UPN
UNIVERSIDAD PRIVADA DEL NORTE

FACULTAD DE INGENIERÍA

Carrera de **INGENIERÍA DE SISTEMAS COMPUTACIONALES**

“IMPLEMENTACIÓN DE UNA PLATAFORMA SERVER UTILIZANDO LA ARQUITECTURA SURICATA OPEN SOURCE PARA LA DETECCION Y PREVECCIÓN DE INTRUSOS EN LA RED”

Tesis para optar al título profesional de:

Ingeniero de Sistemas Computacionales

Autor:

Mario Jesus Farro Cachay

Asesor:

Dr. Ing. Laura Sofía Bazán Díaz
https://orcid.org/0000-0001-6377-8328

Lima - Perú

2023

JURADO EVALUADOR

Jurado 1 Presidente(a)	**BOJORQUEZ SEGURA JORGE**	**10318709**
	Nombre y Apellidos	Nº DNI

Jurado 2	**NARVAEZ VILLACORTA JORGE**	**41455569**
	Nombre y Apellidos	Nº DNI

Jurado 3	**REYES RODRIGUEZ EDUARDO MARTIN**	**41212791**
	Nombre y Apellidos	Nº DNI

INFORME DE SIMILITUD

Informe tesis final

INFORME DE ORIGINALIDAD

13%	12%	1%	4%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	red.com.mx Fuente de Internet	1%
2	1library.co Fuente de Internet	1%
3	openaccess.uoc.edu	1%

DEDICATORIA

A mis padres por ser ejemplo de perseverancia y por su confianza en mí educación, son mi fortaleza para continuar con mis objetivos profesionales.

A mi esposa e hija que ahora son mi motor y motivo para ser un padre ejemplar en todo los aspectos de la vida.

AGRADECIMIENTO

A Dios por mantenerme con buena salud y por ser mi guía por el camino correcto.

A mis maestros y a mi alma mater Universidad Privada del Norte por la formación y educación en transmitir los conocimientos para formar futuros profesionales.

Tabla de contenido

Índice de tablas

Índice de figuras

RESUMEN

El presente trabajo de investigación tuvo como objetivo general implementar una plataforma Server utilizando la arquitectura Suricata Open Source para la detección y prevención de intrusos en la red. El enfoque de investigación fue cuantitativo, de tipo aplicada, que buscó resolver el problema en seguridad y protección de datos de manera alternativa con nuevas herramientas, brindando el aporte científico basado en el conocimiento. La muestra estuvo conformada por doce personas del área TI, correspondiente a 12 empresas, donde se tuvo en cuenta el diseño pre experimental. Con respecto a la evaluación de la implementación como herramienta de seguridad. Se logró implementar una plataforma Server utilizando la arquitectura Suricata Open Source para la detección y prevención de intrusos en la red, donde los resultados demostraron con la prueba no paramétrica de Wilcoxon un valor p de 0.317 que indicó que la implementación presentó la conformidad para el IDS/IPS. Con estos resultados se concluyó que la implementación de la plataforma server Suricata Open Source para el IDS/IPS cumplió con su conformidad por ser de gran utilidad en su implementación por su eficacia, desempeño, alcance y accesible demostrando su efectividad en la detección y prevención de intrusos en una red.

PALABRAS CLAVES: Seguridad en redes informáticas, Detección y prevención de intrusos, IDS Open Source,

CAPÍTULO I: INTRODUCCIÓN

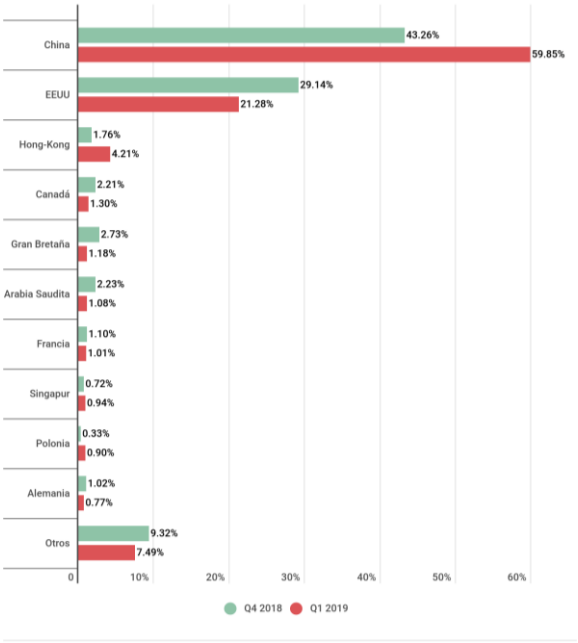
1.1. Realidad problemática

En la actualidad la seguridad en los centros de datos es de gran importancia para la protección de cualquier empresa, sin embargo, su alto costo en licencias o desinformación acerca de la importancia hace dejar de lado este aspecto. En este contexto, el software Open Source Suricata surge en la industria de la tecnología como una alternativa que facilita la detección y prevención de intrusos. Con el paso del tiempo, Internet ha evolucionado en diferentes ámbitos y su expansión también ha traído consigo un aumento en el número y la intensidad de ataques a nivel mundial; se requiere un alto nivel de seguridad para proteger los recursos de la red, datos y comunicaciones Esto presenta una gran responsabilidad para cualquier compañía (Eldow et al., 2016).

Los ataques cibernéticos están a la orden del día. Sin embargo, es necesario contar con sistemas sofisticados que eviten los ataques DDoS y demás variantes. Los informes del año 2019, la compañía de seguridad informática más conocida por su antivirus Kaspersky detalla un aumento en los ataques DDoS entre el año 2018 y el primer trimestre del 2019. En este periodo, China alcanzó el 67.89% de ataques, seguido de Estados Unidos 17.17%, y Hong Kong con 4.81%. Países latinoamericanos como Brasil dejó del TOP 10, lo que significa que otros países tampoco sean ajenos a los ataques cibernéticos. La figura 1 muestra el top 10 de ataques DDOS del primer trimestre 2019.

Figura 1

Resultado de la cantidad de ataques DDoS por país.



Fuente: Kaspersky (2019).

Los ataques cibernéticos son acontecimientos que suceden en cualquier momento, cuando un ciberatacante toma posición del sistema, es capaz de robar información confidencial, a cambio de dinero o incluso dañar el sistema por completo. En ciertos casos, los hackers e intrusos utilizan los recursos del sistema, como CPU, memoria, ancho de banda entre otros, para llevar a cabo su cometido delictivo (Ramírez, 2009).

El mayor crecimiento de dispositivos conectados a Internet se ha venido multiplicando en el tráfico IP en los últimos años, lo que ha provocado congestión en la red en varios países de Europa y Latinoamérica, notándose haber temido repercusión que afecta a instituciones públicas y privadas. Según un informe de seguridad Global de Radware, identifica el excedente de ancho de banda y el retraso en la entrega de paquetes son causados por ataques DDOS (Sánchez, 2017).

Linux cuenta con muchas distribuciones preparadas para ser usadas como servidores. Sin embargo, se convierte en un desafío para los administradores de red poder elegir la distribución más adecuada para la implementación de servicios en red. En 2014, el gobierno de Venezuela, con apoyo gubernamental, creó una distribución de GNU/Linux que se utilizó a nivel público; otro país como Alemania desarrolló sus propias distribuciones. En los últimos años, se ha venido difundiendo el uso de software libre como Linux. No obstante, surge la pregunta de por qué elegir una distribución en lugar de otra, esto genera interrogantes sobre si la distribución elegida ofrece madurez en la seguridad siendo necesaria para garantizar actualizaciones regulares. Los problemas de seguridad se generan por un asesoramiento incorrecto o por elegir una distribución equivocada que normalmente no debería ser usada como servidor. Esto pone en alto riesgo los datos manejados en en una distribución Linux que no fue elegida correctamente (Badillo, 2015).

La ciberseguridad en Latinoamérica no es ajena a los problemas de ataques, como antecedentes principales e importantes ha hechos reales, en distintos países se han visto afectados por un crecimiento de riesgos de seguridad; Chile fue víctima de los ataques dirigidos a entidades financieras robando millones de dólares y divulgando datos de más de 67 mil tarjetas de crédito ocasionado por un grupo hacktivista conocido como Shadow Brokers (Cifuentes, 2018).

En Perú, la ciberseguridad ha tomado mayor relevancia dentro de las empresas debido a los daños que podrían ocasionar en sus operaciones. Según la empresa de ciberseguridad Bafing, el presupuesto destinado en ciberseguridad de TI en las empresas peruanas se encuentra entre el 4% o 5%, una cifra variable al tipo de empresa, además se estima un crecimiento de 7% del presupuesto total de TI de cada empresa. Sin embargo, las

pymes y medianas empresas suelen carecer de inversión en seguridad, a pesar de los conocidos casos de empresas que fueron afectadas por hackers (Diario Gestión,2017).

1.2. Formulación del problema

¿De qué manera la implementación de una plataforma Server, utilizando la arquitectura Suricata open Source cumple con la prevención y detección de intrusos en una red perimetral?

1.2.1 Problemas específicos

- ¿Cuál es la conformidad del software actual para el IDS/IPS en la red?
- ¿Cómo implementar una plataforma Server utilizando la arquitectura Suricata Open Source?
- ¿Cuál es la conformidad de la implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS?

1.3. Objetivos

1.3.1. Objetivo general

Implementar una plataforma Server utilizando la arquitectura Suricata Open Source, para la detección y prevención de intrusos en la red.

1.3.2. Objetivos específicos:

- Determinar la conformidad del software actual para IDS/IPS en la red.
- Diseñar e implementar una plataforma Server utilizando la arquitectura Suricata Open Source.
- Determinar la conformidad de la implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS.

1.4. Hipótesis

1.4.1. Hipótesis general

La implementación de una plataforma Server utilizando la arquitectura Suricata Open Source cumple significativamente con la detección y prevención de intrusos.

1.4.2. Hipótesis especificas

- El software actual para IDS/IPS en la red es conforme.
- Se logra implementar la plataforma Server utilizando la arquitectura Suricata Open Source.
- La implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS es conforme.

1.5. Marco teórico

1.5.1. Antecedentes Internacionales:

Según Perdigón (2022), en Cuba, en su paper "Suricata como detector de intrusos para la seguridad en red de datos empresariales", tuvo como objetivo analizar la importancia y su gran utilidad basado en software libre como sistema de detección y prevención de intrusos para fortalecer la seguridad en las redes empresariales. La investigación fue descriptiva identificando la importancia, características, funcionalidades y beneficios de la investigación, además de utilizar el método experimental para su implementación. Como resultado, se comprobó su efectividad para detectar ataques de tipo DoS, fuerza bruta entre otros. La implementación Suricata contribuyó en identificar comportamientos anómalos y un consumo de recursos de hardware eficiente. Se concluyó que Suricata reduce costos además de ser una solución viable y accesible en inversión y rendimiento para las pymes.

Según Rangel (2021), en México, en su tesis “Sistemas de detección de intrusos y gestión de eventos e información de seguridad basados en nuevas tecnologías de código abierto”, tuvo como objetivo documentar la implementación del sistema de detección de intrusos (IDS) y un sistema de gestión de eventos e información de seguridad (SIEM) en efecto logró detectar, analizar eventos anómalos manteniendo una red controlada. Se realizó una investigación documental recopilando información sobre posibles amenazas. En el resultado los IDS permitieron identificar las amenazas a través de sus sensores y enviar alertas de seguridad basado y configurado en reglas. El sistema SIEM proporciono control en tiempo real ante posibles ataques. Se concluyó con la elección de IDS Suricata como herramienta de análisis en la detección y prevención de intrusos, destacando su ventaja en actualizaciones constantes y de mantener una defensa muy activa. Además, se utilizó la solución de código abierto ELK Stack para la gestión de eventos del SIEM, lo que permitió un control efectivo el monitoreo.

Para Cabrera (2022), en Ecuador, en su tesis “Diseño de una red de seguridad perimetral basada en Open Source para aplicación de IDS e IPS para el control de amenazas informáticas en la universidad técnica de Babahoyo”, tuvo como objetivo mejorar la seguridad en la red de la universidad UTB con fines de garantizar la protección de la información. La metodología fue cualitativa, con el método deductivo. Como resultado se evidenciaron mejoras en el tráfico y análisis de la red en sus diferentes facultades; en efecto la administración de sus redes fue más controlada con IDS e IPS. Se concluyó destacando la importancia del diseño en una red perimetral basada en Open Source en sus mejoras de seguridad, además de mantener al administrador informado ante algún evento o amenaza informática.

Según Da Costa Calado (2018), en Portugal, en su tesis "Open source IDS/IPS in a production environment: comparing, assessin", tuvo como objetivo evaluar la posibilidad de crear un entorno de defensa alternativa al resto de fabricantes licenciados, contando con la versatilidad y funcionalidad desde el sistema operativo. Se desarrolló una metodología de investigación descriptiva con diseño experimental donde se realizó un enfoque comparativo de Snort y Suricata. Mediante un laboratorio, se analizó la capacidad de respuesta de ambas herramientas en diferentes escenarios. Se comprobó que tanto Snort como Suricata son suficientemente confiables, personalizables para adaptarse al hardware con menos de 1GB de RAM. Sin embargo, en términos de rendimiento y capacidad de respuesta, se concluyó que, existen similitudes entre ambas, también hay diferencias significativas. En su arquitectura Suricata supera ampliamente a Snort, a pesar de los intentos de afinación y configuración.

Según Miño (2020), en Ecuador, tuvo como objetivo diseñar e implementar un sistema de detección de intrusos (IDS) para la plataforma OpenStack utilizando código abierto. Se desarrolló un diseño cuasiexperimental en la elección de métodos, normativas para la creación de la plataforma de seguridad, además la investigación fue de tipo descriptiva y aplicada basándose en conocimientos previos y existentes en el ámbito de seguridad de datos. Como resultado, se evaluó el IDS en dos escenarios diferentes, en el nodo de control se consiguió una eficiencia del 99.96% en comparación al 21.39% del escenario virtualizado en servicio de nube OpenStack. Las pruebas se realizaron con tres tipos de ataques DDoS, SQL Injection, Buffer Overflow. Se concluyó mediante el análisis y comparación que el nodo o máquina virtual resultó ser más eficiente. Se obtuvo un resultado

estadístico en la eficiencia de detección de ataques con un promedio del 62.8 % en un escenario IDS directamente al nodo para el monitoreo del tráfico de red.

1.5.2. Antecedentes Nacionales

Según Dios y Ortiz (2018), en Trujillo, en su tesis "Diseño lógico y simulación de una red espejo virtual (honeynet) para la detección de intrusos informáticos en zona perimetral, el objetivo fue elaborar un algoritmo para la detección de intrusos informáticos mediante una red espejo virtual (Honeynet) para resguardar la zona perimetral en la Universidad Nacional de Trujillo. Los autores aplicaron el método deductivo para el desarrollo del proyecto, en efecto se utilizaron la recolección de datos, análisis de datos, implementación, pruebas y conclusiones. La población y la muestra estuvieron conformados por la Dirección de Sistemas y Comunicaciones de la universidad. Como resultado, se logró resguardar eficazmente la información de datos en la infraestructura de la red mediante un control y monitoreo de tráfico. Se concluyó con la verificación del bloqueo de tráfico malicioso y de la cabecera, que contienen los paquetes, en la simulación de un diseño actual y el diseño lógico.

Según Coyla (2019), en Juliaca, en su tesis tuvo como objetivo implementar un sistema de detección y prevención de intrusos (IDS/IPS), basado en la norma ISO 27001, para el monitoreo perimetral de la seguridad informática, en la red de la Universidad Peruana Unión". Se realizó una investigación de tipo aplicada, basada en los hallazgos y experiencias de los aportes tecnológicos en las empresas. Como resultado se consiguió una visualización para el monitoreo, además se consiguió una mejor interpretación del tráfico en la red de la universidad. Además, se mejoró el análisis y bloqueo con respecto a las reglas configuradas. Se concluyó que la herramienta Snort es de gran utilidad en la seguridad de la universidad,

también se notó una mejora significativa por el uso de la metodología ISO27001 en la calidad de implementación conformada por cuatro etapas: planeación, implementación, medición y mejoramiento.

Según Ochoa (2019), en Lima, en su tesis, tuvo como objetivo diseñar y elaborar un plan para una red de seguridad informática para la protección del sistema web de un call center ante ataques informáticos aplicando la norma ISO 27033. Se realizó un estudio de investigación explicativo, identificando el problema de las amenazas cibernéticas y en mitigar el impacto a la red. La población estuvo conformada por la empresa y la muestra fue el Call Center. En cuanto a los resultados del estudio, se identificó que un 41% fue de nivel crítico en los activos de la información que ponen en riesgo su situación financiera, un 18% fue de nivel alto que comprometen sus operaciones, 18% fue de nivel de criticidad medio y un 23% presenta un nivel bajo. Se concluyó con la etapa de planeación del proyecto en el diseño de la red de seguridad informática, adecuando la protección en el sistema web del Call Center y su interacción con los usuarios. Además, se observó que la norma 27033 otorgó recomendaciones mínimas para la gestión de riesgos, por tanto, el autor recomienda revisar la ISO 27001 por tener un mejor alcance en el tratamiento de riesgos.

Según Riveros (2019), en Huancayo, en su tesis, tuvo como objetivo implementar las políticas de la seguridad informática para mejorar el acceso y seguridad lógica de la red en la oficina departamental de estadística e informática de Junín. Se utilizó la metodología Top Down y se realizó una investigación aplicada, buscando entender los hechos usando los conocimientos para implementar políticas de seguridad. El resultado después de aplicar la metodología Top Down permitió un mejor análisis, diseño y desarrollo en la infraestructura de la red. Se concluyó que al implementar políticas de seguridad informática los usuarios

notaron cambios de navegación disminuyendo latencia de 43% además, hubo disminución de quejas en un 40% lo que ayudó en la optimización y control de la red.

Según Pintacur (2019), en Lima, su tesis tuvo como objetivo el diseño e implementación de una red de datos y seguridad perimetral de la empresa Corporación Cayman S.A.C. En el desarrollo metodológico de la investigación, se realizó un análisis en servidores y estaciones de clientes. Como resultado, se mejoró la infraestructura del datacenter en términos de administración y seguridad. Después de realizar el diseño de red para la empresa corporación Cayman S.A.C, se logró tener acceso controlado de sus servicios, también se implementaron cortafuegos para una mayor seguridad. Se concluyó sobre la importancia de implementar el cortafuego, IDS/IPS, antivirus, proxys, ancho de banda, etc. Además, se comprobó la operatividad de la DMZ e implementación de la red LAN mediante reglas de seguridad.

DEFINICIONES CONCEPTUALES:

Código abierto (Open Source) o Software Libre

Open Source conocido como software libre está basado en una gran cantidad de líneas programables o código fuente que es de libre acceso y libertad para ser modificado, personalizado y optimizado por usuarios programadores. A lo largo del tiempo se han desarrollado distintas distribuciones usando el código abierto a nivel mundial algunos de ellos podemos nombrar como Centos, Debian, Fedora y Ubuntu. El S.O Kali Linux es una distribución basada principalmente para la auditoria y seguridad informática en forma general (Badillo 2015).

IPS (Sistema de Prevención de Intrusos)

Es un software o dispositivo que se encarga de analizar paquetes completos con la finalidad de prevenir ataques, cuando encuentra un evento anómalo es capaz de alertar, rechazar y bloquear en tiempo real, tomando acciones para mantener estable la red de datos. Los IPS se encuentra en la clasificación de lo siguiente (Hernández y Santana, 2018):

- Se puede definir como un sistema activo y/o pasivo, que realiza alguna acción (drop, reject, alert, pass)
- Detección de políticas de seguridad: El IPS requiere la declaración de las políticas de seguridad.
- Detección basada en anomalías: En función al comportamiento de patrones del tráfico.

IDS (Sistema de Detección de Intrusos)

Es un software o dispositivo que analiza el tráfico o comportamientos sospechosos, se puede definir como un sistema pasivo, revisa el contenido y comportamientos sospechosos, realiza el escaneo de puertos, paquetes entrantes, etc. El IDS no es capaz de detener ataques por sí solo, su integración con un IPS es capaz de valerse como herramienta poderosa aplicando el bloqueo. Generalmente los IDS su función es generar mensaje log ante un evento inusual, realiza la comparativa con firmas en su base de datos sobre ataques conocidos, también toma la acción de monitorear y compartir datos al administrador de red ante eventos sospechosos (Rodríguez 2016).

SELKS

Es una distribución de código abierto basado en Debian, disponible para ser usado como un sistema de detección y prevención de intrusos (IDS e IPS) es de fácil despliegue para una red perimetral, además, no consume grandes recursos. Cuenta con librerías necesarias para obtener un sistema robusto, dentro de ello se integra los siguientes softwares (Dobladez 2019):

- Suricata: Software de prevención y detección de intrusos.
- ElasticSearch: Busca y analiza todos los logs antes de ser alojado en su base de datos.
- Logstash: Recopila y transforma todos los logs de forma centralizada.
- Kibana: Es un Dashboard que permite visualizar la información.

Suricata

Es un software Open Source de código abierto. Sus funciones principales incluyen la detección y prevención de intrusos en la red, realiza monitoreo de seguridad en la red, análisis de archivos y el registro de tráfico entre otros.

Destaca entre los demás IDS/IPS con sus características en Multi-Treading soporta sistemas multicore y multiprocesador de alto rendimiento ofreciendo mayor eficiencia en el análisis de tráfico de red, además cuenta con herramientas de gestión y actualizaciones de reglas. El motor está diseñado para procesar la mayor potencia de procesamientos por los últimos chips de CPU multinúcleo. Cuenta con un analizador HTTP, en la identificación, extracción y registros de archivos (Agra, 2017, p.27).

NMAP:

Es una herramienta multiplataforma, permitiendo rastrear puertos abiertos evaluando la seguridad de sistemas informáticos, también permite visualizar los servicios en función y las vulnerabilidades que presenta (de Haro Bermejo, 2015).

Denegación de servicio (DDOS):

Es un ataque que envía una gran cantidad de peticiones a un servidor en consecuencia sobrecargando el ancho de banda o sobrepasando la capacidad de procesamiento a tal punto que los servicios colapsen, interrumpiendo así los servicios de una organización (Dios y Ortiz, 2018, p.31).

CAPÍTULO II: METODOLOGÍA

2.1. Tipo de investigación

La presente investigación fue de enfoque cuantitativo, de tipo aplicada, pues se refiere en la recolección de datos aportando información a las futuras investigaciones, además aporta conocimientos científicos a la problemática (Perdigón, 2022). Por otro lado, Janampa et al. (2021) identificó que las PYMES carecen de un sistema de seguridad para la prevención de intrusos en la red de datos, asimismo mencionó que el principal objetivo de la investigación es analizar la pertinencia y lo aplicable que es Suricata Open Source sin preocuparse por los altos costos de inversión. La presente investigación fue de diseño pre experimental, consistió en tener un acercamiento al problema de investigación que permitió conocer el efecto de una o más variables (Acurio y Rimachi, 2019).

2.2. Población y muestra

Población:

La presente investigación representó a 12 pequeñas y medianas empresas dela ciudad de Lima; según el reporte de INEI, en el primer trimestre 2021 se crearon 26877 empresas, cifra que representa el 39% del total nacional; en Lima se registró 36,6 % del total, lo que indica que de cada 10 nuevas empresas en el país cuatro inician actividades de comercio en la ciudad de Lima.

Muestra:

Siendo un análisis donde la población es infinita y de difícil acceso debido a restricciones de seguridad, la elección por una muestra seleccionada fue por conveniencia de tipo no probabilística. Esta nuestra estuvo representada por 12 personas especialistas en el área de sistemas que aceptaron brindar información para el desarrollo de la implementación.

A continuación, en las tablas 1 y 2 se muestra la operacionalización de las variables.

Tabla 1

Variable Independiente: Implementación de una plataforma Server utilizando la arquitectura Suricata Open Source

Variable Independiente	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores
Implementación de una plataforma Server utilizando la arquitectura Suricata Open Source	Es el diseño del cimiento conformado por el hardware del servidor y del software del sistema operativo, para su implementación y ejecución en la administración de la red (Sánchez, 2017).	Para la implementación de la arquitectura Suricata Open Source se instala en el servidor Linux y se configura con reglas operacionales llamado sensores que serán los encargados de analizas y detectar eventos anómalos	Diseño	Elaboración del diagrama de red
			Ejecución	Desarrollo y configuración
			Producción	Resultados e informes

Tabla 2

Variable Dependiente: Detección y Prevención de Intrusos en la red

Variable Dependiente	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores	Ítem
Detección y Prevención de Intrusos en la red	El sistema IDS observa, analiza e inspecciona el tráfico entrante ante eventos inusuales en la seguridad. El sistema IPS como software se encarga de supervisar el tráfico con la finalidad de prevenir ataques, alertar, rechazar y bloquearlos en tiempo real, para garantizar la fiabilidad, funcionalidad y seguridad de la red (Rodríguez, 2016)	Para evaluar el riesgo y seguridad, se aplicó un cuestionario en donde se formula preguntas al administrador sobre la importancia de contar con un sistema de detección. Para evaluar el riesgo y seguridad, se aplicará un cuestionario en donde se formula preguntas al administrador sobre la importancia de contar con un sistema de prevención de intrusos.	Fiabilidad	**Conformidad sobre:** Eficiente	1,2,3
				Efectivo	
			Seguridad	Confianza	4,5,6
				Utilidad	
			Flexibilidad	Actualización	7,8,9
				Monitoreo	
			Inspección	Registra eventos	10,11
				Identifica eventos	
			Funcionalidad	Supervisa trafico	12,13
				Realiza acciones	
			Observación	Tiempo de respuesta	14,15
				Resultado	

Nota. En esta tabla se muestra la variable dependiente Sistema de Prevención de Intrusos con sus dimensiones e indicadores, los ítems corresponden al instrumento (Anexo 1).

2.3. Técnicas e instrumentos de recolección y análisis de datos

2.3.1. Técnicas

La técnica utilizada fue la encuesta debido a la utilidad y objetividad, fue seleccionada como herramienta estratégica y de gran utilidad bajo su clasificación para el investigador (Carrasco, 2019, p.274).

2.3.2. Instrumentos

El cuestionario, que consta de 15 preguntas formuladas de acuerdo a las dimensiones e indicadores de las variables, fue elaborado como instrumento de investigación (anexo 1). Además, fue adaptado y validado por tres expertos en la materia, quienes completaron una ficha observacional, este proceso resulto en un coeficiente de valoración promedio del 94.7% (anexo 5). La confiabilidad se alcanzó un Alfa de Cronbach de 0,752 (Tabla 3), el resultado indica que el instrumento es más confiable.

Tabla 3

Estadística de fiabilidad

Estadísticas de fiabilidad	
Alfa de Cronbach	**N de elementos**
0.752	15

Fuente: SPSS V.26

2.3.3. Recolección y análisis de datos

En la recolección y análisis de datos se utilizó el software estadístico IBM SPSS Statistics v26.0 en su versión trial y el software Microsoft Excel. Se realizó la prueba de normalidad utilizando Shapiro-Wilk, por lo que contaba con una muestra menor a 50. Se obtuvo un valor de probabilidad (p) de 0.04 siendo menor al valor de significancia de 0.05, lo que confirma que los datos no siguen una distribución normal. Por lo tanto, se seleccionó la estadística no paramétrica para el análisis.

Tabla 4

Pruebas de normalidad Shapiro Wilk

	Estadístico	gl	p
Sistema de Detección y Prevención de Intrusos	0.852	12	0.04

Fuente: SPSS V.26

Aspectos éticos

En la presente investigación se respetaron las autorías referidas de diversas fuentes bibliográficas en el formato APA 7. Asimismo, los técnicos e ingenieros de diferentes empresas brindaron su colaboración en el llenado del cuestionario para realizar el análisis y recolección de datos, expresando la confidencialidad de sus datos para fines de investigación.

CAPÍTULO III: RESULTADOS

Los resultados se evaluaron mediante estadística descriptiva, utilizando la escala de Likert para la valoración de conformidad. Los niveles de escala son: 1 Totalmente desacuerdo, 2 En desacuerdo, 3 Ni de acuerdo ni en desacuerdo, 4 De acuerdo y 5 Totalmente de acuerdo, con relación a la variable dependiente y sus dimensiones.

ESTADISTICA DESCRIPTIVA

Objetivo General: Implementar una plataforma Server utilizando la arquitectura Suricata Open Source, para la detección y prevención de intrusos en la red.

Del análisis de datos de la tabla 5 se puede apreciar el resultado, donde se muestra el total de las respuestas en cinco niveles conseguidas en la escala de Likert, un total de 79 respuestas corresponde al nivel de totalmente de acuerdo y un total de 4 respuestas corresponde al nivel totalmente desacuerdo.

Tabla 5

Resultado de Sistema de Detección y Prevención de Intrusos

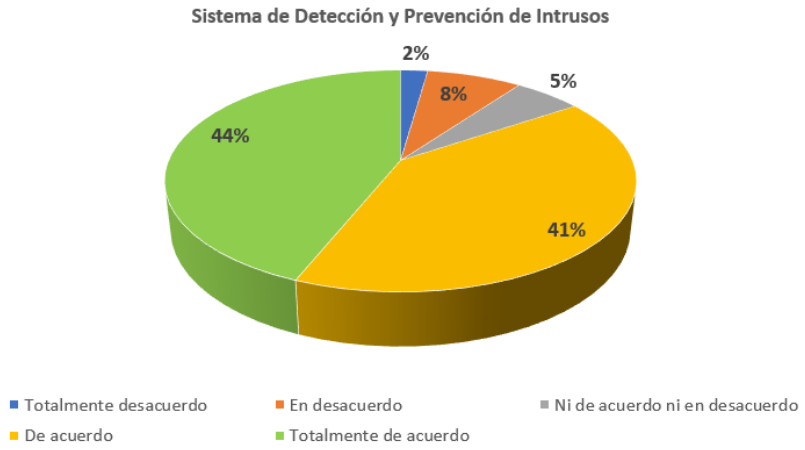
Sistema de Detección y Prevención de Intrusos				
		Total	**Porcentaje**	**Porcentaje Valido**
Valido	**Totalmente desacuerdo**	4	2%	2%
	En desacuerdo	14	8%	8%
	Ni de acuerdo ni en desacuerdo	10	5%	5%
	De acuerdo	73	41%	41%
	Totalmente de acuerdo	79	44%	44%
	Total	180	100%	100%

Fuente: SPSS V.26

En la figura 2, la totalidad de los encuestados fueron del área TI de diferentes pymes y medianas empresas, se evidenció que al 100% de un total de 180 respuestas de los evaluados un 44% considera estar totalmente de acuerdo, un 41% considera estar acuerdo mientras que un 2% consideró un totalmente desacuerdo, por lo tanto, el resultado demuestra un mayor porcentaje de estar totalmente de acuerdo en la implementación de una plataforma Server utilizando la arquitectura Suricata Open Source para el sistema de detección y prevención de intrusos en la red.

Figura 2

Resultado de Sistema de Detección y Prevención de Intrusos.



Para el **objetivo específico 1**: "Determinar la conformidad del software actual para IDS/IPS en la red.", se consiguió la media de las dimensiones del IDS/IPS que fueron fiabilidad, seguridad, flexibilidad, inspección, funcionalidad y observación.

Variable IDS/IPS:

Tabla 6

Resultado Pre-Test de Media en Dimensión Fiabilidad

	N	Media	Desv. estándar
Fiabilidad			
1. Considera usted que el software Open Source IDS sea eficiente en analizar en el tráfico entrante	12	4.33	.492
2. Considera usted que el software Open Source IDS sea efectivo como lo es un software IPS con licencia	12	4.17	.389
3. Considera usted que implementar software Open Source IDS demande de mucha inversión	12	2.83	1.337
N válido (por lista) y media total	12	4	

Fuente: SPSS V.26

El resultado conseguido por el cuestionario, donde N es la muestra, siendo el promedio 4 lo más relevante de la Tabla 6 en su dimensión fiabilidad, esto quiere decir que la conformidad se encuentra en un nivel de acuerdo, por tanto el software IDS/IPS Open Source es confiable y fiable como lo es como un software con licencia.

Tabla 7

Resultado Pre-Test de Media en Dimensión Seguridad

	N	Media	Desv. estándar
Seguridad			
4. Considera usted la importancia de la seguridad en su centro de datos	12	4.75	.452
5. Considera usted que el software Open Source IDS ayudaría a sus problemas de contrarrestar ataques cibernéticos	12	4.25	.622
6. Considera usted que un antivirus sea basto en salvaguardar la información de una pyme o mediana empresa	12	2.00	1.044
N válido (por lista)	12	4	

Fuente: SPSS V.26

El resultado conseguido por el cuestionario, donde N es la muestra, siendo el promedio 4 lo más relevante de la Tabla 7 en su dimensión seguridad, esto quiere decir que la conformidad se encuentra en un nivel de acuerdo por tanto el software IDS/IPS Open Source tiende a complementar la seguridad y en mitigar los ataques cibernéticos.

Tabla 8
Resultado Pre-Test de Media en Dimensión Flexibilidad

	N	Media	Desv. estándar
Flexibilidad			
7. Considera usted que el software Open Source IDS cuente con soporte de ayuda	12	3.83	.718
8. Considera usted que el software Open Source IDS cuente con actualizaciones constantes en su base de datos	12	4.33	.651
9. Considera usted que sería necesario implementar un software IDS en la empresa donde trabaja	12	4.58	.669
N válido (por lista)	12	4	

Fuente: Resultado obtenido en SPSS V.26

El resultado conseguido por el cuestionario, donde N es la muestra, siendo el promedio 4 lo más relevante de la Tabla 8 en su dimensión flexibilidad, esto quiere decir que la conformidad se encuentra en un nivel de acuerdo por tanto el software IDS/IPS Open Source mantiene actualizaciones y mejoras haciéndolo flexible para la implementación.

Tabla 9
Resultado Pre-Test de Media en Dimensión Inspección

	N	Media	Desv. estándar
Inspección			
10. Considera usted que el software Open Source IPS registra los eventos anómalos en el tráfico de paquetes entrantes	12	4.42	.515
11. Considera usted que el software Open Source IPS realice identificación de vulnerabilidades	12	4.42	.669
N válido (por lista)	12	4	

Fuente: SPSS V.26

El resultado conseguido por el cuestionario, donde N es la muestra, siendo el promedio 4 lo más relevante de la Tabla 9 en su dimensión inspección, esto quiere decir que la conformidad se encuentra en un nivel de acuerdo por tanto el software IDS/IPS Open Source registra e identifica eventos anómalos en el tráfico de red.

Dimensión Funcionalidad

Tabla 10
Resultado Pre-Test de Media en Funcionalidad

	N	Media	Desv. estándar
Funcionalidad			
12. Considera usted que el software Open Source IPS monitorea en tiempo real el trafico	12	4.83	.389
13. Considera usted que el software Open Source IPS realice acciones para contrarrestar los ataques	12	4.58	.515
N válido (por lista)	12	5	

Fuente: SPSS V.26

El resultado conseguido por el cuestionario, donde N es la muestra, siendo el promedio 5 lo más relevante de la Tabla 10 en su dimensión funcionalidad, esto quiere decir que la conformidad se encuentra en un nivel de totalmente de acuerdo por tanto el software IDS/IPS Open Source tiene la funcionalidad en el monitoreo real y en contrarrestar los ataques.

Tabla 11
Resultado Pre-Test de Media en Dimensión Observación

	N	Media	Desv. estándar
Observación			
14. Considera usted que el software Open Source IPS muestra los resultados en tiempo real	12	4.67	.492
15. Considera usted que el software Open Source IPS muestre resultados con gráficos mediante la integración con otro software	12	4.42	.669
N válido (por lista)	12	5	

Fuente: SPSS V.26

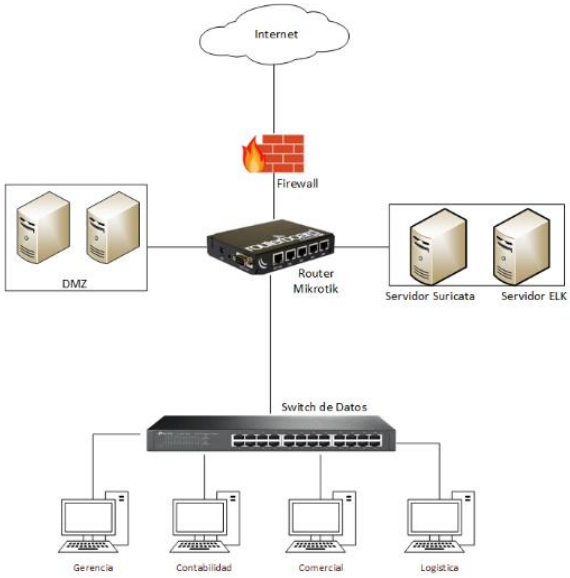
El resultado conseguido por el cuestionario, donde N es la muestra, siendo el promedio 5 lo más relevante de la Tabla 11 en su dimensión observación, esto quiere decir que la conformidad se encuentra en un nivel totalmente de acuerdo por tanto el software IDS/IPS Open Source muestra los resultados en el Dashboard siendo observacional y visual por los eventos ocurridos en el tráfico de la red.

Análisis: Los resultados del objetivo específico 1 determinaron la conformidad del software actual para IDS/IPS en la red, en las dimensiones de fiabilidad, seguridad, flexibilidad e inspección, los encuestados determinaron un nivel conformidad de 67% estar de acuerdo por otro lado las dimensiones de funcionalidad y observación determinaron un nivel de conformidad de 33% estar totalmente de acuerdo por lo tanto esto afirma que el software actual IDS/IPS en la red refuerzan la seguridad en un centro de datos.

Para el **objetivo específico 2** “Diseñar e implementar una plataforma Server utilizando la arquitectura Suricata Open Source.”, para el diseño e implementación se ha desarrollado un diseño pre experimental simulando un escenario de red real para una pyme o mediana empresa, se describe a continuación en la figura 3.

Figura 3

Diseño de esquema para el desarrollo IDS/IPS Suricata Open Source



Para ello se creó un laboratorio local usando el software de virtualización Oracle VM VirtualBox, también se instaló Ubuntu como servidor IDS e IPS, en estación cliente se usó Microsoft Windows 10, además de KaliLinux un sistema operativo en Linux para las pruebas y ataques para la realización de auditoria en seguridad, como servidor de base de datos se utilizó ElasticSearch para la recolección de datos y muestreo en un Dashboard.

El escenario constó de los siguientes elementos:

- Router Mikrotik
- Servidor Suricata IDS/IPS en Ubuntu
- Servidor KaliLinux (para las pruebas en seguridad de la red)
- Servidor ELK o Elastic Stack (software que muestra los datos de eventos a consultar)

Router Mikrotik:

Este Router brinda múltiples funcionalidades para la administración de la red desde crear reglas de Firewall, Vlans, control de ancho de banda y QoS(Calidad de Servicio) y demás funciones de un router sin ser ajeno a las marcas reconocidas que pueda ofrecer en sus configuraciones, su principal función de Mikrotik será direccionar el tráfico entrante al servidor IDS e IPS Suricata . Se eligió este router Mikrotik por tratarse de un equipo robusto y que se encuentra al alcance de conseguirlo en el mercado informático a un precio módico.

Sistema Operativo Ubuntu

Es un sistema operativo de código abierto, de distribución de Linux, su interfaz gráfica de entorno factible es usado en ordenadores además posee una versión orientada para servidores. Su patrocinador Canonical mantiene a Ubuntu brindando soporte y nuevas versiones que se libera cada seis meses.

Servidor IDS/IPS Suricata:

Es un sistema que tiene la capacidad en la Detección de Intrusos (IDS) y Sistema de Prevención de Intrusos (IPS), desarrollado como un proyecto en código abierto además de ser respaldado por Open Information Security Foundation (OISF), su funcionalidad se basa en un conjunto de reglas y de crear reglas personalizadas también es útil para en el control y monitoreo de posibles amenazas en una red informática , al igual que Snort es otro IDS/IPS que cuenta con las mismas funcionalidades, sin embargo como se muestra en la Tabla 12 se eligió Suricata por su versatilidad en su manejo en su bajo costos de implementarlo además de contar con algunas características de valor agregado que se mencionan a continuación:

Tabla 12

Comparación IDS /IPS Open Source Snort y Suricata

Características	Snort	Suricata
Multi-Threading	No	Si
Soporte IPV6	Si	Si
Detección automática de protocoles	No	Si
Análisis Avanzado de HTTP	No	Si
Aceleración de GPU	No	Si
Variables Globales/G	No	Si
GeoIP	No	Si

Además, el sistema operativo Linux al contar con diferentes distribuciones como se muestra en la tabla 13 tiene el soporte para ser instalado el software de prevención de intrusos Suricata, haciéndolo versátil en sus diferentes plataformas para su desarrollo e implementación.

Tabla 13

Sistema Operativos Soportados

Sistema Operativo	Soporte
Ubuntu	Si
Centos	Si
Fedora	Si
RedHat Enterprise Linux.	Si
Windows	Si
Debian	Si

KALI LINUX

Es una distribución desarrollada por Debian GNU/LINUX cuenta con múltiples herramientas o aplicaciones como se muestra en la Tabla 14, cada aplicación cuenta con su funcionalidad de infringir la seguridad, estas herramientas se ponen a prueba para la seguridad en las redes informáticas donde es usado por muchos aprendices o expertos en seguridad además es usado para las auditorías en la seguridad informática.

Tabla 14

Muestra de algunas herramientas KALI LINUX

Aplicaciones Kali Linux	Funcionalidad
• NMAP	Recopila información de todos los puertos vulnerables para posteriores ataques.
• METASPLOIT Y EXPLOITS	Vulnera puertos privilegiados de equipos locales y remotos
• SQL INJECTOR	Se obtiene información y filtración en distintas bases de datos vulnerables.
• AICRACK	Herramienta inalámbrica que analiza y hackea contraseña de protocolos WEP/WAP/WPA2

Servidor Elastic Stack:

ELK que por sus siglas representan (ElasticSearch, Logstash, Kibana), este conjunto de herramientas basado en código abierto, representa los tres softwares realizar un análisis de registros de manera eficiente proporcionando resultados relevantes y de visualización de información factible. Sin embargo, se necesita una herramienta más para la recopilación de datos llamada Beats. Por tanto, ELK pasó a renombrarse como Elastic Stack y se representa en la Figura 4.

Figura 4

Arquitectura Elastic Stack.

- **Logstash:**

Es el proceso de recopilación de datos, consta de tres componentes entrada, filtro y salida. En entrada se encarga de recibir los Logs en canalizar los datos y eventos, seguidamente son filtrados para realizar una acción y por último por medio de un index tiene como destino o salida llegar a Elasticsearch.

- **ElasticSearch:**

Es un motor de almacenamiento, lo que permite la búsqueda y análisis de grandes volúmenes de datos. ElastasticSearch escucha el puerto 9200 que viene por default en su configuración.

- **Kibana:**

Es la herramienta de entorno web de visualizaciones que a través de la información obtenida por ElasticSearch lo muestra a través de Dashboard, se utilizará para visualizar las alertas de Suricata. Kibana escucha el puerto 5601 que viene por default en su configuración.

- **Filebeat:**

Pertenece a la familia Beats, su función es recopilar una mayor masa de datos sin consumir muchos recursos y enviarlos a Logstash. Filebeat escucha el puerto 5400 que viene por default en su configuración.

Implementación del esquema de una red IDS/IPS Suricata Open Source

Una vez definida la arquitectura para la Implementación de una plataforma Server utilizando Suricata Open Source para la detección y prevención de intrusos en la red, se realizó un despliegue en un laboratorio interno, para llevar a cabo la instalación, de configuración se eligió la plataforma de virtualización para Sistemas Operativos a Oracle VM Virtual Box tanto para Elastic Stack o ELK, Suricata y Kali Linux.

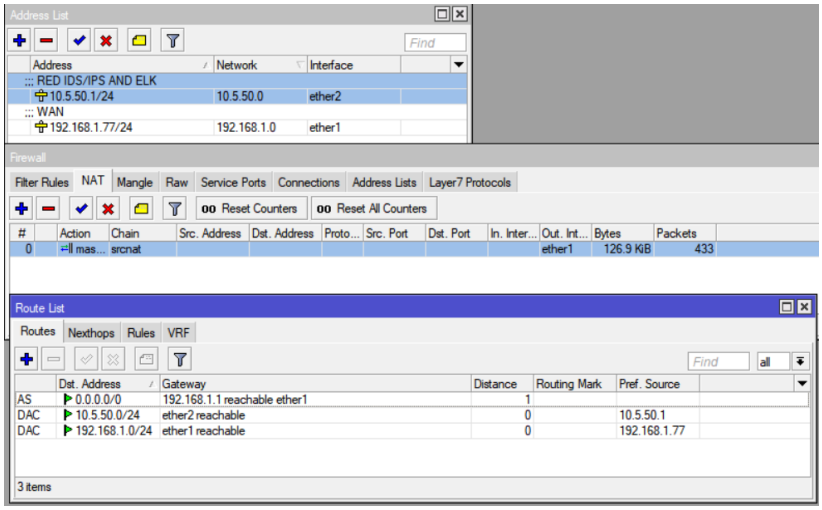
Configuración en Router Mikrotik

Se eligió un hardware Mikrotik modelo Routerboard RB450, mediante el software Winbox (software de interfaz gráfica para acceder a su configuración) se procede a iniciar las configuraciones para el acceso a Internet y asignado el segmento de IP para la creación de una red local, ver Figura 5:

IP > Address

Figura 5

Asignando IP WAN y LAN a Router Mikrotik.



Desde la interfaz ether1, se asignó una dirección IP fija del ISP (Proveedor de Servicio de Internet) y fue nombrado con un comentario WAN, dicha interfaz permitió el ingreso del Internet al Router Mikrotik.

Posteriormente desde la interfaz ether 2 se procedió a crear la red 10.5.50.0/24 que tuvo como objetivo analizar todo el tráfico de dicho segmento de red y que fue destinado tanto para el servidor IDS e IPS Suricata como para el otro servidor Elastic Stack, así mimo la misma red se le asignó a cada host (Laptop o PC) una IP Fija que perteneció cada host a la red administrativa. Siguiendo con la configuración del router Mikrotik se configuró lo siguiente:

IP > Firewall > NAT

Se realizó el enrutamiento de la dirección IP origen ether1 a otra IP determinada por la interfaz ether2.

IP > DNS

Para este ejemplo se agregó los DNS de Google o podría ser los DNS del ISP.

IP > Route List

En esta sección se agregó el Gateway del ISP.

New Terminal: Desde el terminal se comprueba la salida a internet, digitando: ping google.com, ver Figura 6. De esta manera se comprobó exitosamente que el Mikrotik tuvo conexión a internet.

Figura 6:

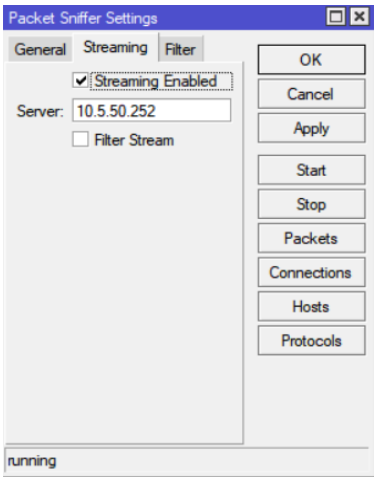
Comprobación en salida a Internet.

```
[admin@MikroTik] > ping google.com
  SEQ HOST                                     SIZE TTL TIME  STATUS
    0 64.233.190.101                             56  41 310ms
    1 64.233.190.101                             56  41 383ms
    2 64.233.190.101                             56  41 192ms
    3 64.233.190.101                             56  41 198ms
    4 64.233.190.101                             56  41 171ms
    5 64.233.190.101                             56  41 195ms
    sent=6 received=6 packet-loss=0% min-rtt=171ms avg-rtt=241ms max-rtt=383ms
```

Tools > Packet Sniffer

Figura 7:

Direccionamiento de paquetes a Suricata



Se eligió la herramienta Packet Sniffer del Mikrotik para realizar copias de todos los paquetes de datos que atraviesan en el router para el direccionamiento del tráfico hacia la IP del servidor IDS Suricata, como se muestra en la Figura 7 con fines de realizar el análisis y comportamiento de la red.

Configuración en S.O UBUNTU

Para la instalación del servidor IDS/IPS Suricata se instaló el sistema operativo Ubuntu 18.4.3. Desde la distribución de Linux Ubuntu se prosiguió a configurar la tarjeta de red del servidor asignándole una dirección IP fija 10.5.50.252/24 como se muestra en la Figura 8.

Figura 8:

Asignando IP a servidor Suricata.

Instalación de Servidor IDS SURICATA

Antes de iniciar con la instalación de Suricata, se comprobó el contar con todas las dependencias, para ello se abrió un terminal como se muestra en la Figura 9, se digitó lo siguiente:

**sudo apt-get –y install libpcre3 libpcre3-dbg libpcre3-dev **

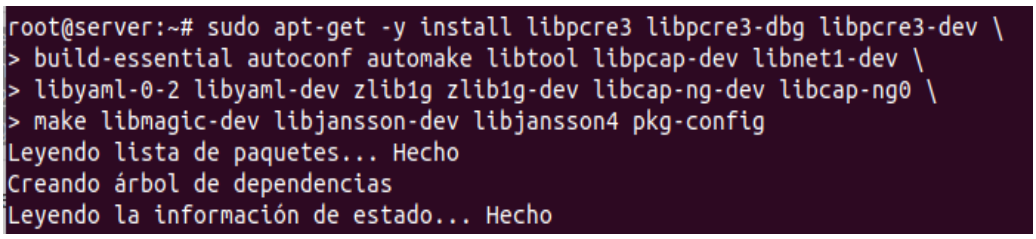
**build-essential autoconf automake libtool libpcap-dev libnet1-dev **

**libyaml libyaml-0-2-dev zlib1g zlib1g-dev libcap-ng-dev-libcap NG0 **

marcar libmagic-dev libjansson-dev libjansson4 pkg-config00

Figura 9

Instalación de dependencias en Suricata.



IPS (Sistema de Prevención de Intrusos)

Suricata funciona como IDS (Sistema de Detección de Intrusos), sin embargo, para contar con la función IPS digitaremos lo siguiente:

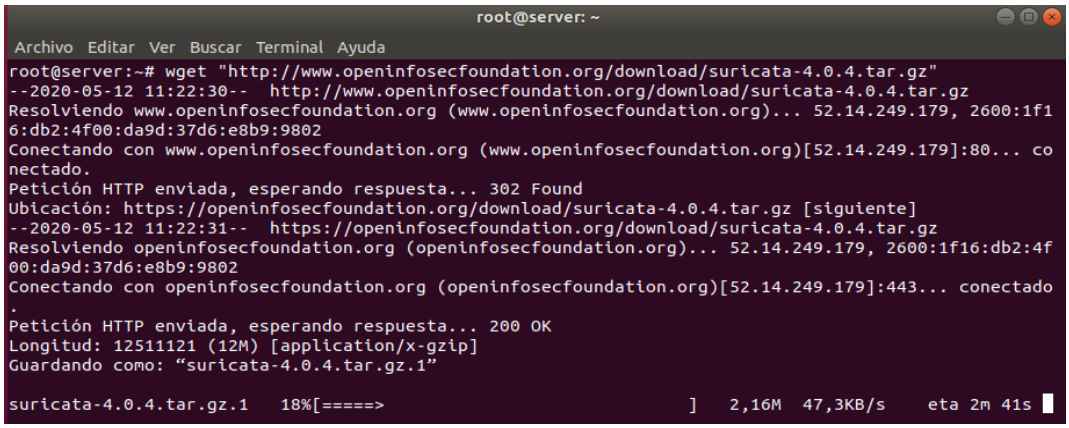
apt-get –y install libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev libnfnetlink0

Continuando con la instalación de Suricata, se procede a descargar el código fuente de Suricata como se muestra en la Figura 10, se digitó el siguiente comando:

wget “http://www.openinfosecfoundation.org/download/suricata-4.0.4.tar.gz”

Figura 10

Descarga del código fuente Suricata



Una vez descargado, se descomprime el archivo e ingresamos al directorio de Suricata.

Tar –xvf suricata-4.0.4.tar.gz
cd Suricata-4.0.4

Tras ingresar al directorio, se procedió a compilar e instalar el motor

./configure –enable-nfqueue –prefix = / usr –sysconfdir = / etc –localstatedir = / var –disable-gccmarch-native

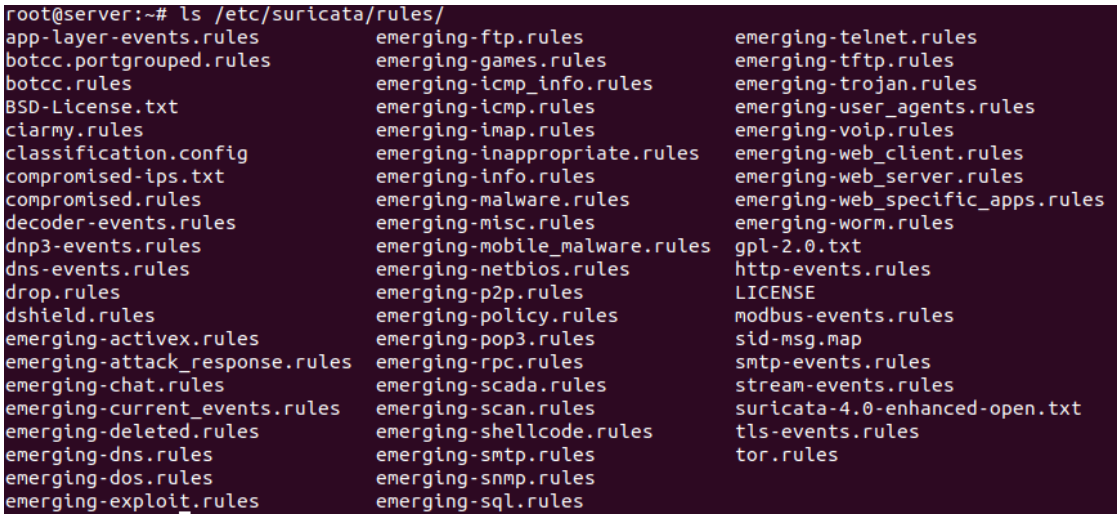
Se procedió con la instalación de Suricata y sus configuraciones predeterminadas, así también se agregó de forma automática un conjunto de reglas completas y disponibles para el sistema IDS.

Make
make install-full

Una vez instalado Suricata, se consultó las reglas que fueron instaladas como se muestra en la Figura 11, se digitó el siguiente comando: **ls /etc/suricata/rules**

Figura 11*:*

Reglas en Suricata.

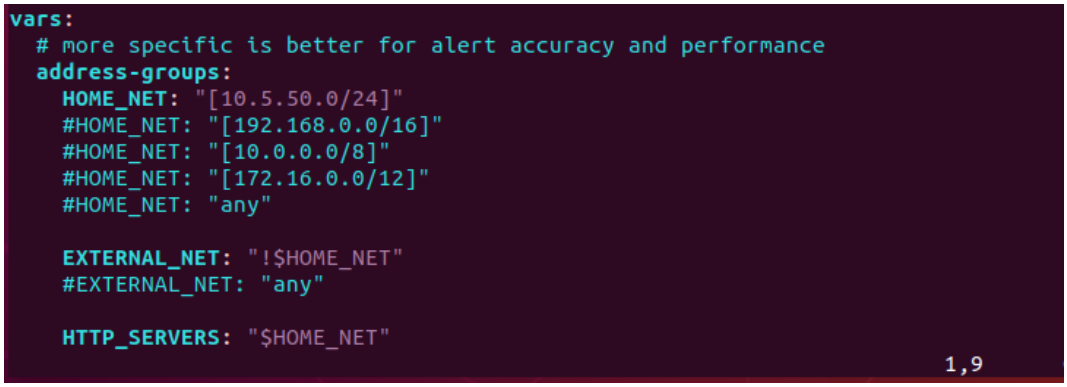


A continuación, se definió la red que se deseó defender desde HOME_NET, se digitó 10.5.50.0/24 sobre la que funcionó el Sistema de Detección Intrusos como se muestra en la Figura 12, se editó el archivo con el siguiente comando:

vim /etc/suricata/suricata.yml

Figura 12*:*

Configuración de red en Suricata.



Así también dentro del mismo archivo se buscó y reemplazó el nombre de la interfaz de red eth0 por el nombre que tiene la tarjeta de red, en este caso se estableció el nombre de la interfaz de red física por el nombre “enp0s3”, finalmente se guardó los cambios.

Configuración de servidor KLINUX

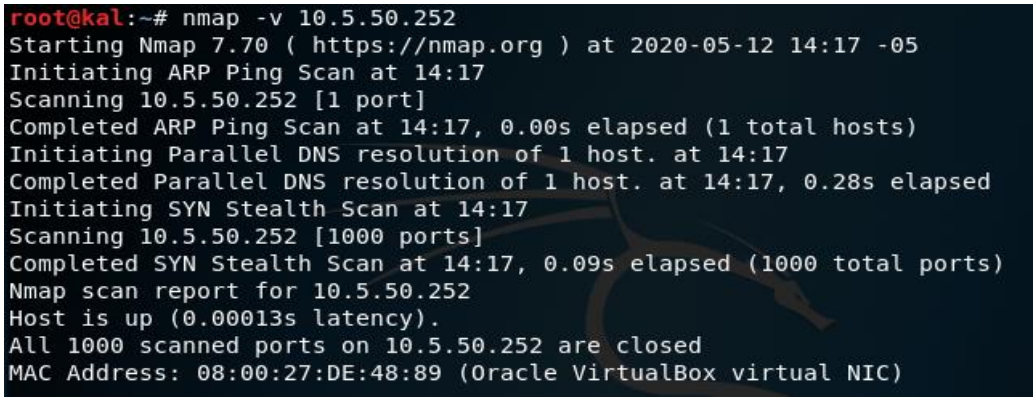
En el servidor Klinux se configuró la tarjeta de red con una dirección IP fija 10.5.50.3/24 que se encontró en el mismo segmento de la red administrativa.

El ataque consistió en enviar un ataque para realizar escaneos de puertos abiertos hacia la víctima, tal como se muestra en la Figura 16, en este ejemplo la víctima fue el host del servidor IDS Suricata, por lo tanto, se digitó lo siguiente:

nmap –v 10.5.50.252

Figura13:

Inspección de puertos en Klinux.



Iniciando servicios de Suricata

Desde un terminal se inició el servicio en vivo de Suricata haciendo un llamado al archivo suricata.yaml seguido del nombre de interfaz de red en nuestro caso “enp0s3”, como se muestra en Figura 13, se digitó lo siguiente:

suricata –c /etc/suricata/suricata.yaml –i enp0s3

Figura14*:*

Servicios habilitados en Suricata.

A su vez se abrió otro terminal y se ingresó al directorio **/var/log/suricata** para comprobar los archivos eve.json y fast.log, ambos archivos se encargaron de almacenar y mostrar información en tiempo real de lo que venía ocurriendo en la red, ver Figura 15.

***Figura 15**:*

Archivos logs en Suricata.

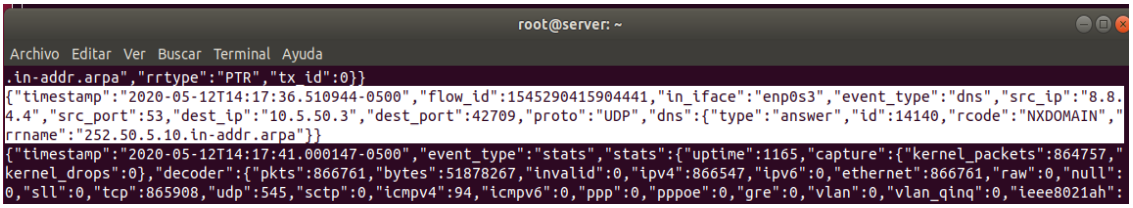
```
root@server:/var/log/suricata# ls /var/log/suricata/
certs  eve.json  fast.log  files  stats.log  suricata.log
```

Se observar en tiempo real lo que ocurría en la red, ver Figura 16. A continuación, se digitó el siguiente comando dentro del mismo directorio.

Tail –f /var/log/suricata/eve.json

Figura 16*:*

Captura de paquetes en Suricata.



Como se puede observar se eligió el archivo eve.json porque muestra los eventos importantes y detallados que son la fecha, hora, tipo de evento, IP origen e IP destino, puertos y protocolos que ocurre en tiempo real.

También se observó y comprobó que el servidor Suricata se encontró listo para la detección de intrusos, sin embargo, tomar lectura del log resultó tedioso por tener que leer muchas líneas desde un terminal. No obstante, se procedió a usar el software Elastic Stack, dicho software permitió contar con interfaz gráfica de control y monitoreo a través de un Dashboard y demás funcionalidades para una mejor administración de los sucesos ocurridos en la red.

Instalación y configuración de Elastic Stack:

Para la instalación de los componentes Elastic Stack se recomendó hacerlo desde otro nuevo servidor con sistema operativo Linux, sin embargo, para el ejemplo se usó el mismo servidor Ubuntu.

- **Elasticsearch**

Antes de proceder con la instalación fue necesario instalar java, y se realizó con el siguiente comando:

apt install openjdk-8-jdk openjdk-8-jre

Seguidamente se realizó la importación de llave PGP en el sistema

wget –q O – https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –

A su vez se agregó los repositorios de Elastic Stack digitando el siguiente comando:

echo “deb https://artifacts.elastic.co/packages/6.x/apt stable main” | sudo tee –a /etc/apt/sources.list.d/elastic-6.x.list

A continuación, se procedió con la instalación de Elasticsearch como se muestra en la Figura 17, se digitó el siguiente comando:

apt-get –y install elasticsearch

Figura 17*:*

Instalación de Elasticsearch.

```
root@server:~# apt-get -y install elasticsearch
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  elasticsearch
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 28 no actualizados.
Se necesita descargar 149 MB de archivos.
Se utilizarán 241 MB de espacio de disco adicional después de esta operación.
Des:1 https://artifacts.elastic.co/packages/6.x/apt stable/main amd64 elasticsearch all 6.8.9 [149 MB]
5% [1 elasticsearch 9.386 kB/149 MB 6%]                                    49,0 kB/s 47min 36s
```

Terminada la instalación, se prosiguió con la configuración del archivo elasticesearch.yml con el siguiente comando: **vim/etc/elasticsearch/elasticsearch.yml**

Así mismo se ubicó las siguientes líneas y se procedió en realizar la edición y se configuró lo siguiente:

…….Network……

network.host: localhost

……Paths……

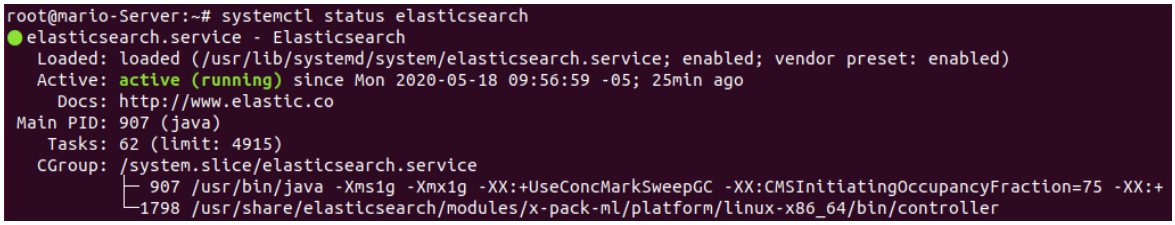
path.data: /var/lib/elasticsearch
patch.logs: /var/log/elasticsearch

Después de guardar los cambios, se levantó los servicios, además para que inicie durante el arranque del sistema, así también se comprobó su estado modo activo como se muestra en Figura 18, se digitó lo siguiente:

systemctl start elasticsearch
systemctl daemon-reload
systemctl enable elasticseach
systemctl status elasticsearch

Figura 18*:*

Estado activo de elasticsearch.



- **Kibana**

A continuación, se procedió en actualizar las dependencias e instalar Kibana como se muestra en la Figura 19, se digitó lo siguiente:

apt-get update

apt-get –y install kibana

Figura 19*:*

Instalación de Kibana.

```
root@server:~# apt-get -y install kibana
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  kibana
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 28 no actualizados.
Se necesita descargar 194 MB de archivos.
Se utilizarán 445 MB de espacio de disco adicional después de esta operación.
Des:1 https://artifacts.elastic.co/packages/6.x/apt stable/main amd64 kibana amd64 6.8.9 [194 MB]
14% [1 kibana 33,5 MB/194 MB 17%]                                    142 kB/s 18min 51s
```

Finalizada la instalación, se prosiguió con la configuración del archivo 53ilebe.yml con el siguiente comando: **vim/etc/53ilebe/53ilebe.yml**

Así mismo se ubicó las siguientes líneas y se procedió en realizar la edición y configuración de:

server.host: “localhost”
elasticsearch.hosts: [“http://localhost:9200”]

Después de guardar los cambios se levantó los servicios para que se inicien junto al arranque del sistema, así también se comprobó el estado en modo activo como se muestra en Figura 20, digitando lo siguiente:

systemctl start kibana.service
systemctl enable kibana
systemctl status kibana

Figura 20*:*

Estado activo de Kibana.

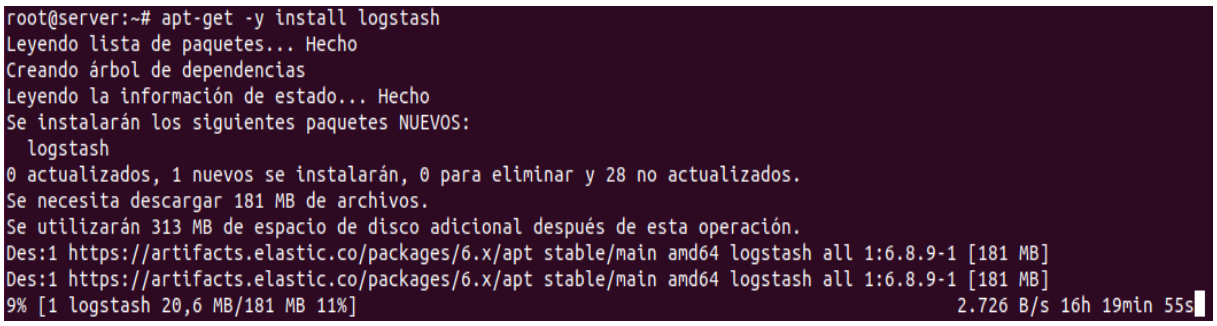
- **Logstash**

A continuación, se procedió en actualizar las dependencias, acto seguido se instaló Logstash como se muestra Figura 21, se digitó lo siguiente:

```
apt-get update
apt-get install 54ilebeat
```

Figura 21*:*

Instalación de Logstash.



A continuación se inició la configuración de Logstash, para el funcionamiento se necesitó conocer la entrada y salida de datos que procesa, por tanto se creó 3 nuevos archivos dentro del directorio /etc/54ilebeat/conf.d/ , se creó el primer archivo cuyo nombre se le asignó 02beats-input.conf , se digitó lo siguiente:

vim /etc/55ilebeat/conf.d/02-beats-input.conf

Dentro del archivo se digitó las siguientes líneas y se guardó los cambios:

```
input {
        beats {
         port => 5044
}
}
```

Se habilitó el puerto de Logstash, digitando lo siguiente:

ufw allow 5044

A continuación, se crea el segundo nuevo archivo con el nombre 10-syslog-filter.conf

vim /etc/55ilebeat/conf.d/10-syslog-filter.conf

Dentro del fichero creado se digitó las siguientes líneas y se guardó los cambios

```
filter {
      grok {
            match => { "message" => "%{MONTHNUM:_month}/%{MONTHDAY:_day}-
%{TIME:_time} \[\*{2}\] \[%
{NUMBER:gid}:%{NUMBER:signature_id}:%{NUMBER:rev}\] %{DATA.msg} \[\*{2}\]
\[Classification: %
{DATA:classification}\] \[Priority: %{NUMBER:priority}\] \{%{DATA.proto}\} % {IP:src_ip}:*%
{NUMBER:src_port}* -> %{IP:dest_ip}:*%{NUMBER:des_port}*" }
          add_field => [ "received_at", "%{@timestamp}" ]
          add_field => [ "received_from", "%{host}" ]
          }
}
```

A su vez se creó el tercer archivo con el nombre 30-elasticsearch-output.conf dentro del mismo directorio. La salida fue a través de un índice en función a un evento de tiempo que pasó por la canalización de Logstash y que fue enviado a ElasticSearch.

Vim /etc/55ilebeat/conf.d/30-elasticsearch-output.conf

Dentro del archivo creado se digitó las siguientes líneas y se guardó los cambios

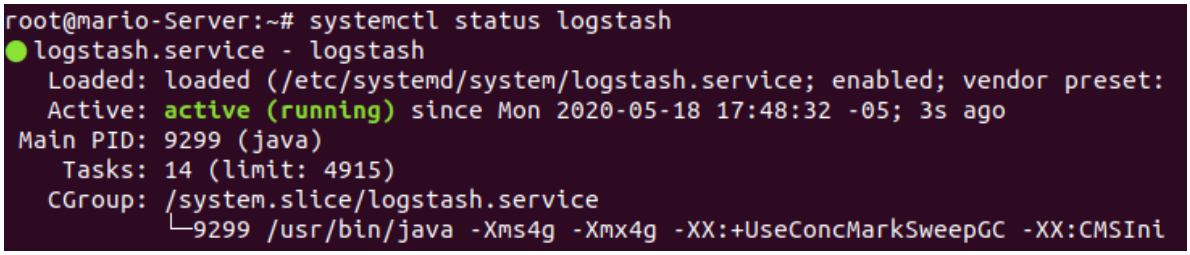
```
output {
 elasticsearch {
 hosts => [“localhost:9200”]
 manage_template => false
 index => “%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}”
         }
}
```

Después de guardar los cambios, se levantó los servicios para que inicie durante el arranque del sistema, así también se comprobó su estado en modo activo Figura 22 y se digitó:

```
systemctl daemon-reload
systemctl start 56ilebeat.service
systemctl enable 56ilebeat.service
systemctl status 56ilebeat
```

Figura 22*:*

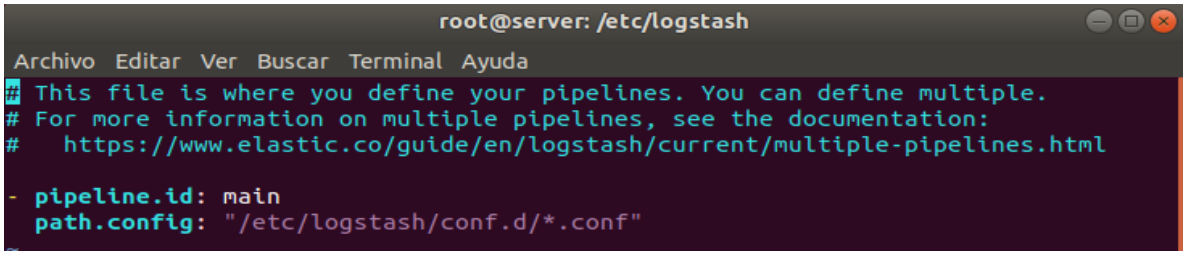
Estado activo de Logstash.



Además, se editó el archivo pipelines.yml, este se encontraba en el directorio /etc/56ilebeat, se procedió en editar las siguientes líneas indicando la ruta de los archivos que fueron creado, ver Figura 23.

Figura 23*:*

Configuración Logstash.



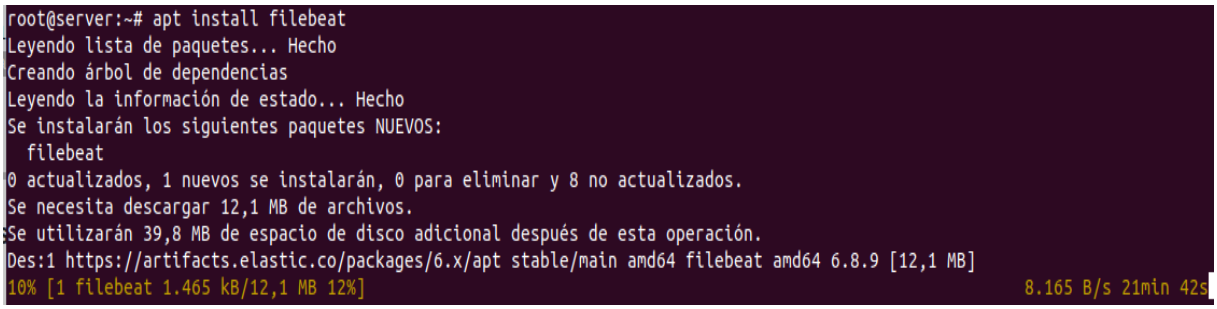
- **Filebeat**

Después de haber instalado todo Elastic Stack, se procedió con la instalación de Filebeat como se muestra en Figura 24, se digitó lo siguiente:

apt-install 57ilebeat

Figura 24*:*

Instalación de Filebeat.



Finalizada la instalación, se prosiguió en configurar el archivo 57ilebeat.yml con el siguiente comando: **vim/etc/57ilebeat/57ilebeat.yml.**

Así mismo se indicó a Filebeat que reenvié la información que generó Suricata a través de su archivo de log eve.json y se editó la sección output de Logstash, como se muestra a continuación:

…….Filebeat inputs…..

enabled: true
paths:
/var/log/suricata/eve.json

……Elasticsearch output……

#output.elasticsearch:
#hosts: [“localhost: 9200”]

……Logstash output……

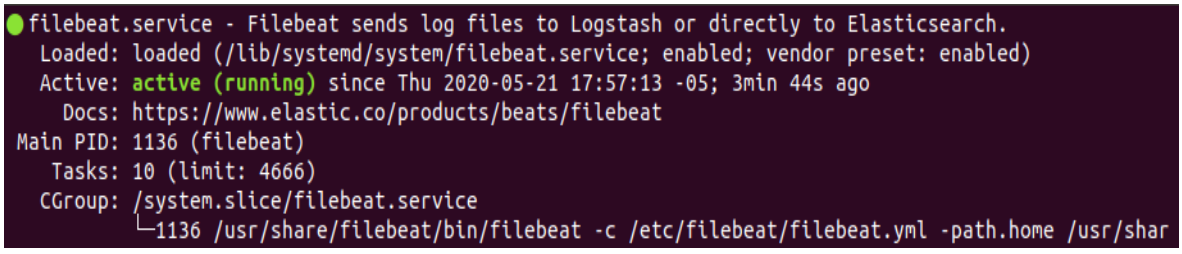
output.logstash:
hosts:[“localhost:5044”]
……Logstash output……

Después de guardar los cambios se levantó los servicios para que se inicien junto al arranque del sistema, así también se comprobó el estado en modo activo ver Figura 25. Se digitó lo siguiente:

systemctl daemon-reload
systemctl start filebeat.service
systemctl enable filebeat.service
systemctl status filebeat

Figura 25:

Estado activo de Filebeat.



Para que se ejecute filebeat especificando el archivo de configuración pipeline se ingresó al directorio /usr/share/filebeat, acto seguido se digitó el siguiente comando:

bin/filebeat –f /etc/filebeat/.conf.d/ --patch.settings /etc/filebeat

Una vez validados los servicios que se encentraron en ejecución, se prosiguió en la validación de los puertos, como se recuerda Elasticsearch utiliza el puerto 9200 y 9300,

Kibana utiliza el puerto 5601, y Filebeat utiliza el puerto 5044. Para comprobarlo ver Figura 26, se digitó el siguiente comando:

netstat –tupnl

Figura 26*:*

Puertos iniciados de Elastic Stack y Filebeat.

```
root@mario-Server:/etc/logstash# netstat -tupnl
Conexiones activas de Internet (solo servidores)
Proto  Recib Enviad Dirección local         Dirección remota       Estado       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               ESCUCHAR    625/systemd-resolve
tcp        0      0 127.0.0.1:631           0.0.0.0:*               ESCUCHAR    5130/cupsd
tcp        0      0 127.0.0.1:5601          0.0.0.0:*               ESCUCHAR    704/node
tcp        0      0 0.0.0.0:80              0.0.0.0:*               ESCUCHAR    930/nginx: master p
tcp6       0      0 ::1:631                 :::*                    ESCUCHAR    5130/cupsd
tcp6       0      0 127.0.0.1:9600          :::*                    ESCUCHAR    5049/java
tcp6       0      0 127.0.0.1:9200          :::*                    ESCUCHAR    903/java
tcp6       0      0 :::5044                 :::*                    ESCUCHAR    5049/java
tcp6       0      0 127.0.0.1:9300          :::*                    ESCUCHAR    903/java
udp        0      0 0.0.0.0:40815           0.0.0.0:*                           686/avahi-daemon: r
udp        0      0 127.0.0.53:53           0.0.0.0:*                           625/systemd-resolve
udp        0      0 0.0.0.0:68              0.0.0.0:*                           876/dhclient
udp        0      0 0.0.0.0:631             0.0.0.0:*                           5131/cups-browsed
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           686/avahi-daemon: r
udp6       0      0 :::32788                :::*                                686/avahi-daemon: r
udp6       0      0 :::5353                 :::*                                686/avahi-daemon: r
```

A continuación, desde un navegador web se hizo el llamado de Kibana escribiendo la siguiente URL: **https://localhost:5601** como se muestra en Figura 27, seguidamente se configuró el index de filebeat y se seleccionó @timestamp.

Figura 27:

Configuración Kibana.

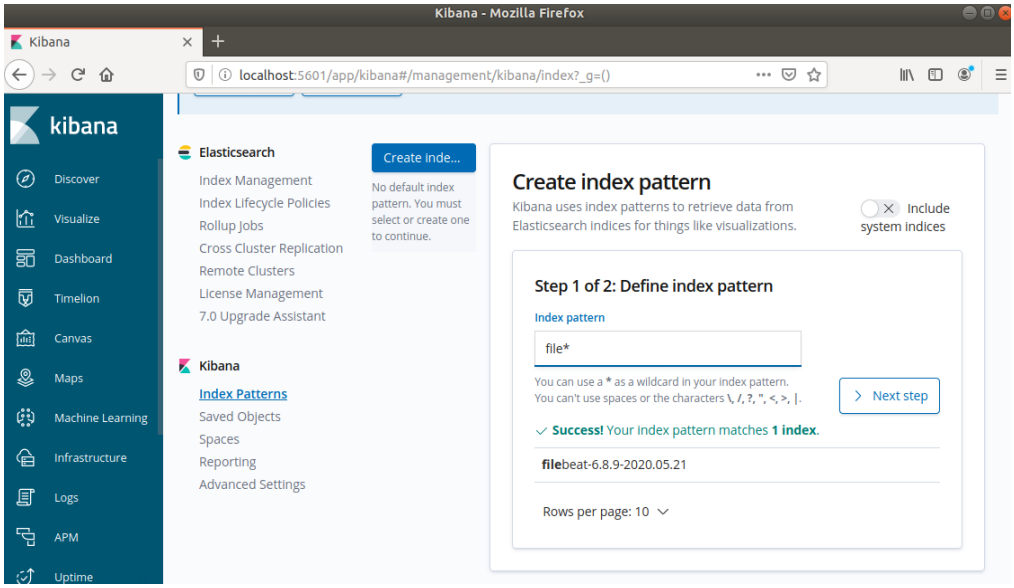
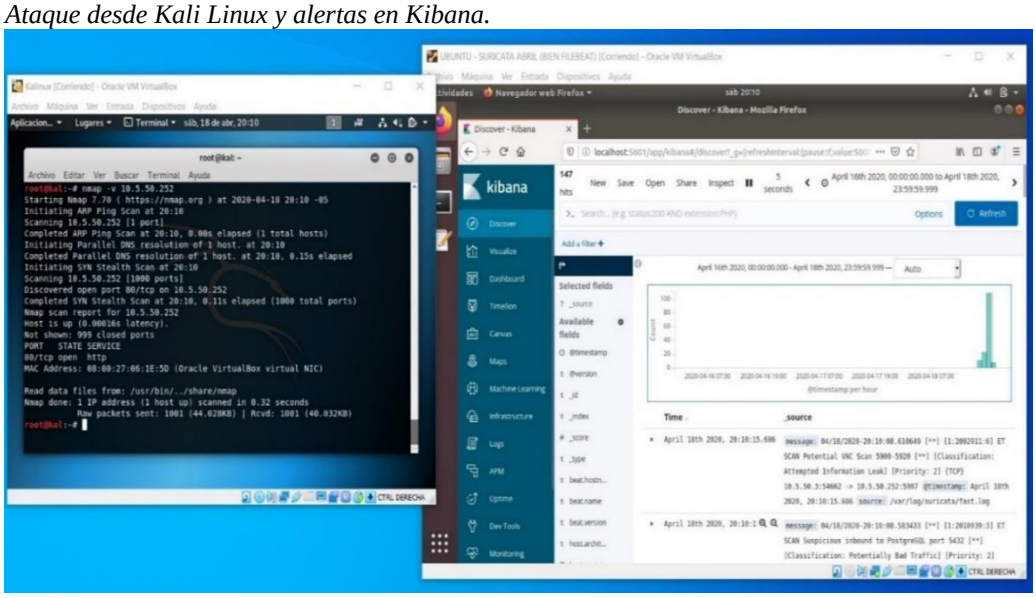


Figura 28:

Ataque desde Kali Linux y alertas en Kibana.

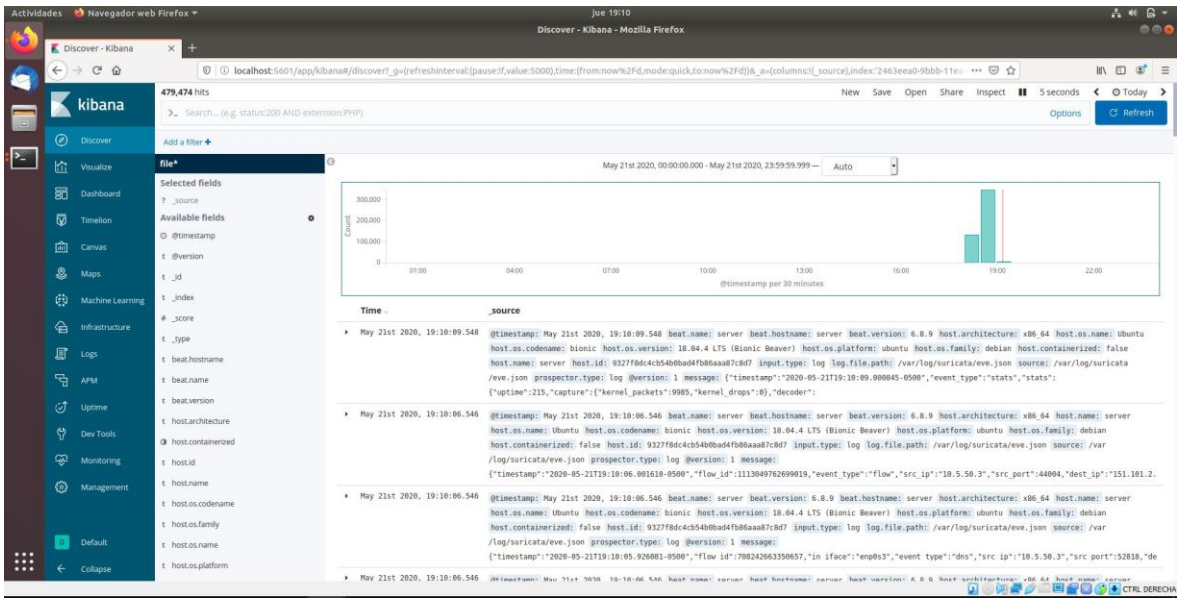


Desde la sección Discover de Kibana se mostró los datos de los logs, así también se pudo apreciar un ataque de escaneo de puertos y la procedencia del IP del atacante realizado desde Kali Linux como se muestra en Figura 28.

El resultado fue determinante porque cumplió con el objetivo después de la implementación además se realizó el análisis y protección en tiempo real, ver Figura 29.

Figura 29*:*

Resultados de ataque en Kibana.



Análisis: Los resultados del objetivo específico 2 en diseñar e implementar una plataforma Server utilizando la arquitectura Suricata Open Source fue desarrollado con un diseño de la arquitectura, simulando un escenario de red para un pyme o mediana empresa (figura 2) se llevó a cabo con la elección del software y sus configuraciones como se muestra desde la figura 5 hasta la figura 29.

Para el **objetivo específico 3** que contiene "Determinar la conformidad de la implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS".

Para determinar la conformidad de la implementación se realizó por medio de la estadística descriptiva mostrando su promedio por cada dimensión de la variable dependiente IDS/IPS.

Resultados de Media Post-Test Variable IDS/IPS:

Tabla 15

Resultado Post-Test de Media en Dimensión Fiabilidad

	N	Media	Desv. estándar
1. Verifica usted que el software Open Source IDS sea eficiente en analizar en el tráfico entrante	12	4.58	.515
2. Verifica usted que el software Open Source IDS sea efectivo como lo es un software IPS con licencia	12	4.42	.515
3. Verifica usted que implementar software Open Source IDS demande de mucha inversión	12	2.25	.452
N válido (por lista) y promedio total	12	4	

Fuente: SPSS V.26

Lo más relevante de la Tabla 15 es el promedio total 4 de la dimensión fiabilidad, esto determina la conformidad en un nivel de acuerdo, por tanto, se verifica que el software IDS/IPS Open Source resultó ser confiable y fiable como un software con licencia.

Tabla 16

Resultado Post-Test de Media en Dimensión Seguridad

	N	Media	Desv. estándar
4. Verifica usted la importancia de la seguridad en su centro de datos	12	4.92	.289
5. Verifica usted que el software Open Source IDS ayudaría a sus problemas de contrarrestar ataques cibernéticos	12	4.50	.522
6. Verifica usted que un antivirus sea basto en salvaguardar la información de una pyme o mediana empresa	12	1.67	.492
N válido (por lista) y promedio total	12	4	

Fuente: SPSS V.26

Lo más relevante de la Tabla 16 es el promedio total 4 de la dimensión seguridad, esto determina la conformidad en un nivel de acuerdo, por tanto, se verifica que el software IDS/IPS Open Source resultó ser seguro para mitigar los ataques cibernéticos.

Tabla 17

Resultado Post-Test de Media en Dimensión Flexibilidad

	N	Media	Desv. estándar
7. Verifica usted que el software Open Source IDS cuente con soporte de ayuda	12	3.83	.718
8. Verifica usted que el software Open Source IDS cuente con actualizaciones constantes en su base de datos	12	4.42	.515
9. Verifica usted que sería necesario implementar un software IDS en la empresa donde trabaja	12	4.67	.492
N válido (por lista)	12	4	

Fuente: SPSS V.26

Lo más relevante de la Tabla 17 es el promedio total 4 de la dimensión flexibilidad, esto determina la conformidad en un nivel de acuerdo, por tanto, se verifica que el software IDS/IPS Open Source resultó mantener y recibir actualizaciones siendo flexible para la implementación.

Tabla 18

Resultado Post-Test de Media en Dimensión Inspección

	N	Media	Desv. estándar
10. Verifica usted que el software Open Source IPS registra los eventos anómalos en el tráfico de paquetes entrantes	12	4.83	.389
11. Verifica usted que el software Open Source IPS realice identificación de vulnerabilidades	12	5.00	.000
N válido (por lista) y promedio total	12	5	

Fuente: SPSS V.26

Lo más relevante de la Tabla 18 es el promedio total 5 de la dimensión inspección, esto determina la conformidad en un nivel totalmente de acuerdo, por tanto, se verificó que el software IDS/IPS Open Source resultó registrar e identificar eventos anómalos en el tráfico de red.

Tabla 19

Resultado Post-Test de Media en Dimensión Funcionalidad

	N	Media	Desv. estándar
12. Verifica usted que el software Open Source IPS monitorea en tiempo real el trafico	12	4.92	.289
13. Verifica usted que el software Open Source IPS realice acciones para contrarrestar los ataques	12	4.83	.389
N válido (por lista) y promedio total	12	5	

Fuente: SPSS V.26

Lo más relevante de la Tabla 19 es el promedio total 5 de la dimensión funcionalidad, esto determina la conformidad en un nivel totalmente de acuerdo, por tanto, se verifica que el software IDS/IPS Open Source resultó tener la funcionalidad de monitoreo real además de contrarrestar los ataques con acciones.

Tabla 20

Resultado Post-Test de Media en Dimensión Observación

	N	Media	Desv. estándar
14. Verifica usted que el software Open Source IPS muestra los resultados en tiempo real	12	5.00	.000
15. Verifica usted que el software Open Source IPS muestre resultados con gráficos mediante la integración con otros softwares.	12	5.00	.000
N válido (por lista)	12	5	

Fuente: SPSS V.26

Lo más relevante de la Tabla 20 es el promedio total 5 de la dimensión observación, esto determina la conformidad en un nivel totalmente de acuerdo, por tanto, se verificó que el software IDS/IPS Open Source resultó ser observacional y visual a través del Dashboard mostrando los eventos ocurridos en el tráfico de la red

Tabla 21

Resultado Post Test de Sistema de Detección y Prevención de Intrusos

Sistema de Detección y Prevención de Intrusos				
		Total	**Porcentaje**	**Porcentaje Valido**
Valido	**Totalmente desacuerdo**	4	2%	2%
	En desacuerdo	17	9%	9%
	Ni de acuerdo ni en desacuerdo	7	4%	4%
	De acuerdo	41	23%	23%
	Totalmente de acuerdo	111	62%	62%
	Total	180	100%	100%

De la totalidad de los encuestados en el área TI en pymes y medias empresas se puede apreciar en la Tabla 21 que después de la implementación de una plataforma Server utilizando la arquitectura Suricata Open Source al 100% de un total de 180 respuestas de los evaluados un 62% verifica estar totalmente de acuerdo mientras que un 23% verificó estar

de acuerdo, esto quiere decir que existe el interés en Implementar una plataforma Server utilizando la arquitectura Suricata Open Source en su centro de datos

Análisis: Los resultados del objetivo específico 3, determinaron la conformidad de la implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS, los encuestados determinaron un nivel de conformidad de 50% estar de acuerdo y el otro 50% determinaron un nivel de conformidad totalmente de acuerdo en mencionada implementación.

3.2 Contraste de hipótesis especificas

a) **La primera hipótesis específica planteada estable que "**El software actual para IDS/IPS en la red es conforme."

Por medio del estadístico descriptivo, donde N es la muestra y el promedio IDS tuvo un nivel de 4 significando estar de acuerdo y el promedio IPS tuvo un nivel de 5 significando estar totalmente de acuerdo, por lo tanto se afirma que el software actual Suricata Open Source IDS e IPS es conforme a sus funciones y resultados eficientes. (Tabla 22).

Tabla 22

Promedio IDS e IPS

Estadísticos descriptivos

	N	Media	Desv. estándar	Mínimo	Máximo
Promedio IDS	12	4	.389	3	5
Promedio IPS	12	5	.296	4	5

Fuente: SPSS V.26

b) **La segunda hipótesis específica planteada estable que** "Se logra implementar la plataforma Server utilizando la arquitectura Suricata Open Source"

En el despliegue del laboratorio se inició con el diseño de la arquitectura Suricata Open Source, la implementación de la plataforma Server utilizando la arquitectura Suricata se logró simulando un caso real desplegando un laboratorio, el resultado fue lo esperado cuando se realizó las configuraciones con el software Suricata haciendo que se encontrará listo para la detección y prevención de intrusos.

c) **La tercera hipótesis específica planteada establece que** "La implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS es conforme"

Después de haber realizado la implementación en un laboratorio con el software IDS e IPS utilizando la arquitectura Suricata Open Source y simulando ataques desde el servidor Kali Linux, los eventos logs de Suricata donde se muestra la captura de paquetes del presunto ataque, asi mismo para un mejor entorno visual a traves del software ELK muestra los mismos resultados de los eventos logs. Esto mejora significativamente la seguridad en el centro de datos de la información para prevenir o contrarrestar diferentes ataques que provenga desde la red,estos resultados determinaron que la implementación fuera conforme.

Contraste de hipótesis general

Estadística Inferencial

La hipótesis general planteada establece que "La implementación de una plataforma Server utilizando la arquitectura Suricata Open Source cumple significativamente con la detección y prevención de intrusos **"**

En la siguiente figura 30 de las doce personas encuestadas se consiguió el Promedio IDS e IPS en un antes de la implementación y un después de la implementación.

Figura 30:

Resultado del antes y después del Promedio IDS e IPS.

Encuestados	Promedio_IDS_IPS_Antes	Promedio_IDS_IPS_Despues
1	4	4
2	4	4
3	5	5
4	4	4
5	4	4
6	4	4
7	4	4
8	4	4
9	5	5
10	5	4
11	4	4
12	4	4

Fuente: SPSS V.26

Para contrastar la hipótesis, se utilizó el estadístico **Wilcoxon** como se muestra en tabla 23, mediante el reporte del SPPS con los datos conseguidos de los Promedio IDS e IPS de un antes de la implementación y un después de la implementación. (Tabla 24).

Tabla 23

Prueba de rangos con signo de Wilcoxon

		N	Rango promedio	Suma de rangos
Después de la Implementación – Antes de la Implementación	Rangos negativos	1[a]	1.00	1.00
	Rangos positivos	0[b]	.00	.00
	Empates	11[c]		
	Total	12		

a. Después de la Implementación < Antes de la Implementación

b. Después de la Implementación > Antes de la Implementación

c. Después de la Implementación = Antes de la Implementación

Fuente: SPSS V.26

Tabla 24

Estadística de prueba

	Después de la Implementación - Antes de la Implementación
Z	-1.000[b]
Sig. asin. (bilateral)	.317

a. Prueba de rangos con signo de Wilcoxon

b. Se basa en rangos positivos.

Fuente: SPSS V.26

En la tabla 23, se utilizó el estadístico Wilcoxom, se analizaron un total de 12 personas encuestadas. En el resultado se obtuvo cero positivos, 1 negativo y 11 empates. En la estadifica de prueba se observó la Sig. Asin su valor es de 0.317. Por lo tanto, se aceptó la hipótesis nula sin rechazar la hipótesis alterna, es decir los resultados indican que entre el antes y después de la implementación se tiene criterios de decisión para la implementación.

CAPÍTULO IV: DISCUSIÓN Y CONCLUSIONES

4.1 Discusión

La implementación de una plataforma Server utilizando la arquitectura Suricata Open Source para la detección y prevención de intrusos en la red, presentó una conformidad significativa con la prueba no paramétrica de Wilcoxon un valor p de 0.317, a partir de demostrar un alto porcentaje en el promedio con un nivel de conformidad de 63% estar de acuerdo y de 33% en un nivel de conformidad de totalmente de acuerdo en el pre test y luego de la implementación y simulación de un escenario real de Suricata, en el post test se alcanzaron: conformidad de 50% estar de acuerdo y el otro 50% un nivel de conformidad totalmente de acuerdo.

Estos resultados guardan relación con Miño (2020), Diseño de Implementación de un sistema de seguridad de detección de intrusos (IDS) para la plataforma OpenStack utilizando código abierto, se obtuvo un resultado estadístico en la eficiencia de detección de ataques con un promedio de 62.8% en un escenario IDS que consiste directamente al nodo para el monitoreo del tráfico de la red.

En la fase de diseñar e implementar una plataforma Server utilizando la arquitectura Suricata Open Source.” Estuvo basado en un diseño pre experimental, la implementación fue realizada en un laboratorio interno simulando un caso real, se inició diseñando el esquema para el desarrollo IDS/IPS Suricata Open Source, también se definió la arquitectura en el uso del hardware y software posteriormente se procedió en la instalación y configuración se logró exitosamente encontrándose listo para detectar y prevenir los ataques sospechosos.

Estos resultados guardan relación con Quintero (2018) respecto en la implementación de un sistema de intrusos en la red interna de la alcaldía Montería usando software libre", el investigador nos dice que un antivirus no basta para proteger de ataques en efecto implementaron un software Open Source de detección y prevención de intrusos como resultado evidenciaron las alertas y notificaciones ante sucesos no autorizados, además precisaron la monitorización sea administrada.

En la fase determinar la conformidad de la implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS, después de la implementación se realizaron pruebas simulando ataques desde otro server KLinux como escaneo de puertos en efecto el server Suricata Open Source detecto en tiempo real la procedencia del atacante en efecto aplicó la acción de notificar al administrador de redes para tomar las acciones correspondientes.

Estos resultados guardan relación con Perdigón (2022) respecto a Suricata como detector de intrusos para la seguridad por su importancia de analizar los paquetes de datos además de su gran utilidad de ser un software libre IDS/IPS fortaleciendo la seguridad en las redes para empresas. En cuanto a sus resultados se comprobó su efectividad detectando ataques como DoS, fuerza bruta u otros.

Las implicancias de implementar un server utilizando la arquitectura Suricata software Open Source es que básicamente se contribuye como una alternativa para pequeñas o medianas empresas por la reducción de costos y de inversión, además de ser una solución accesible y efectiva fortaleciendo la seguridad de la información y ayudando a tener un mejor control administrable en las redes de datos; también constituye una alternativa de ayuda al

administrador en tomar acción de algún evento sospechoso o anómalo salvaguardando la seguridad.

Con respecto a las limitaciones de la presente investigación, se tuvo dificultad para encontrar al personal TI en las empresas para su colaboración del llenado del cuestionario, además se evidenció que algunas pymes no contaban con un área TI y otras solicitaban servicios externos de algún personal de sistemas para el desarrollo de soluciones, limitando la estandarización de la evaluación, por lo que se determinó considerar la conformidad de los expertos.

4.2 Conclusiones

Se logró implementar una plataforma Server utilizando la arquitectura Suricata Open Source para la detección y prevención de intrusos en la red, los resultados demostraron con la prueba no paramétrica de Wilcoxon un valor p de 0.317 donde la implementación presenta la conformidad para el IDS/IPS. Esta propuesta constituye una alternativa de seguridad para pequeñas y medianas empresas ya que al ser un software libre, minimiza los altos costos de inversión en desarrollo e implementación.

En cuanto al primer objetivo en determinar la conformidad del software actual para IDS/IPS en la red, los resultados pre test demostraron un alto porcentaje en el promedio con un nivel de conformidad de 63% estar de acuerdo y de 33% en un nivel de conformidad de totalmente de acuerdo.

En cuanto al segundo objetivo específico se logró diseñar e implementar una plataforma Server utilizando la arquitectura Suricata Open Source simulando un escenario real, se definió la interconexión de comunicación con el motor de base de datos actualizado de Suricata y el software ELK para el monitoreo.

En cuanto al tercer objetivo específico determinar la conformidad de la implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS, los resultados post test demostraron el promedio con un nivel de conformidad de 50% estar de acuerdo y el otro 50% un nivel de conformidad totalmente de acuerdo estos resultados afirman la conformidad en la implementación de la herramienta del software IDS/IPS Suricata.

Referencias

Agra Monte, A. (2017). *Diseño e implementación de infraestructura NIDS (Network Intrussion Detection System) para PIMES*. (Tesis de bachillerato. Universidad Politécnica de Valencia, Valencia, España). Recuperado de http://hdl.handle.net/10251/88856

Acurio, B. y Rimachi, J. (2019). *Dispositivo electrónico de alarma en tiempo real en la detección de intrusos en viviendas de Iquitos 2019.* (Tesis de título. Universidad Nacional de la Amazonia Peruana, Perú). Recuperado de https://repositorio.unapiquitos.edu.pe/handle/20.500.12737/6803

Astudillo, J., Ortiz, F. y Jiménez,A. *Adaptación del IDS/IPS Suricata para que se pueda convertir en una solución empresarial*. Recuperado de https://www.dspace.espol.edu.ec/bitstream/123456789/20914/1/Paper-Suricata%20%281%29.pdf

Badillo Bernal, D. (2015). *Estudio comparativo de las distribuciones Linux orientado a la seguridad de redes de comunicación*. (Tesis de maestría. Pontificia Universidad Católica del Ecuador, Quito, Ecuador). Recuperado de http://repositorio.puce.edu.ec/handle/22000/10002?show=full

Castillo Mendoza, J. (2022). *Análisis de los sistemas de detección de intrusos (IDS) Open Source y software propietario.* Recuperado de http://dspace.utb.edu.ec/handle/49000/12548

Cabrera Vásquez, F. (2022). *Diseño de una red de seguridad perimetral basada en Open Source para aplicación de IDS para el control de amenazas informáticas en la Universidad Técnica de Babahoyo*. Recuperado de http://dspace.utb.edu.ec/handle/49000/13035

Carrasco Díaz, S. (2019). *Metodología de la investigación científica. San Marcos*

Chapman, C. (2016). *Network Performance and Security: Testing and Analyzing Using Open Source and Low-CostTools*. Cambridge, USA: Syngress. Recuperado de https://books.google.com.pe/books?id=XFrBBwAAQBAJ

Chebyshev, V., Sinitsyn, F., Parinov, D., Larin, B., Kupreev, O., Lopatin, E. (2019). *Desarrollo de las amenazas informáticas en el primer trimestre de 2019*, Recuperado de https://securelist.lat/it-threat-evolution-q1-2019-statistics/88983/

Cifuentes, G. (2018). *Nuevo hackeo de tarjetas de crédito: más de 50 mil clientes de bancos latinoamericanos*. Recuperado de https://www.biobiochile.cl/noticias/nacional/chile/2018/07/28/nuevo-hackeo-de-tarjetas-de-credito-mas-de-50-mil-clientes-de-banco-latinoamericanos.shtml

Coyla Jarita, Y. (2019). *Implementación de un sistema de detección y prevención de intrusos (IDS/IPS), basado en la norma ISO 27001, para el monitoreo perimetral de la seguridad informática, en la red de la Universidad Peruana Unión – Filial Juliaca.* (Tesis de título. Universidad Peruana Unión, Perú). Recuperado de https://repositorio.upeu.edu.pe/handle/20.500.12840/2002

Da Costa Calado, J. P. (2018*). Open Source IDS/IPS in a production environment: comparing, assessing and implementing*. (Tesis de maestría. Universidad de Lisboa, Lisboa, Portugal). Recuperado de http://repositorio.ul.pt/bitstream/10451/35418/1/ulfc121871_tm_Jo%C3%A3o_Paulo_Calado.pdf

Dios, S. y Ortiz, D. (2018). *Diseño lógico y simulación de una red espejo virtual (HONEYNET) para la detección de intrusos informáticos en zona perimetral*. (Tesis de bachillerato. Universidad Nacional de Trujillo, Trujillo-La Libertad, Perú). Recuperado de http://dspace.unitru.edu.pe/handle/UNITRU/11016

Diario Gestión. (2017). *Empresas peruanas destinarán hasta 7% de su presupuesto de TI a ciberseguridad en próximos años*. Recuperado de https://gestion.pe/tecnologia/empresas-peruanas-destinaran-7-presupuesto-ciberseguridad-proximos-anos-153235-noticia/

De Haro Bermejo, F. (2015). *Detección de intrusiones con Snort.* (Tesis de posgrado. Universitat Oberta de Catalunya, Barcelona, España). Recuperado de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/43100/5/fdeharobTFM0615memoria.pdf

Dobladez, M. (2019). *Analizando nuestro tráfico. Implementación de IPS/IDS*. Recuperado de https://mum.mikrotik.com/presentations/PE19/presentation_6661_1550167626.pdf

Eldow, O., Chauhan, P., Lalwani, P. y Potdar, M.B. (2016). *Computer Network Security Ids Tools and Techniques (Snort/Suricata). International Journal of Scientific and Research Publications, 6(1), 593-597.* Recuperado de http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.735.1457

Hernández Ortíz, D. y Santana Barrionuevo, A. (2018*). Diseño e implementación del sistema de prevención y detección de intrusiones en la empresa Proauto C.A.* (Tesis de bachillerato. Universidad de las Américas, Quito, Ecuador*).* Recuperado de http://dspace.udla.edu.ec/handle/33000/10325

Janampa, H., Huamani, H. y Meneses, Y. (2021). *Snort Open Source como detección de intrusos para la seguridad de la infraestructura de red.* Recuperado de http://scielo.sld.cu/pdf/rcci/v15n3/2227-1899-rcci-15-03-55.pdf

Kaspersky. (2019). *Ataques DDoS en el primer trimestre de 2019.* Recuperado de https://securelist.lat/ddos-report-q1-2019/88828/

Losada, S. (2018). ¿Qué es ELK? ElasticSearch, Logstash y Kibana. https://openwebinars.net/blog/que-es-elk-elasticsearch-logstash-y-kibana/

Maquera Mamani, G. (2022). *Sistema de gestión de seguridad de la información y su relación con la calidad de servicio de las redes LAN en la municipalidad distrital de Ilabaya.* (Tesis título profesional. Universidad Privada de Tacna, Perú). Recuperado de https://repositorio.upt.edu.pe/handle/20.500.12969/2524

Miño Verdezoto, C. (2020). *Diseño de Implementación de un sistema de seguridad de detección de intrusos (IDS) para la plataforma OpenStack utilizando código abierto*. (Trabajo de titulación profesional. Escuela Superior Politécnica de Chimborazo, Ecuador). Recuperado de http://dspace.espoch.edu.ec/handle/123456789/14504

Olguin, M., Rivera, I. y Perez, P. (2006). Sistema de Detección de Intrusos (IDS), Seguridad en Internet. Recuperado de https://www.redalyc.org/articulo.oa?id=402640447006

Ochoa Palomino, A. (2019). *Diseño de una red de seguridad informática para la protección del sistema web de un call center ante ataque informáticos aplicando la norma ISO*. Recuperado de https://repositorioacademico.upc.edu.pe/handle/10757/625726

Perdigón, R. (2022). Suricata como detector de intrusos para la seguridad en redes de datos empresariales. CIENCIA UNEMI, 15(39) ,44-35. Recuperado de https://doi.org/10.29076/issn.2528-7737vol15iss39.2022pp44-53p

Penz, R. (2014). How to setup a Mikrotik RouterOS with Suricata as IDS. https://robert.penz.name/849/howto-setup-a-mikrotik-routeros-with-suricata-as-ids/

Pintacur, Fernández, B. (2019). *Diseño e implementación de una red de datos y seguridad perimetral de la empresa Corporación Cayman S.A.C.* (Tesis de título. Universidad Tecnológica del Perú). Recuperado de https://repositorio.utp.edu.pe/handle/20.500.12867/5933

Quintero, J. (2018). *Implementación de un sistema de detección de intrusos en la red interna de la alcaldía de Montería usando software libre*. (Tesis de bachillerato. Universidad Nacional Abierta y a Distancia, UNAD, Colombia). Recuperado de https://repository.unad.edu.co/bitstream/10596/17438/1/8867918.pdf

Rangel, J. (2021). *Sistema de detección de intrusiones y gestión de eventos e información de seguridad basados en nuevas tecnologías de código abierto*. (Tesis de título. Universidad de Quintana Roo, México). Recuperado de http://risisbi.uqroo.mx/handle/20.500.12249/2782

Ramirez, A. (2009). Estudio de una plataforma de detección de intrusos Open Source. Recuperado de https://upcommons.upc.edu/handle/2099.1/7483?locale-attribute=en

Rodríguez Gutiérrez|, A. F. (2016). Diseño de un sistema de detección de intrusos con Snort para la compañía Silverit SAS. (Tesis de bachillerato. Universidad Piloto de Colombia, Bogotá, Colombia). Recuperado de http://repository.unipiloto.edu.co/handle/20.500.12277/2706

Riveros Paraguay, J.K. (2019). *Implementación de políticas de seguridad informática para mejorar el acceso y la seguridad lógica de la red en la oficina departamental de estadística e informática de Junín.* (Tesis de titulación. Universidad Nacional del centro del Perú). Recuperado de https://repositorio.uncp.edu.pe/handle/20.500.12894/5434

Sánchez Cunalata, D. (2017). *Implementación de un sistema de monitoreo y protección de datos en la red de la facultad de ingeniería en sistemas, electrónica e industrial*. (Tesis de bachillerato. Universidad Técnica de Ambato, Ambato, Ecuador). Recuperado de http://repositorio.uta.edu.ec/jspui/bitstream/123456789/25707/1/Tesis_t1254si.pdf

Sánchez Restrepo, J. (2017). *Sistema de detección de intrusos mediante Snort y comparación con otras aplicaciones.* (Tesis de bachillerato. Universidad Estatal del Sur de Manabí, Manabí, Ecuador). Recuperado de http://repositorio.unesum.edu.ec/bitstream/53000/842/1/UNESUM-ECU-COMPR-19.pdf

Treneman, A. (2018). *¿Qué tan protegidos están los peruanos en Internet?* Recuperado de https://peru21.pe/tecnologia/ciberseguridad-seguros-peruanos-internet-entrevista-396789-noticia/

Vieira, J. F. (2016). *Desenvolvimiento de un IDS basado en técnicas de Data Mining. (*Tesis de bachillerato. Universidades de Santa Cruz do Sul, Rio Grande do Sul, Brasil). Recuperado de http://repositorio.unisc.br:8080/jspui/handle/11624/2131

Văn Hoàng Anh. (11 agosto de 2018). Config Suricata on Ubuntu 16.04.Youtube. https://www.youtube.com/watch?v=QtjLttQOJw0

Yauri, E. (2017). *Aplicación móvil para sistemas de detección y prevención de intrusiones basados en Snort.* (Tesis de bachillerato. Universidad Nacional de Ingeniería, Lima, Perú). Recuperado de https://alicia.concytec.gob.pe/vufind/Record/UUNI_a51439257fe31beb809f807c4d21b89d/Details

Anexos

Anexo 1: **Formato de Encuesta**: I**mplementación de una plataforma Server Open Source en la Detección y Prevención de Intrusos (IDS e IPS) en la Red de Datos"**

Valora de acuerdo a la siguiente escala: marca con una **"X"** el casillero de su preferencia.
(1) Totalmente desacuerdo
(2) En desacuerdo
(3) Ni de acuerdo ni en desacuerdo
(4) De acuerdo
(5) Totalmente de acuerdo

Variables		Cuestionario	Valoración				
Sistema de Detección Prevención de Intrusos	**Dimensión Fiabilidad**		1	2	3	4	5
	1	¿Considera usted que el software Open Source IDS sea eficiente en analizar en el tráfico entrante?					
	2	¿Considera usted que el software Open Source IDS sea efectivo como lo es un software IPS con licencia?					
	3	¿Considera usted que implementar software Open Source IDS demande de mucha inversión?					
	Dimensión Seguridad						
	4	¿Considera usted la importancia de la seguridad en su centro de datos?					
	5	¿Considera usted que el software Open Source IDS ayudaría a sus problemas de contrarrestar ataques cibernéticos?					
	6	¿Considera usted que un antivirus sea basto en salvaguardar la información de una pyme o mediana empresa?					
	Dimensión Flexibilidad						
	7	¿Considera usted que el software Open Source IDS cuente con soporte de ayuda?					
	8	¿Considera usted que el software Open Source IDS cuente con actualizaciones constantes en su base de datos?					
	9	¿Considera usted que sería necesario implementar un software IDS en la empresa donde trabaja					
	Dimensión Inspección						
	10	¿Considera usted que el software Open Source IPS registra los eventos anómalos en el tráfico de paquetes entrantes?					
	11	¿Considera usted que el software Open Source IPS realice identificación de vulnerabilidades?					
	Dimensión Funcionalidad						
	12	¿Considera usted que el software Open Source IPS monitorea en tiempo real el tráfico?					
	13	¿Considera usted que el software Open Source IPS realice acciones para contrarrestar los ataques?					
	Dimensión Observación						
	14	¿Considera usted que el software Open Source IPS muestra los resultados en tiempo real?					
	15	¿Considera usted que el software Open Source IPS muestre resultados con gráficos mediante la integración con otros software?					

ANEXO 2: MATRIZ DE CONSISTENCIA

<table>
<tr><th>TITULO</th><th>PROBLEMA GENERAL Y ESPECIFICOS</th><th>OBJETIVO GENERAL Y ESPECIFICO</th><th>HIPOTESIS GENERAL Y ESPECIFICO</th><th>VARIABLES, DIMENSIONES E INDICADORES</th><th>DISEÑO DE INVESTIGACIÓN</th><th>MÉTODOS Y TÉCNICAS DE INVESTIGACIÓN</th><th>POBLACIÓN Y MUESTRA DE ESTUDIO</th></tr>
<tr><td rowspan="6">Implementación de una plataforma Server utilizando la arquitectura Suricata Open Source</td><td>Problema General</td><td>Objetivo General</td><td>Hipótesis General</td><td>Variable Independiente
Implementación de una plataforma Server utilizando la Arquitectura Suricata Open Source</td><td rowspan="6">Pre-Experimental</td><td rowspan="6">Técnicas:

Entrevista

De muestreo:

Estadística inferencial
Estadística descriptiva

Recolección de datos:

Encuesta por cuestionario

De Procesamiento:

Razones
Porcentaje</td><td>Población

12 pequeñas o medianas empresas</td></tr>
<tr><td>¿De qué manera la implementación de una plataforma Server, utilizando la arquitectura Suricata open Source cumple con la prevención y detección de intrusos en una red perimetral?</td><td>Implementar una plataforma Server utilizando la arquitectura Suricata Open Source, para la detección y prevención de intrusos en la red.</td><td>La implementación de una plataforma Server utilizando la arquitectura Suricata Open Source cumple significativamente con la detección y prevención de intrusos</td><td>Dimensiones:
Diseñar, ejecución, producción</td><td rowspan="2">Muestra:

12 personas del área de TI</td></tr>
<tr><td>Problemas Específicos</td><td>Objetivo Específicos</td><td>Hipótesis Específicas</td><td>Indicadores:
Elaboración diagrama de red.
Desarrollo y configuración
Resultados e informes</td></tr>
<tr><td>1. ¿Cuál es la conformidad del software actual para el IDS/IPS en la red?</td><td>1. Determinar la conformidad del software actual para IDS/IPS en la red.</td><td>El software actual para IDS/IPS en la red es conforme.</td><td>Variable Dependiente:

Sistema de Detección y Prevención de Intrusos</td><td>Tipo de muestra:

No probabilística e intencionada</td></tr>
<tr><td>2. ¿Cómo implementar una plataforma Server utilizando la arquitectura Suricata Open Source?</td><td>2. Diseñar e implementar una plataforma Server utilizando la arquitectura Suricata Open Source.</td><td>Se logra implementar la plataforma Server utilizando la arquitectura Suricata Open Source.</td><td>Dimensiones: Fiabilidad, seguridad, Flexibilidad Inspección, Funcionalidad, Observación|</td></tr>
<tr><td>3. ¿Cuál es la conformidad de la implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS?</td><td>3. Determinar la conformidad de la implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS</td><td>La implementación de la plataforma Server utilizando la arquitectura Suricata Open Source para el IDS/IPS es conforme.</td><td>Indicadores: Conformidad sobre el Sistema: Eficiente, efectivo, registra eventos, identifica anomalías, supervisa tráfico de datos, tiempo de respuesta, resultado en el monitoreo, actualización
monitoreo en tiempo real</td></tr>
</table>

Anexo 3

Matriz de datos – PreTest

Nro de	Dimensión 1: Fiabilidad			Dimensión 2: Seguridad			Dimensión 3: Flexibilidad			Dimensión 4: Inspección		Dimensión 5: Funcionalidad		Dimensión 6: Observación	
Encuestados	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
E1	4	4	2	4	4	1	3	4	5	4	5	4	4	5	4
E2	5	4	2	4	4	2	3	4	5	4	4	5	5	5	3
E3	5	5	5	5	4	4	5	5	5	5	4	5	4	5	5
E4	5	4	2	5	5	2	4	4	3	5	5	5	4	5	5
E5	4	4	2	5	5	2	4	4	4	4	4	5	5	4	4
E6	4	4	2	4	3	2	4	5	5	4	5	4	5	4	4
E7	4	4	2	5	4	1	4	4	4	4	4	5	4	4	4
E8	4	4	2	5	4	1	4	5	5	4	4	5	4	4	5
E9	5	5	5	5	4	1	5	5	5	5	5	5	5	5	5
E10	4	4	5	5	5	2	4	4	5	5	5	5	5	5	5
E11	4	4	3	5	4	4	3	3	4	4	3	5	5	5	5
E12	4	4	2	5	5	2	3	5	5	5	5	5	5	5	4

Anexo 4

Matriz de datos – PostTest

Nro de	Dimensión 1: Fiabilidad			Dimensión 2: Seguridad			Dimensión 3: Flexibilidad			Dimensión 4: Inspección		Dimensión 5: Funcionalidad		Dimensión 6: Observación	
Encuestados	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
E1	5	5	3	5	5	1	3	4	5	5	5	5	5	5	5
E2	5	4	2	5	5	2	3	4	5	5	5	5	5	5	5
E3	5	5	2	5	4	2	5	5	5	5	5	5	5	5	5
E4	5	5	2	5	5	2	4	4	4	5	5	5	4	5	5
E5	5	5	2	5	5	2	4	4	4	5	5	5	5	5	5
E6	4	4	2	4	4	2	4	5	5	4	5	4	5	5	5
E7	4	4	2	5	4	1	4	4	4	4	5	5	4	5	5
E8	5	4	2	5	4	1	4	5	5	5	5	5	5	5	5
E9	5	5	3	5	4	1	5	5	5	5	5	5	5	5	5
E10	4	4	2	5	5	2	4	4	5	5	5	5	5	5	5
E11	4	4	3	5	4	2	3	4	4	5	5	5	5	5	5
E12	4	4	2	5	5	2	3	5	5	5	5	5	5	5	5

Anexo 5

Matriz de evaluación de expertos

UPN UNIVERSIDAD PRIVADA DEL NORTE

"IMPLEMENTACIÓN DE UNA PLATAFORMA SERVER UTILIZANDO LA ARQUITECTURA SURICATA OPEN SOURCE PARA LA DETECCIÓN Y PREVENCIÓN DE INTRUSOS EN LA RED"

FICHA PARA VALIDACIÓN DEL INSTRUMENTO - FICHA DE OBSERVACIÓN

I. REFERENCIA

1.1. **Experto:** Ortiz Quintana, Danny Michael
1.2. **Especialidad:** Ingeniería de Sistemas
1.3. **Cargo actual:** jefe de Sistemas
1.4. **Grado académico:** Magister
1.5. **Institución:** Universidad ESAN
1.6. **Tipo de instrumento:** Cuestionario
1.7. **Lugar y fecha:** Lima, 21 de abril del 2023

II. TABLA DE VALORACIÓN POR EVIDENCIAS

N.°	EVIDENCIAS	VALORACIÓN					
		5	4	3	2	1	0
1	Pertinencia de indicadores		X				
2	Formulación con lenguaje apropiado	X					
3	Adecuado para los sujetos en estudio		X				
4	Facilita la prueba de hipótesis		X				
5	Suficiente para medir la variable		X				
6	Facilita la interpretación del instrumento	X					
7	Acorde al avance de la ciencia y tecnología	X					
8	Expresado en hechos perceptibles	X					
9	Tiene secuencia lógica		X				
10	Basado en aspectos teóricos		X				
	Total	20.0	24.0	-	-	-	-

Coeficiente de valoración porcentual c= 88%

III. OBSERVACIONES Y/O RECOMENDACIONES

Recomiendo detallar más la definición de la población a la que va esta implementación, así como la sustentación del porqué la elección del software sobre otros que existen en el mercado.

Con respecto a la evaluación ya de la implementación lo veo correcto y detallado.

GULDA & CIA. S.A.C.
DANNY ORTIZ QUINTANA
JEFE DE SISTEMAS

FICHA PARA VALIDACIÓN DEL INSTRUMENTO - FICHA DE OBSERVACIÓN

I. REFERENCIA

1.1. Experto: Cachay Arana, Jorge Andrés
1.2. Especialidad: Ingeniería de Sistemas
1.3. Cargo actual: Jefe de Sistemas
1.4. Grado académico: Ing. De Sistemas
1.5. Institución: Universidad Señor de Sipan
1.6. Tipo de instrumento: Cuestionario
1.7. Lugar y fecha: Lima, 19 de febrero del 2023.

II. TABLA DE VALORACIÓN POR EVIDENCIAS

N°	EVIDENCIAS	VALORACIÓN					
		5	4	3	2	1	0
1	Pertinencia de indicadores	X					
2	Formulación con lenguaje apropiado		X				
3	Adecuado para los sujetos en estudio		X				
4	Facilita la prueba de hipótesis	X					
5	Suficiente para medir la variable	X					
6	Facilita la interpretación del instrumento	X					
7	Acorde al avance de la ciencia y tecnología	X					
8	Expresado en hechos perceptibles	X					
9	Tiene secuencia lógica	X					
10	Basado en aspectos teóricos	X					
	Total	40	8				

Coeficiente de valoración porcentual c= 96%

III. OBSERVACIONES Y/O RECOMENDACIONES

Se evidencia correctamente que las variables tienen relación con las preguntas formuladas del cuestionario.

Se recomienda revisar la redacción de las hipótesis específicas.

.....
Firma y sello del Experto

FICHA PARA VALIDACIÓN DEL INSTRUMENTO - FICHA DE OBSERVACIÓN

I. REFERENCIA

1.1. Experto: Cárdenas García, Orlando José
1.2. Especialidad: Ingeniería de Sistemas
1.3. Cargo actual: Analista de Sistemas
1.4. Grado académico: Ing. De Sistemas
1.5. Institución: Universidad Privada del Norte
1.6. Tipo de instrumento: Cuestionario
1.7. Lugar y fecha: Lima, 14 de febrero del 2023.

II. TABLA DE VALORACIÓN POR EVIDENCIAS

N°	EVIDENCIAS	VALORACIÓN					
		5	4	3	2	1	0
1	Pertinencia de indicadores	X					
2	Formulación con lenguaje apropiado	X					
3	Adecuado para los sujetos en estudio	X					
4	Facilita la prueba de hipótesis	X					
5	Suficiente para medir la variable	X					
6	Facilita la interpretación del instrumento	X					
7	Acorde al avance de la ciencia y tecnología	X					
8	Expresado en hechos perceptibles	X					
9	Tiene secuencia lógica	X					
10	Basado en aspectos teóricos	X					
	Total	50					

Coeficiente de valoración porcentual c= 100%

III. OBSERVACIONES Y/O RECOMENDACIONES

Desde mi punto de vista está bien argumentado la hipótesis, variables, población y muestra acorde al título del desarrollo tecnológico.

.....
Firma y sello del Experto