, 2020).
[42] “GitHub - gkbrk/slowloris: Low bandwidth DoS tool. Slowloris rewrite in Python.” https://github.com/gkbrk/slowloris (accessed Dec. 21, 2020).

5. Anexos

Anexo1. Configuración de routes en la red.

Router ISP:

```
Current configuration : 1300 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ISP
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip tcp synwait-time 5
```

```
!
!
ip cef
ip name-server 8.8.8.8
!
!
interface Ethernet0/0
 ip address 192.168.2.2 255.255.255.0
 ip nat outside
 ip virtual-reassembly
 half-duplex
!
interface Ethernet0/1
 ip address 172.17.0.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 half-duplex
!
interface Ethernet0/2
 no ip address
 ip nat outside
 ip virtual-reassembly
 shutdown
 half-duplex
!
interface Ethernet0/3
 no ip address
 shutdown
 half-duplex
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 192.168.2.1
ip route 172.17.1.0 255.255.255.0 172.17.0.2
ip route 172.17.2.0 255.255.255.0 172.17.0.2
!
!
ip nat inside source list 1 interface Ethernet0/0 overload
!
access-list 1 permit 172.0.0.0 0.255.255.255
access-list 1 permit 172.17.0.0 0.0.255.255
no cdp log mismatch duplex
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
```

```
End
```

Router_Firewall:

```
Current configuration : 981 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router_Firewall
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip tcp synwait-time 5
!
!
ip cef
ip name-server 8.8.8.8
!
!
interface Ethernet0/0
 ip address 172.17.2.1 255.255.255.0
 half-duplex
!
interface Ethernet0/1
 ip address 172.17.0.2 255.255.255.0
 half-duplex
!
interface Ethernet0/2
 ip address 172.17.1.1 255.255.255.0
 half-duplex
!
interface Ethernet0/3
 ip address 172.17.3.1 255.255.255.0
 half-duplex
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 Ethernet0/1
!
!
no cdp log mismatch duplex
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
```

```
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end
```

Anexo2. Aplicación web

Conexión a BDD

```
<?php

$hostname = "localhost";
$username = "uocuser";
$password = "Is@.2013";
$dbname = "prac";

try {
    $connect = new PDO('mysql:host=localhost;dbname=prac', $username,
$password);
    $connect->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
   echo 'ERROR: ' . $e->getMessage();
}

?>
```

Login

```
<?php
$hostname = "localhost";
$username = "uocuser";
$password = "Is@.2013";
$dbname = "prac";
$conn = mysqli_connect($hostname, $username, $password, $dbname);
if(!$conn) {
        die("No conectado");
}
if($_GET) {
        $uname = $_GET["Email"];
        $pass = $_GET["Password"];

        $sql = "SELECT * FROM Users WHERE Email = '$uname' AND Password =
'$pass'";
        $result = mysqli_query($conn, $sql);
        if(mysqli_num_rows($result) == 1) {
                echo "Welcome, user!";
        } else {
                echo "Incorrect Username/Password";
```

```
        }
}
?>
<!DOCTYPE html>
<html>
<head>
        <title>Login Portal</title>
        <style type="text/css">
                input[type=text],input[type=password] {
                        padding: 16px;
                        margin: 8px;
                        border: 1px solid #f1f1f1;
                        letter-spacing: 1px;
                        border-radius: 3px;
                        width: 240px;
                }
                input[type=submit] {
                        margin-left: 8px;
                        width: 274px;
                        border-radius: 3px;
                        border: 1px solid #4285f4;
                        background-color: #4285f4;
                        padding: 16px;
                        color: white;
                        font-weight: 600;
                        cursor: pointer;
                }
        </style>
</head>
<body>

                <form action method="GET">
                <input   type="text"   name="Email"   placeholder="Username"
/><br />
                <input           type="password"           name="Password"
placeholder="********" /><br />
                <input type="submit" name="login" value="LOGIN" />
        </form>
</body>
</html>
```

Sección de noticias

```
<?php

include("config.php");

$Id=$_GET['id'];
$stmt = $connect->prepare("SELECT * FROM News WHERE Id=".$Id);


try {
      $stmt->execute();
} catch(PDOException $e) {
   echo 'ERROR: ' . $e->getMessage();
}

while($sa = $stmt->fetch(PDO::FETCH_ASSOC)) {
```

```
echo "<h1>T&iacutetulo: ";
echo $sa['Title'];
echo "</h1><br>Fecha: <i>";
echo $sa['Datetime'];
echo "</i><h3 align=justify width=160>";
echo $sa['Body'];
echo "</h3><h4>by Henry Farinango</h4>";

$uno = 1;
$siguiente = $Id + $uno;
$atras = $Id - $uno;

echo "<br><a href=\"news.php?id=$sa[Id]\">Read More...</a>
|| <a href=\"news.php?id=$siguiente\">Siguente</a>
|| <a href=\"news.php?id=$atras\">Atras</a><br><hr>";}

?>
```

Anexo3. Respuesta Activa para envio de alertas a Telegram

```
#!/bin/sh
# Author: Henry Farinango

UNAME=`uname`
ECHO="/bin/echo"
GREP="/bin/grep"
GENFILT="/usr/sbin/genfilt"
LSFILT="/usr/sbin/lsfilt"
MKFILT="/usr/sbin/mkfilt"
RMFILT="/usr/sbin/rmfilt"
PWD=`pwd`


ACTION=$1
USER=$2
IP=$3
ALERTID=$4
RULEID=$5


LOCAL=`dirname $0`;
cd $LOCAL
cd ../
filename=$(basename "$0")

LOG_FILE="${PWD}/../logs/active-responses.log"

echo "`date` $0 $1 $2 $3 $4 $5" >> ${LOG_FILE}


# Getting alert time
ALERTTIME=`echo "$ALERTID" | cut -d  "." -f 1`
```

```
# Getting end of alert
ALERTLAST=`echo "$ALERTID" | cut -d  "." -f 2`

# Getting full alert
ALERT=`grep -A 5 "$ALERTTIME" ${LOG_FILE} | grep -v ".$ALERTLAST: " -A 5`


curl -s \
-X POST \
https://api.telegram.org/bot1438618574:AAHrt62UesDt8L9cRJWfqRpUHFdgzhTS3lI/
sendMessage \
-d text="$ALERT" \
-d chat_id=765515016
```