



**ANÁLISIS DE SEGURIDAD Y CIBERATAQUES PARA ENTRENAR SISTEMAS
AUTÓNOMOS EN REDES INDUSTRIALES**

Alumno: Esteban Emilio Mas

Profesora de Trabajo Final: Marcela Rosalba Samela

Tutor técnico: Jorge Alejandro Kamlofsky

Trabajo Final de carrera presentado para obtener el título de
Licenciatura en Gestión de Tecnología Informática

Diciembre, 2025

Resumen

La automatización y la inteligencia artificial han permitido a las fábricas y a las infraestructuras críticas optimizar su productividad y reducir costos. Sin embargo, esta revolución en el sector industrial también ha planteado un desafío significativo: las brechas de seguridad en las redes industriales. Estas vulnerabilidades son atractivas para los cibercriminales, con la expectativa de que los ataques contra fábricas y edificios inteligentes se intensifiquen y se vuelvan más peligrosos en los próximos años. Dado que el desafío de la ciberseguridad es considerable, pero las ventajas de la Industria 4.0 son mayores, es crucial aplicar una metodología que fortalezca las redes industriales sin comprometer la eficiencia de los nuevos procesos de fabricación. En este trabajo se presenta una metodología basada en buenas prácticas y en la implementación de honeypots. La investigación se centró en la implementación de un honeypot llamado CONPOT, que simula ser un equipo ICS/SCADA vulnerable y registra toda la actividad sobre él. El propósito de este enfoque fue comprender mejor las tácticas y procedimientos utilizados por los atacantes, así como identificar vulnerabilidades y errores en configuraciones. Para complementar esta estrategia, se desarrollaron tres aplicaciones Python: una para emular un sistema SCADA utilizando Scapy para escuchar tráfico TCP/IP y SNMP, otra para analizar datos capturados en tiempo real con Pandas para detectar actividades sospechosas, y una tercera para conectarse a la API de ELK, descargar registros, analizarlos y detectar patrones o anomalías. Estas aplicaciones permitieron enviar registros a ELK, detectar patrones y recomendar medidas preventivas. La investigación aporta una guía práctica para fortalecer la ciberseguridad en la Industria 4.0 sin comprometer la eficiencia operativa.

Palabras clave: aprendizaje automático, aprendizaje por refuerzo, ciberseguridad, CONPOT, honeypot, ICS, inteligencia artificial, protocolos, redes industriales, redes neuronales, SCADA, sistemas expertos.

Abstract

Automation and artificial intelligence have enabled factories and critical infrastructures to optimize their productivity and reduce costs. However, this revolution in the industrial sector has also posed a significant challenge: security gaps in industrial networks. These vulnerabilities are attractive to cybercriminals, with the expectation that attacks against factories and smart buildings will intensify and become more dangerous in the coming years. Given that the cybersecurity challenge is considerable, but the advantages of Industry 4.0 are greater, it is crucial to apply a methodology that strengthens industrial networks without compromising the efficiency of new manufacturing processes. This work presents a methodology based on best practices and the implementation of honeypots. The research focused on implementing a honeypot called CONPOT, which simulates a vulnerable ICS/SCADA device and records all activity on it. The purpose of this approach was to better understand the tactics and procedures used by attackers, as well as to identify vulnerabilities and configuration errors. To complement this strategy, three Python applications were developed: one to emulate a SCADA system using Scapy to listen to TCP/IP and SNMP traffic, another to analyze real-time captured data with Pandas to detect suspicious activities, and a third to connect to the ELK API, download logs, analyze them, and detect patterns or anomalies. These applications allowed sending logs to ELK, detecting patterns, and recommending preventive measures. The research provides a practical guide to strengthen cybersecurity in Industry 4.0 without compromising operational efficiency.

Keywords: artificial intelligence, CONPOT, cybersecurity, expert systems, honeypot, ICS, industrial networks, machine learning, neural networks: protocols, reinforcement learning, SCADA.

Dedicatoria

Este Trabajo está dedicado a mi esposa, Ana Chilano, por su acompañamiento y sostén para que pueda desarrollar una carrera universitaria en conjunto con nuestro proyecto familiar, sin ella no hubiera podido balancear mis deberes como padre y como alumno universitario.

A mis hijos, Matilda y Joaquín, por ser mi fuente de energía y motivación en todo momento.

A mis padres por haberme enseñado a nunca bajar los brazos, por haberme inculcado los valores con que rijo mi día a día y por sobre todos por estar siempre.

A mis hermanas, les agradezco por su presencia constante en mi vida, por su amor incondicional y por ser mis mejores amigas.

Agradecimientos

Mi especial agradecimiento a Guillermo Calabrese por haberme insistido para que retome mis estudios universitarios y por haberme apoyado y facilitado en poder complementar el trabajo y el estudio.

A mis hijos y esposa, quienes son mi motor y mi mayor inspiración.

A mis padres, quienes, a través de su amor, paciencia, buenos valores, ayudaron a trazar mi camino.

A mi tutor, el profesor Jorge Kamlofsky, por guiarme y asesorarme en el desarrollo de mi trabajo de investigación.

A todos ellos, y a cada uno de ellos, (no alcanzan los adjetivos calificativos para agradecerles), muchísimas gracias.

Índice General

Índice de Gráficos	8
Índice de Tablas	10
Capítulo 1	11
1.1. Introducción	11
1.2. Tema de Investigación	13
1.3. Justificación.....	13
1.4. Pregunta de Investigación	14
1.5. Hipótesis.....	14
1.6. Objetivo General	14
1.7. Objetivos Específicos.....	14
1.8. Metodología de Investigación	15
Capítulo 2.....	16
Marco Teórico	16
2.1. La cuarta Revolución Industrial y la Transformación Digital.....	16
2.2. Las Tecnologías de la Información y las Tecnologías de la Operación (IT/OT). 17	
2.3. Convergencia de Integración IT/OT.....	19
2.4. Ventajas de la Integración IT/OT	22
2.5. Desventajas de la Convergencia IT/OT.....	22
2.6. Antecedentes de Ciberataques	23
2.7. Redes Industriales.....	23
2.8. Tipos de Información.....	26
2.9. Componentes de Una Red Industrial	26
2.10. Arquitectura de Redes Industriales.....	30
2.11. Honeypot	37
2.12. Lenguaje de Programación Python.....	39
2.13. ELK (Elasticsearch, Logstash y Kibana).....	39
2.14. Ciberseguridad.....	40
2.15. Estado del arte	46
2.16. Aplicación de Inteligencia Artificial en la Ciberseguridad Industrial.....	48
2.17. Técnicas de Inteligencia Artificial Aplicadas a la Ciberseguridad.....	48
2.18. Aplicaciones de las Técnicas de iA en Ciberseguridad.....	49
2.19. Limitaciones de la Inteligencia Artificial en la Ciberseguridad.....	50

Desarrollo Técnico	52
3.1. Diseño, Implementación y Análisis.....	52
3.2. Primera Implementación. T-POT/CONPOT	53
3.3. Evidencias Encontradas Para Su Posterior Análisis.	54
3.4. Segunda Implementación. Desarrollo del Honeypot con Python.....	57
3.5. Evidencias Encontradas Para Su Posterior Análisis.	60
3.6. Tercera implementación. Desarrollo del emulador SCADA con Python.....	62
3.7. Desarrollo y Configuración.	64
3.8. Instalación Infraestructura Sobre Contenedores.....	67
3.9. Configuración servidor Elasticsearch.....	69
3.10. Configuración Servidor Logstash.....	71
Resultados de la investigación	77
4.1 Datos obtenidos de las implementaciones	77
4.1.1 Primera implementación: honeypot industrial.....	77
4.1.2 Segunda implementación: honeypot SSH en Python	78
4.1.3 Tercera implementación: emulador SCADA y monitoreo de red	78
4.2 Visualización y análisis de datos en ELK	79
4.2.1 Registros ELK	79
4.2.2 Consultas estructuradas	81
4.3 Evaluación de la eficacia del sistema	82
4.4 Discusión de los resultados.....	83
Conclusiones y Aportes.....	84
5.1. Conclusión.....	84
5.2. Aportes de la investigación.....	85
5.3. Líneas Futuras de Investigación	85
Acrónimos.....	88
Referencias.....	90

Índice de Gráficos

Figura 1 - Pirámide de los sistemas involucrados en la gestión empresarial.	20
Figura 2 - Arquitectura de referencia empresarial Purdue.....	30
Figura 3 - Modelo de arquitectura de referencia de la Industria 4.0.....	31
Figura 4 - Siete niveles del Modelo OSI.....	33
Figura 5 - Equivalencia del protocolo TCP/IP vs Modelo OSI.....	34
Figura 6 - Esquema Honeynet.....	38
Figura 7 - Muestra servicios y honeypots en ejecución y sus puertos de escucha.....	54
Figura 8 - Dashboard de estadísticas de ataques al Honeypot CONPOT.....	55
Figura 9 - Dashboard Histograma de ataques al Honeypot CONPOT.....	55
Figura 10 - Mapa con la ubicación geográfica de los ataques al Honeypot.....	55
Figura 11 - Dashboards de reputación, países de origen, tipos de eventos, protocolos protocolos y distribución horaria de los ataques al Honeypot CONPOT.....	56
Figura 12 - Top 10 de los proveedores e IPs origen.....	56
Figura 13 - Registro de los ataques.....	57
Figura 14 - Listado de archivos accedidos.....	58
Figura 15 - Estadísticas Servidor Web.....	60
Figura 16 - Evidencias sobre los registros del log “Últimos comandos”.....	61
Figura 17 - Evidencias sobre los registros del log “Últimos países conexiones”.....	62
Figura 18 - Evidencias sobre los registros del log “Últimas muestras obtenidas.....	62
Figura 19 - Desarrollo Emulador SCADA.....	64
Figura 20 - Muestra de la función para el análisis de tráfico TCP, UDP y SNMP.....	65
Figura 21 - Muestra de la función para análisis de caducidad de contraseñas.....	66
Figura 22 - Muestra de la función para cambio de contraseña por periodo de vigencia	66
Figura 23 - Instalación Docker.....	67
Figura 24 - Búsqueda de la imagen en Docker Hub.....	67
Figura 25 - Descarga de las imágenes de los contenedores.....	68
Figura 26 - Verificación de la descarga de los contenedores.....	68
Figura 27 - Creación de la red a usar por los contenedores.....	68
Figura 28 - Iniciación de los contenedores.....	69
Figura 29 - Verificación del estado de los contenedores.....	69
Figura 30 - Archivo de configuración Elasticsearch. Fuente propia.....	69

Figura 31 - Muestra de registro indexado en Elasticsearch.....	71
Figura 32 - Archivo de configuración de salida de Logstash a Elasticsearch.....	72
Figura 33 - Muestra 1 de registro indexado en Elasticsearch.....	73
Figura 34 - Muestra 2 de registro indexado en Elasticsearch.....	74
Figura 35 - Muestra 3 de registro indexado en Elasticsearch.....	74
Figura 36 - Muestra 4 de registro indexado en Elasticsearch.....	75
Figura 37 - Muestra 5 de registro indexado en Elasticsearch.....	75
Figura 38 - Registro de eventos del Honeypot CONPOT en entorno SCADA.....	78
Figura 39 - Extracto del log de comandos capturados en el honeypot SSH.....	78
Figura 40 - Alerta generada por actividad anómala en el emulador SCADA.....	79
Figura 41 - Ataque de fuerza SSH.....	79
Figura 42 - Ataque SNMP flood desde Asia.....	79
Figura 43 - Escaneo de puertos industriales.....	80
Figura 44 - Inyección de comandos HTTP sospechosos.....	80
Figura 45 - Acceso no autorizado a endpoint SCADA.....	80
Figura 46 - Registro de Alertas.....	81
Figura 47 - Ejemplo de consulta estructurada en Kibana para filtrar eventos relacionados con el protocolo SSH en los últimos siete días.....	81

Índice de Tablas

Tabla 1 - Comparativa de las IT/OT.....	18
Tabla 2 - Características de una Red Industrial.....	25
Tabla 3 - Resultados cuantitativos obtenidos por cada implementación.....	82
Tabla 4 - Resumen de líneas futuras de investigación propuestas.....	87

Capítulo 1

1.1. Introducción

El concepto de ciberseguridad integral se estableció como una estrategia de la Unión Europea para lograr un entorno seguro (Comisión Europea, 2013) . En este marco, la Comisión Europea (2013) considera que, “para que el ciberespacio siga siendo abierto y libre, deben aplicarse en línea los mismos valores y normas que la UE promueve fuera de línea” (p. 22). De este modo, se establece que los derechos, la democracia y el Estado de Derecho deben ser protegidos también en el ciberespacio.

No obstante, la libertad en los sistemas necesita de seguridad, dado que el ciberespacio debe ser protegido frente a acciones irregulares. Por lo tanto, el Estado debe custodiar y garantizar un ciberespacio libre y seguro, considerando que entre sus funciones se encuentran el control del acceso y la apertura, el respeto y la protección de los derechos, así como el mantenimiento de la fiabilidad e interoperabilidad de Internet.

Este estudio se justifica debido a la importancia que reviste la seguridad del ciberespacio, en particular en las redes de control industrial. Estas redes, en el contexto de la cuarta revolución industrial, conocida como Industria 4.0, se espera que se integren con una gran variedad de tecnologías emergentes con el fin de maximizar la eficiencia, lo cual las expone a nuevas amenazas.

Por otra parte, los avances en materia de comunicaciones y la disminución de los costes, han permitido la creación de una red, en donde, se producen diariamente aspectos positivos como resultado de la utilización del ciberespacio.

En tal sentido, el acceso de las empresas y las personas a las redes ha proporcionado un entorno con grandes oportunidades. No obstante, este mismo acceso ha generado nuevos retos en materia de ciberseguridad, donde la protección de los sistemas se ve comprometida ante múltiples situaciones que emplean el ciberespacio como medio para alterar las actividades de organizaciones o personas.

Este escenario ha dado lugar a un nuevo contexto en el que la Inteligencia Artificial (IA), asociada a diversos mecanismos tecnológicos, ha contribuido significativamente a la automatización de procesos de ciber inteligencia (Gobierno de España, 2018). Entre sus aplicaciones más destacadas se encuentra la capacidad de evaluar, de manera sistemática y continua, la naturaleza, velocidad y gravedad de las amenazas cibernéticas. Esto se logra mediante el análisis de fuentes abiertas disponibles

en Internet, incluyendo tanto la red visible como la invisible (darknet), con el fin de identificar indicios de ataques en preparación o posibles filtraciones de información sensible relacionada con empleados y sistemas empresariales.

Para que la IA cumpla con estas funciones de forma efectiva, ha sido necesario entrenar sus algoritmos con grandes volúmenes de información actualizada, tanto en la fase de aprendizaje como en su implementación en entornos de producción. Este proceso implica un uso intensivo de recursos tecnológicos, particularmente en términos de procesamiento, memoria y consumo energético (Buczak & Guven, 2016).

En este contexto, la intervención humana es fundamental para asegurar y detectar pronósticos erróneos, los cuales podrían generar consecuencias graves. Por consiguiente, la IA facilita la automatización de las funciones habituales en ciberseguridad (Gobierno de España, 2018), aunque la intervención humana sigue siendo necesaria.

Es previsible que su empleo se generalice y se convierta en una mercancía (*commodity*), lo que reducirá los costos operativos y la necesidad de personal altamente especializado, aspectos importantes en un contexto de amenazas crecientes en número y complejidad en diversos casos, promovidas por organizaciones. No obstante, esta accesibilidad tecnológica también conlleva riesgos, ya que el uso masivo y sin supervisión de sistemas automatizados puede dar lugar a una falsa sensación de seguridad. Además, la estandarización de herramientas basadas en inteligencia artificial podría facilitar su adquisición y eventual aprovechamiento por parte de actores maliciosos, lo que incrementa la necesidad de establecer marcos éticos, regulatorios y técnicos que garanticen un uso responsable, seguro y controlado de estas tecnologías en el ámbito de la ciberseguridad.

En este contexto, se considera pertinente evaluar de manera específica el uso de la Inteligencia Artificial en redes industriales orientadas a la ciberseguridad. El análisis de su implementación en estos entornos permitirá comprender no solo sus beneficios operativos, sino también los riesgos asociados a su adopción masiva sin supervisión. Asimismo, los resultados obtenidos representarán un aporte significativo a la comprensión integral del papel que cumple la IA en la protección de infraestructuras críticas, reforzando la necesidad de desarrollar políticas de implementación seguras y éticamente responsables.

Existen dispositivos, llamados honeypot que son utilizados para alcanzar un “estado de seguridad”, ya que permiten simular un sistema que hace de señuelo ante los

ciberdelincuentes, adquirir nuevos métodos de ataque y distraerlos del objetivo real. Uno de estos es el proyecto llamado CONPOT (CONPOT, s.f.), honeypot de ICS/SCADA de baja interacción mantenido por el equipo de desarrollo de CONPOT y de The HoneyNet Project (The HoneyNet Project, s.f.). Las principales características de CONPOT son que está diseñado para modificar, extender e implementar fácilmente dispositivos ICS simulados. La configuración predefinida emula un sistema PLC Siemens, modelo SIMATIC S7-200 con funcionalidad básica, y contiene un módulo de E/S y un CP 443-1, procesador de comunicación que permite la conexión a SIMATIC a través de una red Ethernet (Siemens, s.f.).

CONPOT está diseñado para trabajar con protocolos de control industrial y con los módulos básicos para construir su propio ICS. Utiliza protocolos de control industrial comunes como Modbus, S7Comm, Bacnet e IPMI.

En este trabajo se implementan técnicas de IA utilizando como insumo datos recopilados por un entorno CONPOT. Los resultados son muy prometedores.

1.2. Tema de Investigación

Aplicación de honeypots industriales y sistemas de análisis en tiempo real basados en inteligencia artificial para mejorar la ciberseguridad en infraestructuras críticas ICS/SCADA.

1.3. Justificación

La transformación digital impulsada por la Industria 4.0 ha generado una creciente interconexión de sistemas de control industrial (ICS/SCADA) con redes externas, lo que ha incrementado significativamente su exposición a ciber amenazas. Las infraestructuras críticas, al ser componentes esenciales en sectores como la energía, el transporte o la manufactura, requieren mecanismos de protección avanzados que no solo reaccionen ante un ataque, sino que lo anticipen. En este contexto, la implementación de honeypots como sistemas de detección pasiva y el uso de algoritmos de inteligencia artificial para el análisis en tiempo real de los datos de conexión, representan enfoques innovadores que permiten identificar patrones de ataque, generar alertas tempranas y reforzar la defensa activa de los entornos industriales. La presente investigación busca contribuir al fortalecimiento de la ciberseguridad en estos sistemas, mediante una propuesta

tecnológica que integre emulación de entornos críticos, monitoreo continuo y análisis automatizado de amenazas.

1.4. Pregunta de Investigación

¿De qué manera la implementación de honeypots industriales y sistemas de análisis y monitoreo en tiempo real puede mejorar la detección de ciber amenazas y fortalecer la respuesta ante ciberataques en entornos ICS/SCADA?

1.5. Hipótesis

La implementación conjunta de honeypots industriales y sistemas de análisis en tiempo real basados en inteligencia artificial mejora significativamente la detección de ciber amenazas y fortalece la capacidad de respuesta ante ciberataques en entornos ICS/SCADA.

1.6. Objetivo General

Diseñar e implementar un sistema de defensa activa para entornos ICS/SCADA, basado en honeypots industriales y herramientas de monitoreo y análisis en tiempo real, con el fin de mejorar la detección de ciber amenazas y fortalecer la respuesta ante ataques cibernéticos.

1.7. Objetivos Específicos

1. Simular un entorno de control industrial mediante la implementación de honeypots ICS, con el fin de atraer tráfico malicioso y registrar intentos de acceso no autorizado.
2. Desarrollar módulos de captura y análisis de tráfico de red en tiempo real utilizando herramientas como Python, Scapy y Paramiko, orientados a detectar comportamientos sospechosos en protocolos industriales.
3. Integrar un sistema de visualización de registros basado en ELK Stack (Elasticsearch, Logstash y Kibana) para facilitar el monitoreo continuo y la identificación de patrones de ataque.

-
4. Evaluar la eficacia del sistema implementado en la detección y respuesta frente a intentos de intrusión, mediante la recopilación y análisis de evidencias registradas en el entorno experimental.

1.8. Metodología de Investigación

En el presente trabajo se utilizan metodologías de investigación cualitativa y cuantitativa.

Se adoptará un enfoque cualitativo para investigar en profundidad los potenciales ataques dirigidos al emulador SCADA. Se profundizará en las características específicas de estos ataques, tales como las direcciones IP implicadas, la naturaleza de la actividad malintencionada y los protocolos que se encuentren en juego. A través de esta óptica cualitativa, se aspira a comprender de manera íntegra las estrategias de los atacantes y las técnicas empleadas para vulnerar el sistema. Este entendimiento será fundamental para fundamentar decisiones más informadas y para el fortalecimiento de las medidas de seguridad.

En relación con la perspectiva cuantitativa, esta será esencial para reunir información numérica concreta y evaluable, con el objetivo de responder a cuestionamientos concretos y valorar la eficacia de la solución propuesta. Se anticipa la recolección de datos referentes al tráfico de red que es interceptado por la aplicación emuladora SCADA, incluyendo, pero no limitándose a, las direcciones IP involucradas, los protocolos en uso, las clases de paquetes y otras variables de interés. Esta información cuantificable facilitará la detección de comportamientos de tráfico atípicos, actividades asociadas con el protocolo SNMP y sentará las bases para decisiones informadas, evaluaciones de riesgos y la optimización constante de la seguridad en el entorno SCADA.

Capítulo 2

Marco Teórico

2.1. La cuarta Revolución Industrial y la Transformación Digital

Actualmente, en los sistemas para la producción de bienes de consumo se vive la cuarta revolución industrial o industria 4.0. En la 46° edición del Foro Económico Mundial (World Economic Forum, WEF, por sus siglas en inglés) efectuado en Davos, Suiza en el año 2016 fue donde los líderes mundiales conversaron sobre el comienzo de la Cuarta Revolución Industrial. En el foro se definió:

La cuarta revolución industrial, caracterizada por la convergencia de tecnologías digitales, físicas y biológicas, abarca avances en campos como la inteligencia artificial, el aprendizaje automático, la robótica, la nanotecnología, la impresión 3D, la genética y la biotecnología. Esta transformación tecnológica provocará, en los próximos años, una disrupción generalizada no solo en los modelos de negocio, sino también en la estructura de los mercados laborales y sociales (Schwab, 2016).

Siguiendo con la puesta en marcha de la Industria 4.0 que se sustenta en la digitalización y la capacidad que deben tener las organizaciones para gestionar de forma combinada los conocimientos, materiales, técnicas y procesos de manera óptima para crear productos y servicios diferentes a los ofrecidos por otras organizaciones y así posicionarse en el mercado como lo menciona Torres & Lamenta, (2015).

El término “Industria 4.0” tiene su origen en Alemania, donde fue presentado en 2011 como parte de una estrategia industrial promovida por el gobierno alemán para consolidar el liderazgo tecnológico del país. Esta política, centrada en el desarrollo de tecnologías avanzadas, buscaba fomentar la automatización, la digitalización de procesos y el uso de sistemas ciberfísicos aplicados a la industria manufacturera (Kagermann, Wahlster & Helbig, 2013).

El concepto de transformación digital se refiere a la incorporación de tecnologías digitales en todas las fases del sistema productivo para mejorar su eficiencia, adaptabilidad y competitividad. Este proceso es un pilar fundamental del paradigma conocido como Industria 4.0, el cual se caracteriza por la integración de sistemas ciberfísicos, la automatización avanzada, el análisis de datos y la interconectividad industrial.

La digitalización se puede entender, en términos generales, como la adopción masiva de la tecnología digital a través de los servicios y los dispositivos conectados. Hay muchas definiciones de digitalización, a veces referida a la revolución digital, haciendo una conexión implícita con la revolución industrial.

En esta misma línea Xu, et al.(2014) mencionan que la industria ha incorporado desde sus inicios una gran cantidad de mejoras tecnológicas y que era de esperarse la incorporación de las tecnologías de la información (autómatas programables, sistemas embebidos, software de supervisión y control, Sistemas de Ejecución de Manufactura, procedimientos digitales, internet de las cosas, Big Data, robótica, inteligencia artificial, realidad aumentada, la computación en la nube entre otras) debido a su efecto positivo en la productividad. Sin embargo, históricamente se ha demostrado que para obtener el mayor beneficio de una tecnología, cualquiera que esta sea, se deben de modificar los comportamientos de la organización y los métodos del trabajo en las áreas afectadas por esta.

Lo anterior nos hace pensar que aquellas organizaciones que logren gestionarse mejor serán aquellas que logren integrar completamente IT y OT permitiendo mejoras en la calidad de productos y servicios a bajo costo hasta una optimización del recurso de toda la organización y minimizando los tiempos de inactividad.

2.2. Las Tecnologías de la Información y las Tecnologías de la Operación (IT/OT).

Los sistemas tecnológicos de información (IT, por sus siglas en inglés) se definen como el conjunto de recursos, procedimientos y técnicas empleados para el procesamiento, transmisión y almacenamiento de información dentro de una organización. Estos sistemas permiten la gestión digital de datos y la toma de decisiones basada en información estructurada (Laudon & Laudon, 2020).

Por otro lado, la tecnología operacional (OT, por sus siglas en inglés) se refiere a los sistemas que gestionan, controlan y supervisan procesos físicos mediante dispositivos como sensores, actuadores, controladores programables (PLC) y sistemas de supervisión en tiempo real. De forma más precisa, la OT se define como “el hardware y software que detecta o provoca un cambio a través de la supervisión directa y/o el control de dispositivos físicos, procesos y eventos en la empresa” (International Society of Automation [ISA], 2019).

A partir de los trabajos de Pancho y Galarza (2014) y Ospina (2019), en la Tabla 1 se presenta una comparación entre las tecnologías de la información (IT) y las tecnologías operativas (OT), que permite visualizar de forma clara las principales diferencias existentes entre ambas en diversos aspectos. No obstante, dichas diferencias no han representado un obstáculo para su convergencia, la cual ha sido fundamental para fortalecer los procesos de gestión industrial.

Tabla 1111

Comparativa de las IT/OT

Característica	Tecnologías de la información	Tecnología operacional
Propósito	Gestión de información. Automatización de procesos de negocio	Manejo de activos. Gestión de procesos de control
Entornos	Multifuncional. Condiciones favorables. Se prioriza la rapidez. Permite reinicio. Actualización entre 2 y 5 años	Funciones limitadas. Condiciones desfavorables. Se prioriza confiabilidad y continuidad. Actualización entre 10 y 20 años
Arquitectura	Transaccional o batch, bases de datos relacionales o en texto	Tiempo real, software embebido
Interfaces	Navegador web, terminales y teclado	Sensores, displays codificados
Compatibilidad	Selección de equipos por funcionalidad (rápidos y adaptables)	Dependencia de terceros. Tolerancia a fallos e imprevistos
Homogeneidad	Funcionalidades claras. Equipamiento homogéneo	Variabilidad de entornos. Equipos de distintos fabricantes y funciones
Seguridad	Concepto maduro. Seguridad física y lógica	Aplicación parcial de conocimientos IT. Exposición a vulnerabilidades externas
Gestores	Director tecnológico, ingenieros informáticos, finanzas y adquisiciones	Ingenieros técnicos. Gerentes de línea de procesos o producción

Conectividad Comunicación	Redes corporativas con protocolo IP, acceso estándar por puertos	Protocolos de procesos. Redes mixtas (IP y Wireless)
Ejemplos	ERP, SCM, CRM, correo electrónico	SCADA, PLC, sistemas de control, modelado de procesos productivos

Nota: Comparación de las características de las Tecnologías de la Información (TI) y la Tecnología Operacional (TO).

Desde el inicio de la digitalización de procesos de negocios como la automatización de los procesos de producción han existido numerosos retos tecnológicos para la integración en todos los aspectos. Sin embargo, no son los únicos ya que existen también retos organizativos. Estos hacen referencia a la relación que existe entre la estructura organizativa, los procesos, las herramientas, la materia y los conocimientos que se tiene en tecnología en una empresa y como convergen para gestionar de manera óptima las tecnologías y explotar las sinergias de los elementos de la organización haciendo conciencia de las particularidades que se poseen. Por lo tanto, es importante conocer la manera en que las IT y las OT interactúan en una industria.

2.3. Convergencia de Integración IT/OT

El término convergencia IT/OT hace referencia a la integración de los sistemas de tecnologías de la información (IT), encargados del procesamiento y análisis de datos, con los sistemas de tecnología operacional (OT), responsables de supervisar y controlar procesos físicos en tiempo real. Esta convergencia permite una toma de decisiones más ágil, basada en información generada directamente desde el entorno de producción (Siddeswaran et al., 2017). En los últimos años, las organizaciones han incrementado su inversión en infraestructura IT con el propósito de optimizar la eficiencia operativa, independientemente del sector al que pertenezcan.

En este sentido, se entiende por empresa industrial digital aquella que ha logrado la conectividad entre sus departamentos de diseño, producción, logística, servicios y administración, como resultado de una integración efectiva entre IT y OT (Accenture, 2017).

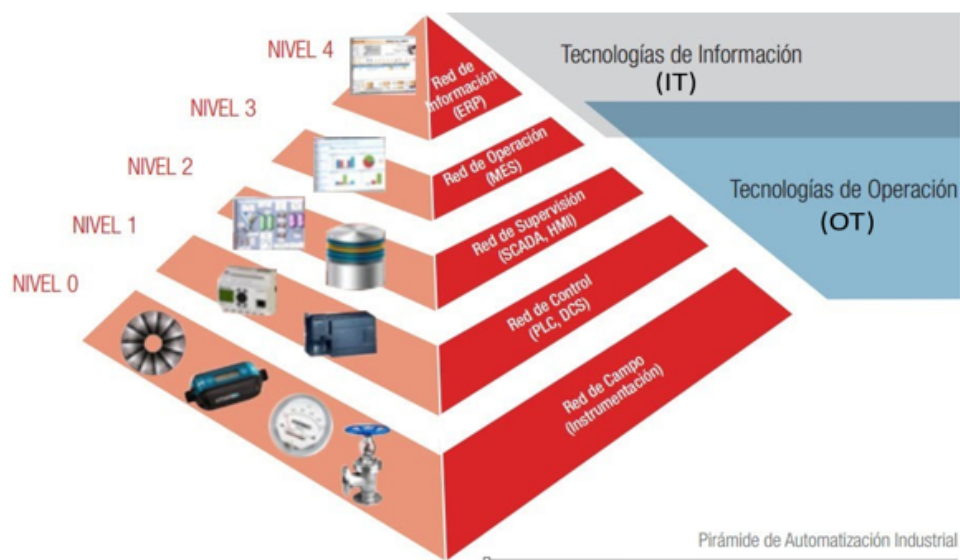
En relación con este enfoque, *Blevins et al. (2003)* presentaron un modelo organizacional estructurado estratégicamente para lograr la intercomunicación entre

todos los niveles del proceso productivo industrial, integrando personas, procesos, tecnología e información como una unidad operativa coordinada.

Para facilitar la comprensión de las tecnologías en una organización industrial, se utiliza el modelo conocido como pirámide de automatización, que clasifica jerárquicamente los niveles de un sistema de producción (Hernández & Díaz, 2020). La **Figura 1** ilustra este modelo, en el cual cada nivel agrupa los instrumentos y herramientas tecnológicas específicas para cada etapa del proceso, desde los dispositivos de campo hasta la gestión corporativa.

Figura 1111

Pirámide de los sistemas involucrados en la gestión empresarial.



Nota. Pirámide de Automatización Industrial que muestra los cinco niveles tecnológicos en un entorno industrial. Adaptado de *Ciberseguridad en la pirámide de automatización industrial* (p. 2), por Centro de Ciberseguridad Industrial, s.f.

Tomando como referencia el modelo de la pirámide de automatización industrial (Centro de Ciberseguridad Industrial, s.f.), a continuación se detalla los niveles y tecnologías que la conforman:

- **Nivel 0: Campo**

Este es el nivel más básico donde se ejecutan los procesos de producción. Al ser el nivel más bajo de la jerarquía, se conforma de maquinaria, sensores, actuadores y transductores que interactúan directamente con el proceso físico. La información que surge de este nivel es de vital importancia para que los niveles superiores puedan tomar decisiones de manera acertada (Centro de Ciberseguridad Industrial, s.f.).

- **Nivel 1: Control**

Este nivel se compone de controladores lógico-programables (PLC), tarjetas de control y computadoras industriales, cuya finalidad es constituir los elementos de mando sobre la maquinaria del Nivel 0. Dicho de otro modo, son los dispositivos que permiten la conexión y el control directo entre el nivel de campo y el de supervisión (Centro de Ciberseguridad Industrial, s.f.).

- **Nivel 2: Supervisión**

Este nivel recopila y centraliza la información generada en los niveles inferiores, como el estado de los procesos y las variables de control. Su función principal es supervisar integralmente el proceso productivo y facilitar la optimización de la operación (Centro de Ciberseguridad Industrial, s.f.). Algunos ejemplos de plataformas utilizadas en este nivel incluyen SIMATIC IT Production Suite, de Siemens, e Industrial IT System 800xA, de ABB, que integran funcionalidades de control y visualización en tiempo real (Siemens, s.f.; ABB, s.f.).

- **Nivel 3: Planificación**

En este punto se desempeñan las tareas de administración de la producción, utilizando la información global del proceso para analizar debilidades y fortalezas. Las herramientas de este nivel se enfocan en la optimización y la eficiencia de la ingeniería, siendo los sistemas de ejecución de manufactura (MES) el ejemplo más representativo (Centro de Ciberseguridad Industrial, s.f.).

- **Nivel 4: Gestión**

Este es el nivel superior de la pirámide, conformado por equipos de cómputo donde se administra toda la información de planeación y gestión de la organización (gestión comercial, financiera, de recursos humanos, etc.). Los sistemas de planificación de recursos empresariales (ERP) son característicos de este nivel, ya que permiten

integrar y automatizar las prácticas de negocio de la empresa (Centro de Ciberseguridad Industrial, s.f.).

Ahora que se ha logrado plasmar como se debe de desarrollar la integración de IT/OT en una organización industrial, debemos analizar cuáles son las ventajas y desventajas de dicha integración.

2.4. *Ventajas de la Integración IT/OT*

En el artículo escrito por Guerrero (2018), nos describe cuales son las mejoras que se presentan al unir las IT y OT, en diferentes ámbitos:

- (i) Energéticos: adaptar el consumo energético en aquellas horas que la energía es más económica y reducirlo en aquellas que es más cara.
- (ii) Medio ambiente: Los dispositivos inteligentes pueden medir datos meteorológicos, energéticos, etc., para que la planta pueda cumplir con los niveles de polución medioambientales establecidos por la ley y adaptar toda la producción a estos.
- (iii) Productivos: hace posible adaptar la demanda del cliente a la producción, para optimizar ésta y ahorrar costes.
- (iv) Control de calidad: permite fabricar con precisión y rapidez sin afectar el ritmo de producción.
- (v) Mantenimiento: con la posibilidad de proceder a un mantenimiento predictivo que reduzca las averías y nos permita diseñar mejor el mantenimiento para reducir las pérdidas que éste puede ocasionar. Así mismo, las mejoras también se incorporan en el proceso comercial y logístico y agilizarán el proceso de venta del producto.

2.5. *Desventajas de la Convergencia IT/OT*

Sin embargo, no todos son beneficios también existen inconvenientes derivados de la unión de estas dos tecnologías en las organizaciones.

El potencial de integración entre las tecnologías de la información (IT) y las tecnologías operacionales (OT) ofrece importantes beneficios en términos de eficiencia, automatización y análisis en tiempo real. No obstante, esta interconexión también introduce riesgos significativos, ya que expone las infraestructuras industriales a las mismas amenazas históricas que han afectado a los sistemas IT, como el malware, el ransomware y el espionaje digital. Estos riesgos se ven amplificados cuando las redes industriales no

fueron originalmente diseñadas para estar conectadas a Internet o a sistemas corporativos abiertos.

Uno de los ejemplos más preocupantes es el espionaje industrial, que ya no requiere acceso físico, sino que puede realizarse de forma remota mediante la interceptación de datos generados por dispositivos industriales conectados. Esta situación es posible gracias a la creciente convergencia entre sistemas OT e IT, lo que aumenta la superficie de ataque de las infraestructuras críticas (Roa Salinas, 2020).

2.6. Antecedentes de Ciberataques

Incluso como lo menciona Cárdenas et al. (2009) los sistemas automáticos representan un peligro cuando están conectados a las redes industriales en caso de ser saboteados ya que pueden afectar el funcionamiento de los servicios básicos trayendo consigo graves consecuencias.

Solo hace falta recordar eventos como el incidente de *Stuxnet* en 2010, un gusano descubierto por la compañía bielorrusa VirusBlokAda, que tenía como objetivo atacar el programa nuclear de Irán. Este ataque logró ralentizar el proceso de enriquecimiento de uranio en Natanz, lo que eventualmente retrasó la producción de armas nucleares (Falliere, Murchu, & Chien, 2011). Este es solo un ejemplo de muchos otros, como los eventos *Watering Hole*, *Industroyer*, y *BlackEnergy*, que tienen como objetivo atacar redes eléctricas, sistemas de control industrial, entre otros (Lipovsky, 2017).

2.7. Redes Industriales

Con base en los argumentos planteados anteriormente, se puede afirmar que el alcance, la evolución tecnológica y la rentabilidad de los sistemas electrónicos han transformado significativamente las actividades humanas, especialmente en el contexto de la automatización industrial. Esta transformación se observa desde la sustitución de mecanismos puramente mecánicos hasta la implementación de dispositivos digitales y microprocesadores que permiten un control más preciso, flexible y eficiente de los procesos industriales (Galloway & Hancke, 2013).

Según estos autores, la evolución del control industrial ha pasado de sistemas manuales y analógicos a redes de control industrial (ICN), en las que sensores, actuadores, controladores lógicos programables (PLC) y sistemas SCADA se integran para supervisar

y optimizar procesos en tiempo real. Estas redes permiten una interconectividad funcional que ha revolucionado la operación de plantas industriales.

Siguiendo con el trabajo de Galloway & Hancke (2013), junto con el desarrollo de dispositivos digitales se crearon nuevos dispositivos de control digital se requieren nuevos protocolos de comunicación que comúnmente se conocen como protocolos de bus de campo (*fieldbus*, en inglés). Como resultado de estos nuevos protocolos de comunicación se incorporaron finalmente las redes que emplean estándares Ethernet y que permiten la comunicación entre los equipos de los distintos niveles empresariales.

Las características principales de una red industrial y que la hacen diferente respecto a una red doméstica o convencional se presentan en la

·

Cuando se habla del diseño de una red industrial no se puede tomar una única opción ya que depende de muchos factores como la especialización de la industria por lo que se hacen uso de una gran variedad de protocolos de comunicación y por consiguiente de dispositivos de comunicación para los sensores y actuadores que deben de adaptarse para cumplir los requisitos necesarios para su gestión en tiempo real.

Uno de los principales factores que debe de tomar en cuenta en el monitoreo y control industrial, es la baja latencia ya que se requiere que los tiempos de transmisión de la información sean mínimos.

Otra de las consideraciones más relevantes al momento de diseñar una red industrial es que se deben contar con los proxy o traductores de las diferentes redes Ethernet y el bus de comunicación del nivel 0. Esto también va de la mano del desarrollo de las nuevas tecnologías y de la compatibilidad de estas con la ya existentes.

Tabla 2222*Características de una Red Industrial.*

Aspecto	Característica de la red industrial	Descripción
Principal función	Control de equipo físico	Para la gestión de los recursos empresariales
Campos de aplicación	Manufactura, procesamiento y distribución de servicios	Actualmente todos los sectores económicos se ven beneficiados de las redes industriales Las redes industriales pueden tener tres o más niveles de profundidad además pueden implementarse diferentes protocolos o medios físicos en cada nivel por lo que se necesitan implementar <i>gateways</i> que permitan la interconexión.
Arquitectura	Profunda, con jerarquías funcionales separadas con protocolo y estándares físicos	Al presentarse la interconexión de los equipos del nivel 0 con la red, un fallo en esta repercute severamente en las actividades y en el costo de producción.
Fallas	Alto	Como en cualquier red, es fundamental asegurar que las señales enviadas y transmitidas sean correctas, ya que un error en la transmisión podría ocasionar fallas en la producción. Esto está directamente relacionado con la gravedad de dichas fallas.
Fiabilidad requerida	Alto	Con la interconexión de los equipos y procesos es importante que todos los datos sean transmitidos, procesados y respondidos tan rápido como sea posible.
Tiempo de transferencia	250 μ s – 10 ms	Las redes industriales deben tener la capacidad de predecir o determinar en la medida de lo posible cuándo se recibirá una respuesta a un dato enviado previamente.
Determinista	Alto	En las redes, se busca manejar únicamente la información necesaria, generalmente en forma de medidas o valores digitales, utilizando pocos bytes o incluso un solo bit. Las transmisiones de datos suelen ser periódicas para enviar muestras, mientras que las transmisiones aperiódicas ocurren ante eventos que requieren activar o detener sistemas de alarma.
Composición de los datos	Paquetes de datos pequeños enviados de manera periódica o aperiódica	Esta característica hace referencia a que es importante registrar el momento exacto en el que ocurrió un evento y que fue registrado por las transmisiones, sobre todo las periódicas.
Consistencia temporal	Requerido	De manera simple debemos considerar a las redes industriales como redes de uso rudo ya que las condiciones físicas en las que son instaladas muchas veces no son las ideales.
Ambiente de operación	Hostil, condicionado. El ambiente a menudo presenta altos niveles de contaminantes como el polvo, así como altas temperaturas, ruidos y vibraciones.	

Nota: Basado en el trabajo de Galloway y Hancke (2013), quienes describen las características fundamentales de una red industrial.

2.8. Tipos de Información

La información que circula en una red industrial puede clasificarse en diferentes categorías, según su función y criticidad dentro del sistema. De acuerdo con Mackay (2004), se pueden distinguir tres tipos principales:

Información de Control: corresponde a los datos transmitidos entre instrumentos, sensores, actuadores y controladores. Esta información forma parte de los bucles de control y está relacionada directamente con el funcionamiento operativo de los procesos.

Información de diagnóstico: se refiere a los datos capturados por sensores u otros dispositivos, cuya finalidad es el monitoreo del estado de los equipos, sin generar necesariamente una acción de control inmediata. Este tipo de información es fundamental para el mantenimiento preventivo y la identificación temprana de fallos.

Información de seguridad: engloba los datos vinculados a funciones críticas, como paradas de emergencia o sistemas de enclavamiento. Esta información requiere redes altamente fiables, con baja latencia y capacidad de respuesta en tiempo real, dado que puede tener implicaciones directas sobre la integridad de las personas, los equipos y el proceso productivo.

2.9. Componentes de Una Red Industrial

Una red industrial se conforma de diversos componentes y aplicaciones donde su configuración es adaptable de acuerdo con las metas planteadas, generalmente estas metas son de supervisión y control del proceso que resultan en acciones de cambio de procesos.

Para estas tareas de control existen diferentes elementos como son los controladores lógicos programables PLC's (por sus siglas en inglés Programmable Logic Controllers); los sistemas de control distribuido DCS (por sus siglas en inglés Distributed Control Systems); los sistemas instrumentados de seguridad SIS (por sus siglas en inglés Safety Instrumented Systems).

Por otro lado, las tareas de supervisión en entornos industriales son realizadas mediante *interfaces hombre-máquina (HMI)* y sistemas *SCADA (Supervisory Control and Data Acquisition)*, los cuales permiten visualizar, monitorear y controlar procesos en tiempo real desde estaciones centralizadas. Estas herramientas constituyen elementos fundamentales en la infraestructura de automatización industrial, al facilitar la interacción entre el operador humano y los sistemas de control (Şahin, 2016) . En cuanto a la organización

de estos sistemas, las arquitecturas industriales tradicionalmente adoptan una estructura jerárquica de tipo piramidal, que permite segmentar responsabilidades entre los diferentes niveles de operación. Entre los modelos de referencia más ampliamente utilizados destacan la *Purdue Enterprise Reference Architecture (PERA)*, que define una jerarquía funcional para la integración de sistemas IT y OT (ISA, 1992), y el *Reference Architecture Model Industry 4.0 (RAMI 4.0)*, desarrollado en Alemania para estandarizar la digitalización de la industria bajo los principios de la Industria 4.0 (Plattform Industrie 4.0, 2015).

A continuación, se definen y describen los orígenes, características, fundamentos y aplicaciones de estos componentes de la red eléctrica.

Elementos Para Tareas de Control.

PLC

Los Controladores Lógicos Programables (PLC, por sus siglas en inglés) son dispositivos electrónicos diseñados para controlar procesos industriales mediante la gestión de múltiples entradas conectadas a sensores, y salidas que envían señales de control a actuadores. Su función principal es recibir información del entorno, procesarla y ejecutar acciones predeterminadas según una lógica programada (Bolton, 2015).

Estos dispositivos surgieron a finales de los años 60, como una alternativa tecnológica eficiente frente a los sistemas de control cableados basados en relés, temporizadores y contactores. Su aparición permitió una mayor flexibilidad, facilidad de mantenimiento y reducción de tiempos de reconfiguración en entornos industriales (Parr, 2013).

Un PLC típico está compuesto por tres elementos fundamentales: entradas, salidas y una unidad central de procesamiento (CPU). Las entradas reciben señales desde los sensores del sistema; la CPU interpreta esta información y ejecuta una lógica de control predefinida; y las salidas activan los dispositivos finales como válvulas, motores o alarmas. Este ciclo de operación puede programarse utilizando lenguajes como el diagrama de escalera (Ladder Diagram), almacenado en la memoria del dispositivo (Bolton, 2015). Además, los PLCs modernos integran fuentes de alimentación, módulos de comunicación y funciones avanzadas como control PID (proporcional–integral–derivativo), ampliando así su capacidad operativa.

En la actualidad, los PLC son ampliamente utilizados por su versatilidad, robustez y durabilidad, y por su capacidad de procesar tanto señales digitales como analógicas, lo que los hace aptos para tareas complejas en entornos industriales exigentes.

DCS.

Estos dispositivos de control son aplicados por lo general en sistemas dinámicos y que tienen la característica que los elementos que generan la información no se encuentran en un punto sino distribuidos como su nombre lo dice a lo largo del sistema o subsistemas de producción los cuales son controlados por uno o más controladores. Las interconexiones se dan mediante las redes de comunicación y de monitoreo puestos en este.

Los DCS aparecieron en la década de los 70's pensados inicialmente como el relevo de los tableros de instrumentación electrónica. Para mediados de la década de los 80's estos componentes podían sustituir totalmente las redes de control analógico de operaciones en una planta y a partir de este momento han sido la base para el surgimiento de microprocesadores.

Tradicionalmente los DCS se componen de cinco subsistemas: 1) estación de trabajo de operaciones, 2) subsistema de control, 3) subsistema de recolección de datos, 4) subsistemas computacionales de procesos y 5) redes de comunicación.

Este tipo de controladores permite tener en un solo dispositivo múltiples computadoras que pueden encargarse un conjunto de controladores PID a lo largo del sistema y enlazados por información compartida entre ellos y las estaciones de trabajo de operaciones.

Cuando se habla de los sistemas de control distribuido (DCS, por sus siglas en inglés), es indispensable abordar el concepto de redundancia, una característica fundamental en los sistemas de automatización industrial de alta disponibilidad. La redundancia en DCS se refiere a la incorporación de elementos de respaldo, como unidades de procesamiento y módulos de adquisición de señales digitales duplicados, que entran automáticamente en funcionamiento en caso de fallo de los componentes principales, garantizando así la continuidad de las operaciones críticas sin interrupciones (Bialek, 2016).

Este diseño con “spares” o módulos de repuesto es común en industrias que no pueden permitirse tiempos de inactividad, como el sector energético, petroquímico o

farmacéutico, donde los sistemas redundantes permiten mantener las funciones de control incluso durante fallas del hardware principal.

SIS.

Estos sistemas, conocidos como Sistemas Instrumentados de Seguridad (SIS), están diseñados para garantizar la seguridad operacional de los procesos industriales, llevando el sistema a un estado seguro predefinido en caso de presentarse una condición peligrosa o anómala. De acuerdo con la norma IEC 61511, un SIS se compone de elementos tales como sensores, controladores lógicos programables de seguridad (Safety PLC) y actuadores, que operan en conjunto para ejecutar las denominadas Funciones Instrumentadas de Seguridad (SIF).

Cada SIF está asociada a un Nivel de Integridad de Seguridad (SIL, por sus siglas en inglés), el cual define la confiabilidad requerida para que la función reduzca el riesgo asociado a niveles tolerables (International Electrotechnical Commission [IEC], 2016).

Elementos para tareas de supervisión.

HMI.

La interfaz hombre-máquina (HMI) es un componente fundamental para la supervisión en la automatización industrial, ya que facilita la interacción directa entre el operario y el proceso. Mediante paneles que integran visualizaciones gráficas, indicadores y controles, el HMI muestra en tiempo real el estado de la producción. Esta información es crucial para que los operadores y supervisores puedan tomar decisiones, coordinar acciones y controlar los subprocesos de la planta de forma segura y eficiente.

SCADA.

Un sistema SCADA (Supervisory Control and Data Acquisition) puede entenderse como la capa intermedia entre los sistemas de control en tiempo real, como los PLC o DCS, y los sistemas de gestión de producción, como los MES (Manufacturing Execution Systems). Los SCADA son aplicaciones de software instaladas en estaciones de trabajo o servidores, que permiten a los usuarios supervisar, controlar y visualizar remotamente los procesos industriales, actuando sobre los equipos conectados a través de sistemas de control (Stouffer et al., 2015).

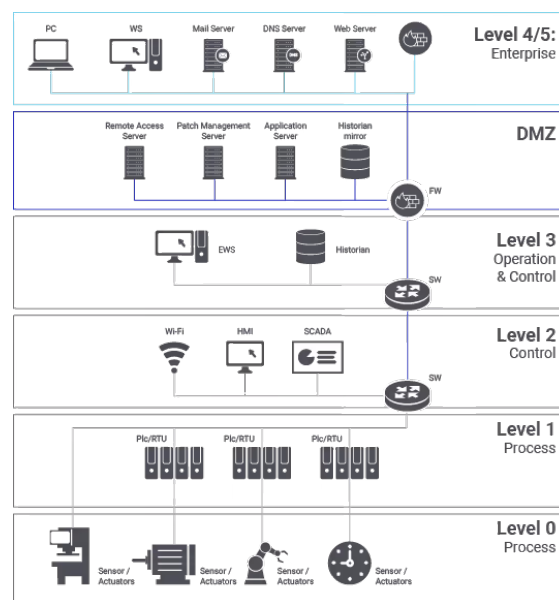
La información recopilada por un sistema SCADA puede ser utilizada para lecturas en tiempo real, registro histórico de datos, generación de alarmas, y visualización mediante gráficas de tendencia. Asimismo, estos sistemas permiten la gestión de usuarios, controlando el acceso y registrando las acciones realizadas por cada operario. Algunos SCADA permiten exportar la información a formatos como Excel o bases de datos, facilitando su análisis posterior y su integración con otros sistemas empresariales.

2.10. *Arquitectura de Redes Industriales*

La Arquitectura de Referencia Empresarial Purdue (PERA, por sus siglas en inglés) fue desarrollada en la Universidad de Purdue, EE. UU. Según Williams (1994), PERA es un modelo arquitectónico que define todos los componentes de una empresa a lo largo de su ciclo de vida, desde su concepto inicial hasta su disolución. Tal como se muestra en la Figura 2, esta arquitectura establece que una empresa se compone de tres elementos principales: las instalaciones de producción, la organización (personas) y la arquitectura de la información.

Figura 2

Arquitectura de referencia empresarial Purdue



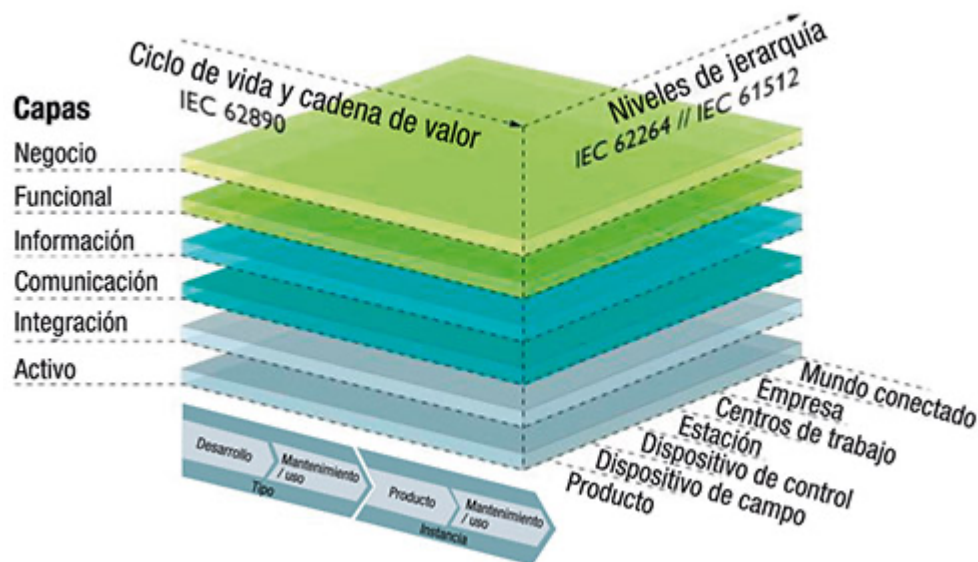
Nota. Modelo jerárquico de la automatización industrial basado en el Modelo de Referencia de Purdue. Adaptado de Converged Plantwide Ethernet (CPwE) Design and Implementation Guide (p. 56), por Rockwell Automation y Cisco Systems, 2011.

El modelo de referencia de arquitectura de la industria 4.0 (RAMI 4.0 por sus siglas en inglés Reference Architecture Model Industry 4.0), creado por Platform Industrie 4.0 como respuesta al nuevo modelo de la industria conectada ya que opta por ser un modelo donde las jerarquías no están estrictamente definidas al contrario todos los dispositivos forman parte de la red y deben entablar comunicaciones libres dentro de la misma. Como se muestra en la figura 3, la organización de la empresa se realiza desde tres enfoques:

- *Jerarquía*: la jerarquía de los elementos de la empresa se distribuye no hacia las actividades a las que están dirigidas sino a su flexibilidad por esto se distribuyen en tres grupos: El mundo real, El Producto inteligente y la Planta Inteligente.
- *Ciclo de vida del producto*: Se consideran las fases implicadas en la creación de producto final, así como el servicio de postventa.
- *Capas*: Son los todos los aspectos que se deben de considerar cuando se fabrica el producto, como lo son los activos, integración, comunicación, información funcional y negocio.

Figura 3

Modelo de arquitectura de referencia de la Industria 4.0



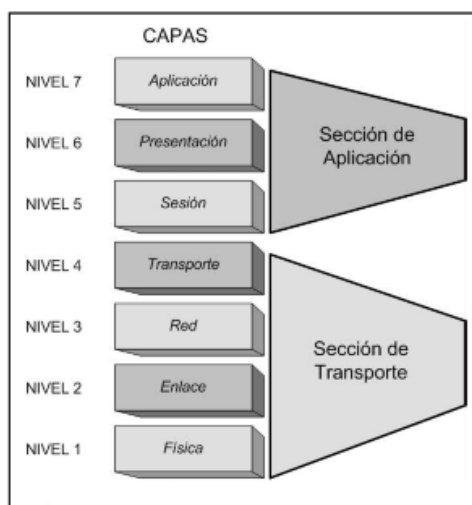
Nota. Modelo Arquitectónico de Referencia para la Industria 4.0 (RAMI 4.0), que estructura los conceptos de la Industria 4.0 a través de tres dimensiones. Adaptado de Reference Architectural Model Industrie 4.0 (RAMI 4.0)

Protocolos de Comunicación.

Modelo OSI de ISO.

Este modelo fue propuesto por la Organización Internacional de Estándares (ISO, por sus siglas en inglés), en la que se plantea una manera de intercomunicar sistemas abiertos y garantizar la operatividad en conjunto de los mismo. Este modelo se forma por 7 capas como se puede ver en la Figura 4. Las funciones que cada capa tiene en este modelo se describen a continuación:

- **Nivel 1 (Capa física):** En este punto se establecen las condiciones de operación de los niveles de corriente, voltaje modulación y frecuencia que permitan entablar una comunicación efectiva por el medio de transmisión.
- **Nivel 2 (Capa Enlace):** se establece la manera de acceder al medio de transmisión de datos, el direccionamiento de las estaciones y el tiempo de transmisión. La función principal se estructura en tramas para la transmisión y codificación de la información con el fin evitar errores.
- **Nivel 3 (Capa de Red):** se encarga de la administración de la información por las redes de comunicación para controlar el tráfico en las subredes.
- **Nivel 4 (Capa de Transporte):** Se encarga del aseguramiento de la inflación durante el transporte y que los datos sean coherentes, mediante la preparación de los extremos de las líneas para el intercambio de datos.
- **Nivel 5 (Capa de Sesión):** Su principal función es controlar las comunicaciones desde la sincronización hasta la supervisión del uso que cada usuario hace de la red.
- **Nivel 6 (Capa de presentación):** Es la transcripción de los datos codificados en un lenguaje común.
- **Nivel 7 (Capa de Aplicación):** Es la capa final en la que el usuario puede interactuar con el sistema por medio de una interfaz de usuario.

Figura 4*Siete niveles del Modelo OSI*

Nota. Las siete capas del Modelo de Interconexión de Sistemas Abiertos (OSI). Adaptado de Redes de computadoras (5.^a ed., p. 35), por A. S. Tanenbaum y D. J. Wetherall, 2012, Pearson.

Modelo TCP/IP.

Este modelo es implementado en las redes de manera más común que el modelo OSI. Este modelo considera cuatro niveles o capas, como se puede observar en la Figura. Pese a esta diferencia de capas el modelo TCP/IP realiza las mismas acciones a continuación describiremos cada una de las capas que lo conforman:

Capa de acceso a la red: las tareas que se desempeñan están relacionadas con la capa física y de enlace del modelo OSI. En este punto se desarrollan las rutinas que permiten el acceso físico a la red.

Capa de Internet: esta capa tiene la función de permitir que el equipo transfiera los paquetes de datos desde cualquier red y a cualquier red.

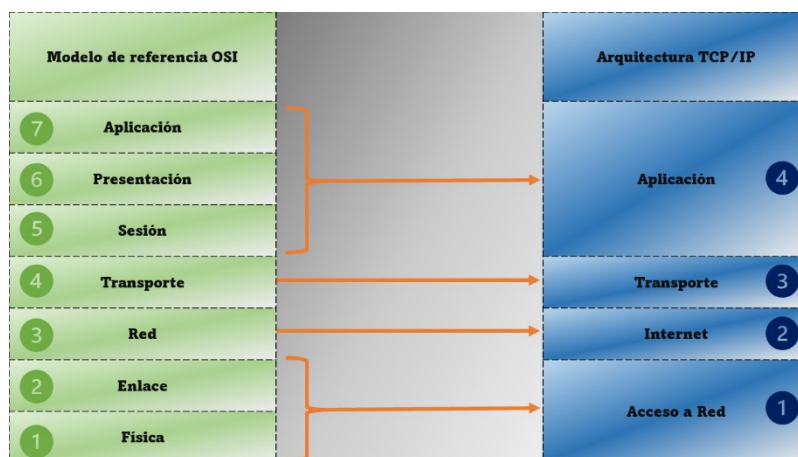
Capa de transporte: esta capa está diseñada para establecer la comunicación entre los equipos que conforman la red industrial. En este modelo se emplean dos protocolos: el TCP, el cual se encarga de una conexión, libre de errores, así como transferencia de bloques de datos; adicionalmente, se encarga de administrar el tráfico de datos; y el UDP, el cual es un protocolo que trabaja sin conexión enviando paquetes de datos simples donde la pérdida de algunos datos no supone relevancia en el sistema.

Capa de aplicación: en esta capa se presentan todos los de alto nivel para la incorporación de elementos de la interfaz de usuario.

En la Figura 5 se muestran las equivalencias entre los modelos OSI y TCP/IP.

Figura 5

Equivalencia del protocolo TCP/IP vs modelo OSI



Nota. Comparación de las capas del modelo de referencia OSI con la arquitectura del protocolo TCP/IP. Adaptado de Redes de computadoras (5.^a ed., p. 40), por A. S. Tanenbaum y D. J. Wetherall, 2012, Pearson

Protocolos de Comunicación Industrial.

En las redes industriales es de suma importancia la comunicación y se considera que la disponibilidad de la información debe ser una prioridad mayor que la integridad o confidencialidad de la información, ya que si no se tiene disponible la información puede implicar pérdidas económicas y fallos en el proceso.

De acuerdo con Sasir Alejandro (2021) define un protocolo industrial como un conjunto de normas que le permiten a dos identidades que se encuentren en un mismo sistema establecer una comunicación, con el objetivo de pasar información a través de diversas variables.

Siguiendo las comunicaciones para la automatización de procesos tienen en el mercado una gran variedad de protocolos que se pueden implementar dependiendo del nivel o capa de la empresa en el que se planea usar para maximizar las capacidades.

Para los procesos de automatización de procesos se utilizan protocolos de bus de campo, por ejemplo, Modbus RTU, PROFIBUS o DeviceNET (CIP).

Por otro lado, los protocolos empleados en los niveles de control industrial suelen ser extensiones de las tradicionales redes de bus de campo, adaptadas para operar sobre Ethernet TCP/IP. Ejemplos de estos protocolos son MODBUS TCP, PROFINET y Ethernet/IP (CIP), los cuales permiten una mayor interoperabilidad, escalabilidad y velocidad de comunicación entre dispositivos industriales (James, 2010).

En el sector industrial, además de los protocolos abiertos, es frecuente el uso de tecnologías propietarias que pueden generar desafíos de compatibilidad e integración entre sistemas. Para resolver estos problemas de interoperabilidad, se han desarrollado estándares de comunicación como *OLE for Process Control (OPC)*.

La función principal de OPC es ofrecer una interfaz común que permita el intercambio de datos entre los dispositivos de control en la planta y el software de supervisión. Las primeras implementaciones de este estándar se basaban en tecnologías más antiguas. Con el tiempo, el protocolo evolucionó hacia la *Arquitectura Unificada (OPC UA)*, un modelo más moderno, seguro y multiplataforma que se ha consolidado como una tecnología clave para la comunicación en el contexto de la Industria 4.0.

Como se pudo observar anteriormente existe una larga lista de protocolos de comunicación de aplicación industrial, aunque, no entraremos de lleno en ellos, realizaremos una revisión de las características más relevantes:

Protocolos de Buses de Campo.

Los buses de campo que son los sistemas de transmisión de los paquetes de datos permitiendo la simplificación de la instalación y operación de máquinas y equipos de nivel industrial empleados en la producción.

La principal actividad de los buses de campo es reemplazar las conexiones punto a punto que existe entre los elementos de la capa física y los existentes en la capa de control. Usualmente estos buses son redes digitales con una corriente de 4–20mA, bidireccionales, multipunto con una señal montada en un arreglo serie.

Protocolo HART.

Este protocolo se emplea para la comunicación de instrumentación inteligente la cual se emite una señal a 1.2 KHz para enviar un “1” y una señal de 2.2 KHz para enviar “0” ambas señales coexisten en la misma línea de transmisión esto se puede debido primeramente a que la corriente es cero para la señal codificada y al estar superpuestas a

una señal de lazo de 10 Hz con una corriente mayor lo que se resuelve con un filtrado de la señal.

PROFIBUS (Process Field Bus).

Es un protocolo estándar europeo (EN50170) abierto, que tiene como característica principal su alta velocidad para transmisión de datos y su larga vida útil en ambientes hostiles y de uso universal. Cuenta con tres versiones FMS (*Fieldbus Message Specification*) diseñado para tareas de comunicación de control de propósito general, DP (*Distributed Peripherals*) protocolo para establecer comunicaciones de alta velocidad y a bajo costo diseñado especialmente para sistemas automáticos de control y E/S instalaciones en la capa física y PA (Process Automation) diseñado para la automatización de procesos permitiendo la conexión de sensores y actuadores a un bus común, está basado en el estándar IEC 1158-2. A este último se le considera un extensión de la versión DP.

Foundation Fieldbus (FF).

Es un protocolo que está diseñado específicamente para ser implementado en actividad de control distribuido, donde se maneja una alta cantidad de información donde existen aplicaciones de lazos complejos de control y automatización de procesos continuos. Es un subconjunto de los estándares IEC61158 e ISA s50.02.

PROFINET

Es uno de los considerados protocolos líderes debido a que permite establecer procesos automáticos integrados de principio a fin, así como crear interfases de red, además de tener una alta calidad, flexibilidad, eficiencia y precisión de los procesos de automáticos.

MODBUS.

Es uno de los protocolos de comunicación más antiguos diseñado para enviar y recibir paquetes de los sistemas SCADA con control centralizado a partir de Unidades Terminales Remotas (RTU por sus siglas en inglés). Los datos comunicados provienen de los controladores y los sensores a través de interfaces de la capa física como es el puerto RS-232. RS-422, RS-485, además se pueden intercambiar en modo RTU o ASCII.

DeviceNET.

Es una red diseñada para conectar tanto sensores simples (fotoeléctricos, magnéticos, pulsadores) como dispositivos complejos (PLC's, HMI, CPU, etc.) para su instalación se emplea fibra óptica o par trenzados además de utilizar conector específico. Basada en el estándar ISO11898.

AS-i.

Este protocolo de comunicación está diseñado para los procesos de control implementados en los niveles más bajos ya que es un protocolo robusto y flexible que se basada en el estándar IEC 62026-2, lo que permite una alta compatibilidad con diversos controladores además se puede implementar como alimentador económico para PROFIBUS y PROFINET.

IO-Link.

Aunque por algunos autores no es considerado un bus de campo sino una conexión punto a punto entre los sensores y actuadores y un dispositivo maestro que puede ser un PLC. Es una comunicación bidireccional con una conexión en serie que permite la transmisión de datos estandarizados.

Open Platform Communications – Unified Architecture (OPC UA).

Este protocolo está diseñado para la comunicación entre máquinas (M2M) a partir de una comunicación vertical, conjuntando las ventajas ofrecidas por los protocolos OPC UA y las de PROFINET.

2.11. Honeypot

Los honeypots son sistemas de seguridad destinados a engañar a los ciberdelincuentes, brindando protección a las infraestructuras informáticas y sistemas legítimos al desviar los ataques. Sirven también como fuentes para recopilar detalles valiosos sobre actividades malintencionadas, incluyendo datos como el momento del ataque, la dirección IP del atacante, el sistema operativo, y las técnicas empleadas para el ataque, como el uso de fuerza bruta y exploits.

En cuanto a su clasificación, los honeypots pueden dividirse según varios criterios: nivel de interacción, finalidad, e implementación. Respecto al nivel de interacción, existen honeypots de baja, media y alta interacción, que varían en la

profundidad del engaño proporcionado. Los de baja interacción ofrecen servicios simulados como SSH, FTP, Telnet, HTTP, etc., y recogen información del ataque. Los de media interacción brindan mayor funcionalidad en los servicios emulados, mientras que los de alta interacción poseen una completa operatividad, permitiendo una interacción plena del atacante, a costa de un mayor riesgo si no son debidamente controlados.

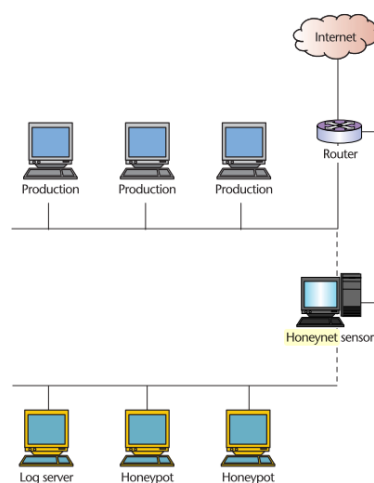
Con relación a su finalidad, encontramos honeypots de investigación, empleados por instituciones académicas para descubrir nuevas técnicas de ataque y entender las motivaciones del atacante, y honeypots de producción, que son utilizados en empresas para desviar y analizar los ataques.

En términos de implementación, distinguimos entre honeypots virtuales, que son sistemas emulados mediante software de virtualización y alojados en un host físico, y honeypots físicos, que implican un hardware específico, proporcionando un mayor grado de realismo, pero a un costo más alto.

Además, cabe destacar las honeynets, redes de honeypots que emulan una infraestructura completa para un análisis más extenso. En el presente trabajo, se propone la construcción de una honeynet de investigación, implementada de forma virtual y destinada a integrarse en una red industrial IoT. Para ello, se utilizarán honeypots de baja y media-alta interacción para simular servicios, recoger datos de interacción y muestras maliciosas. A modo de ejemplo gráfico, la Figura 6 se muestra un esquema de honeynet

Figura 6

Esquema Honeynet.



Nota. Arquitectura de una Honeynet de segunda generación. Adaptado de "The Honeynet Project: Trapping the Hackers", por L. Spitzner, 2003, IEEE Security & Privacy, 1(2).

2.12. Lenguaje de Programación Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general que fue creado por Guido van Rossum y lanzado por primera vez en 1991. Su filosofía de diseño enfatiza la legibilidad del código mediante el uso de una sintaxis clara y concisa que permite a los programadores expresar conceptos en menos líneas de código en comparación con otros lenguajes como C++ o Java (Van Rossum, 1991).

Uno de los principales atractivos de Python es su enfoque en la simplicidad y productividad, lo que lo convierte en una herramienta popular tanto para desarrolladores principiantes como avanzados. Es un lenguaje multiparadigma, lo que significa que admite múltiples estilos de programación, incluidos el procedural, orientado a objetos y funcional. Además, es un lenguaje interpretado, lo que permite ejecutar código línea por línea, facilitando la depuración y el desarrollo rápido de prototipos.

Python ha crecido enormemente en popularidad, principalmente debido a su extensa biblioteca estándar y su ecosistema de bibliotecas externas (PyPI), que ofrecen soluciones para una amplia gama de aplicaciones como desarrollo web, automatización de tareas, análisis de datos, inteligencia artificial, ciberseguridad, y más. Entre las bibliotecas más notables están *NumPy* y *Pandas* (para ciencia de datos), *Scikit-learn* (para aprendizaje automático), *TensorFlow* (para redes neuronales) y *Paramiko* (para comunicaciones SSH).

La comunidad activa de Python también ha permitido la creación de entornos de desarrollo robustos y plataformas educativas, haciendo que Python sea una de las principales opciones en campos como la ciberseguridad, desarrollo de software, automatización y análisis de datos.

2.13. ELK (Elasticsearch, Logstash y Kibana)

ELK es un conjunto de herramientas de software de código abierto que incluye Elasticsearch, Logstash y Kibana. Estas tres herramientas se utilizan comúnmente en combinación para la gestión de datos, análisis y visualización en tiempo real, formando una arquitectura de análisis de registros centralizada, muy utilizada en entornos de grandes volúmenes de datos y ciberseguridad.

- *Elasticsearch*: Es un motor de búsqueda y análisis distribuido basado en Apache Lucene. Es la herramienta que indexa, almacena y permite realizar búsquedas sobre grandes volúmenes de datos en tiempo real. Elasticsearch

es altamente escalable, lo que lo convierte en una opción ideal para empresas que necesitan gestionar grandes cantidades de datos rápidamente y extraer información de valor a través de consultas y filtros avanzados. Es ampliamente utilizado para la gestión de logs y monitoreo en tiempo real de sistemas de IT, incluidos sistemas SCADA en entornos industriales (Clinton Gormley & Zachary Tong, 2015).

- *Logstash*: Es una herramienta de procesamiento de datos que ingesta y transforma datos en tiempo real desde múltiples fuentes, tales como logs, eventos y métricas. Logstash permite recopilar, filtrar y transformar estos datos para posteriormente enviarlos a un destino como Elasticsearch. Su flexibilidad en la gestión de formatos de datos y su capacidad para manipular y filtrar información en tiempo real son claves en la construcción de pipelines de datos robustos en arquitecturas distribuidas.
- *Kibana*: Es la herramienta de visualización que se utiliza junto con Elasticsearch para crear gráficos, tablas y dashboards interactivos. Kibana permite a los usuarios visualizar y explorar los datos almacenados en Elasticsearch mediante una interfaz gráfica amigable. Esto es especialmente útil en el análisis de logs, la ciberseguridad y el monitoreo de infraestructuras en tiempo real, permitiendo a los administradores de sistemas tomar decisiones informadas con base en los datos.

En conjunto, el stack ELK es una solución poderosa para la gestión, análisis y visualización de grandes volúmenes de datos en tiempo real, y es ampliamente utilizado en la administración de infraestructuras críticas, incluyendo sistemas industriales SCADA. Su capacidad para integrar, analizar y presentar datos de manera efectiva lo ha convertido en un estándar en entornos empresariales para la toma de decisiones basada en datos.

2.14. Ciberseguridad

La ciberseguridad constituye una disciplina fundamental dentro del ámbito de la seguridad de la información, orientada a proteger los sistemas, redes, dispositivos y datos frente a accesos no autorizados, alteraciones, pérdidas o interrupciones. Esta área se

encuentra en constante evolución, impulsada por el avance tecnológico y el incremento de amenazas en entornos digitales e industriales (Whitman & Mattord, 2021).

Para estructurar sus objetivos y mecanismos de protección, la ciberseguridad se basa en una serie de principios fundamentales que garantizan el resguardo y uso adecuado de la información, tanto en tránsito como en almacenamiento. Entre estos principios se encuentran la disponibilidad, accesibilidad, integridad, confidencialidad, responsabilidad y confiabilidad, los cuales se describen a continuación.

- 1 *Disponibilidad:* Este principio hace referencia a la operabilidad continua del sistema, es decir, la información y servicios que proporciona la organización deben estar disponibles para satisfacer las necesidades. Ya que si el sistema no está disponible puede causar pérdidas.
- 2 *Accesibilidad:* Se refiere a que los datos o sistemas requieren de autorización para su uso. Este principio existe para garantizar que un sistema trabaje sin retrasos y evitar que el sistema llegara a negar el acceso a usuarios con autorización.
- 3 *Integridad:* Este principio se encarga de garantizar de que los datos del sistema no sean alterados por personas no autorizadas y así evitar la pérdida de la información. La integridad a su vez tiene dos vertientes; la integridad de datos, que indica que los datos no han sido modificados de manera no autorizada ya sea en durante el almacenamiento, procesamiento o transferencia; y, por otro lado, la integridad del sistema, que es la cualidad de un sistema para realizar funciones solicitadas libres de manipulación.
- 4 *Confidencialidad de datos e información:* El objetivo de este requisito es que la información existente en el sistema se mantenga privada y que no esté disponible para usuarios no autorizados. Este principio debe ser aplicados en los datos durante su almacenamiento, procesamiento y transferencia. Además, este rubro es el de mayor relevancia en algunos sistemas y tipos de datos como son los autenticadores como es en el caso de este trabajo.
- 5 *Responsabilidad:* En este principio también es conocido como no repudio, esto es que cuando el sistema recibe un mensaje, éste debe de identificar sin errores al remitente con el fin de que este asuma las responsabilidades derivadas por lo escrito en el mismo.

6 *Confiabilidad o aseguramiento*: Este principio garantiza que todos los objetivos anteriores se cumplen de manera óptima y que en su conjunto aseguran que el sistema queda protegido al igual que la información.

Estos aspectos buscan trazar las líneas de acción para combatir los ataques remotos que puede ser objeto un sistema como resultado de diferentes acciones impredecibles.

Ante el impacto negativo que los ciberataques pueden generar en las industrias, resulta fundamental definir una estrategia de ciberseguridad efectiva. En este sentido, el Instituto Nacional de Ciberseguridad de España (INCIBE) ha desarrollado un *Decálogo de Ciberseguridad para empresas*, concebido como una guía para promover el uso seguro de la tecnología y facilitar la adaptación de las organizaciones al entorno digital (INCIBE, 2017).

Dicho decálogo constituye una base útil para implementar políticas preventivas y de concientización, y propone las siguientes recomendaciones:

1. *Diagnóstico de Riesgos*: Identifica las vulnerabilidades de tu empresa para saber qué proteger.
2. *Plan de Seguridad*: Crea una estrategia clara con prioridades, responsables y recursos definidos.
3. *Control de Acceso*: Limita y registra quién, cómo y cuándo se accede a la información.
4. *Copias de Seguridad*: Realiza respaldos periódicos de tus datos para poder recuperarlos ante cualquier incidente.
5. *Antivirus/Antimalware*: Instala software de seguridad en todos los equipos como defensa básica.
6. *Sistemas Actualizados*: Aplica siempre las últimas actualizaciones para cerrar brechas de seguridad.
7. *Red Segura*: Utiliza firewalls y otras herramientas para proteger tu red de intrusiones.
8. *Cifrado y Borrado Seguro*: Cifra la información importante y elimínala de forma permanente cuando ya no la necesitas.
9. *Plan de Continuidad*: Ten un plan de acción para que tu negocio siga operando después de un ciberataque.

-
10. *Formación de Empleados*: Capacita a tu personal; son tu primera y más importante línea de defensa.

Ciberataques Industriales.

Con los principios de la seguridad debidamente integrados en un sistema debería de ser suficiente para lograr proteger la información sin embargo no se puede garantizar una seguridad total por lo que es importante mantener la guardia arriba en todo momento.

Históricamente se han presentado ciberataques sobre infraestructuras industriales críticas con consecuencias graves como los que mencionaremos a continuación:

STUXNET.

El caso Stuxnet marcó un hito en la historia de la ciberseguridad industrial al ser reconocido como el primer *malware* específicamente diseñado para atacar infraestructuras críticas físicas. Su objetivo principal fue interferir en las operaciones de la planta nuclear de *Natanz, en Irán*, afectando de forma directa los procesos de enriquecimiento de uranio. Aunque su descubrimiento se produjo en el año 2010, los análisis forenses revelaron que el malware había estado activo desde al menos 2007, provocando daños que, durante años, fueron interpretados como fallas técnicas comunes en las centrifugadoras industriales (Langner, 2011).

Los primeros indicios surgieron durante una inspección de la *Agencia Internacional de Energía Atómica*, cuando se detectaron anomalías inusuales en el funcionamiento de los equipos de enriquecimiento. La investigación posterior reveló que Stuxnet había sido introducido posiblemente mediante un *dispositivo USB contaminado*, propagándose en redes que no estaban conectadas a internet. Una vez dentro, el malware se dirigía selectivamente a dispositivos que utilizaran *PLC Siemens S7*, ignorando otros sistemas que no tuvieran relación directa con el control de las centrifugadoras (Falliere, Murchu & Chien, 2011).

El malware *Stuxnet* representó un punto de inflexión en la ciberseguridad industrial. Su complejo diseño le permitía infiltrarse en sistemas de control y reprogramar su lógica interna para inducir fallos físicos en la maquinaria, como la alteración de la velocidad de rotación de las centrifugadoras.

Una de sus características más sofisticadas era su capacidad para ocultar el sabotaje, ya que suprimía las alarmas del sistema y falseaba los datos de los sensores. De

esta manera, los operadores no detectaban ninguna anomalía mientras el daño se producía. Este ataque demostró la viabilidad de las ciberarmas diseñadas para causar un impacto físico directo, combinando la inteligencia digital con el sabotaje en el mundo real.

DragonFly / Havex.

El grupo de ciber amenazas persistentes avanzadas conocido como *DragonFly*, también denominado *Energetic Bear*, ha sido identificado como uno de los ataques más sofisticados contra infraestructuras industriales desde el caso Stuxnet. Esta APT (Advanced Persistent Threat) ha estado activa desde al menos 2013, y su principal objetivo han sido organizaciones del sector energético. Sin embargo, con el paso del tiempo, se han documentado intrusiones en otros sectores como el *alimentario, farmacéutico, de tratamiento de aguas y manufactura industrial* (Symantec, 2017; Vukalović & Delija, 2015).

El propósito de esta campaña maliciosa es comprometer los sistemas de control industrial mediante la recolección de credenciales, el acceso remoto a equipos de red, la interrupción de servicios y la manipulación de entornos combinados IT/OT. Las acciones detectadas incluyen desde la recolección de datos y la escalada de privilegios, hasta la ejecución de código arbitrario y la denegación de servicio en equipos críticos.

Los vectores de ataque utilizados por *DragonFly* combinan múltiples técnicas, entre ellas el *spear phishing mediante correos electrónicos dirigidos, la infección de sitios web legítimos (watering hole attacks)*, y la utilización de software comprometido distribuido desde páginas oficiales. Dentro de estas campañas, se identificó el uso de al menos tres herramientas de acceso remoto (RAT): *Havex, Karagany y Sysmain* (FireEye, 2014).

Uno de los casos más documentados de *spear phishing* consistió en correos dirigidos a ejecutivos del sector eléctrico, que contenían archivos adjuntos con extensión *.XDP* (Adobe XML Data Package). Al ser abiertos, estos archivos activaban la ejecución de *Havex*, una RAT que aprovechaba vulnerabilidades como la *CVE-2011-0611*, relacionada con archivos *PDF/SWF*, para instalarse en el sistema sin ser detectado (US-CERT, 2014).

Los ataques del tipo *watering hole* también fueron altamente efectivos. Se implementaban mediante técnicas de *clickjacking* utilizando iFrames ocultos en Java y HTML insertados en sitios web legítimos de proveedores industriales. Algunas

compañías afectadas por esta técnica fueron *Mesa Imaging*, *eWON* y *MB Connect Line*, cuyos sitios distribuyeron software infectado que contenía troyanos al momento de su descarga (Symantec, 2017).

En cuanto a la funcionalidad del *malware*, se identificaron al menos dos componentes clave:

1. Havex, diseñado para extraer información sensible como la libreta de direcciones de Microsoft Outlook y detalles del sistema de gestión de control.
2. Kragany, un troyano que permitía descargar y subir archivos, así como enviar datos a servidores de comando y control (C2), facilitando así el espionaje industrial y la persistencia en los sistemas comprometidos.

Red Eléctrica de Ucrania.

En el año 2015 alrededor de 225.000 usuarios de la red eléctrica que se abastecían de 30 subestaciones de la empresa AES Kyivoblenrgo se vieron afectadas en el suministro de energía eléctrica por un ataque coordinado durante unas cuantas horas (Hemsley & Fisher, 2018).

El ataque fue planeado a partir del empleo de diversas técnicas como el phishing, manipulación de documentos, ingeniería social, robo de credenciales, además de usar equipos de conexión remota como por ejemplo las IHM, la modificación del Firmware en los dispositivos ethernet.

El malware empleado fue el denominado *BlackEnergy* cuya instalación se logró gracias a los-troyanos provenientes de los esquemas anteriormente mencionados. El éxito de este ataque se debió a la intervención directa y coordinadas de un grupo de atacantes. Establecieron una conexión VPN con los escritorios remotos, lo que permitió enviar instrucciones desde el terminal de control del malware. Inyectaron un firmware corrupto en dispositivos de comunicaciones, tomaron el control de los SAI's y por último emplearon el malware KIIDisk con el cual borraron los discos duros y denegaron el servicio de atención al cliente vía telefónica.

Industroyer/Crashoverride.

La alerta IR-ALERT-H-16-056-01 detalla un ciberataque ocurrido en diciembre de 2015 contra la infraestructura crítica de Ucrania. Los atacantes utilizaron una campaña

de *spearphishing* para distribuir el malware *BlackEnergy 3* y obtener acceso a las redes IT de las compañías eléctricas. Una vez dentro, se movieron hacia las redes SCADA y desplegaron un componente destructivo llamado *KillDisk*, que corrompió los sistemas y los dejó inoperables. Para dificultar la recuperación, el malware también eliminó registros del sistema y atacó las fuentes de alimentación ininterrumpida (UPS), impidiendo el reinicio de los servidores (Cybersecurity and Infrastructure Security Agency, 2016).

Triton / Trisis / Hatman.

Detectado inicialmente en 2017, este software malicioso tenía como objetivo alterar el firmware de los dispositivos de seguridad integrados en Triconex de Schneider Electric. Esencialmente, introducía comandos adicionales, utilizando el protocolo de comunicación propio de estos dispositivos.

Aunque el resultado del ataque no fue el esperado, ya que el sistema entro en modo seguro, generando el detenimiento del proceso, alertó a los operadores encargados de la seguridad de la planta de industrial permitiendo aplicar medidas que lo mitigaron. Este ciberataque, marcó un precedente en cuanto a que los SIS siguen siendo un blanco para ciberataques y a que deben estar aislados del resto de la red de control (Soria-Olivas et al., 2018).

Botnets de Dispositivos IoT.

Como se ha mencionado anteriormente la tecnología IoT ha permitido la integración de todo tipo de dispositivos a la red para su control por lo que, aunque no se tienen registros de incidentes industriales si se debe entender que representan una nueva vulnerabilidad ya que son elementos de bajo costo que están conectados a las redes de las empresas.

Un ejemplo de malware que, aunque no ha afectado a una red industrial nos permite vislumbrar como podría funcionar, es *El Mirai*. El Mirai es un malware capaz de crear y controlar una red de dispositivos por parte del atacante con el fin de crear degeneraciones de servicio masivo (Soria-Olivas et al., 2018).

2.15. Estado del arte

En el contexto actual de la Industria 4.0, la convergencia entre tecnologías operativas (OT) y tecnologías de la información (IT) ha incrementado significativamente

la exposición de las infraestructuras industriales a ciber amenazas. Los sistemas de control industrial (ICS), incluyendo los entornos SCADA, han pasado de operar en redes aisladas a estar conectados a entornos digitales más amplios, lo que ha generado un aumento en los vectores de ataque y la complejidad de su defensa (Cheminod et al., 2012).

El uso de honeypots en Sistemas de Control Industrial (ICS) ha ganado relevancia como un método para estudiar el comportamiento de los atacantes. Investigaciones recientes se han centrado en comparar la efectividad de diferentes estrategias de despliegue, analizando cómo el nivel de interacción afecta la capacidad de atraer tráfico malicioso. En este contexto, herramientas de baja interacción como CONPOT son estudiadas para determinar si pueden capturar inteligencia de amenazas comparable a la de sistemas más complejos, evaluando sus limitaciones a la hora de emular protocolos industriales como Modbus y S7Comm en diversos entornos de red (Ondrikov et al., 2025).

Junto con los honeypots, la implementación de sistemas de análisis en tiempo real como ELK Stack (Elasticsearch, Logstash y Kibana) ha demostrado ser útil para el monitoreo y visualización de registros en redes industriales. Investigaciones como la del Instituto Nacional de Ciberseguridad de España (2018) muestran cómo la integración de honeypots con ELK permite no solo detectar intentos de intrusión, sino también correlacionar eventos y generar alertas automáticas.

En la aplicación de inteligencia artificial para la ciberseguridad de infraestructuras críticas, se han desarrollado enfoques basados en *deep learning* para hacer frente a amenazas complejas. Un ejemplo de ello es el trabajo de Zang et al. (2025), quienes proponen un modelo para la detección, comprensión y proyección de ataques de inyección de datos falsos en sistemas de potencia. Sin embargo, la implementación de estos modelos avanzados en entornos industriales reales plantea desafíos significativos, particularmente por la necesidad de contar con grandes volúmenes de datos para el entrenamiento y una infraestructura computacional robusta.

A pesar del avance en estas áreas, el estado actual del conocimiento evidencia que aún persisten limitaciones importantes: la mayoría de las soluciones han sido validadas en entornos simulados, existe escasa integración práctica entre honeypots, análisis en tiempo real e inteligencia artificial, y se requiere mayor investigación orientada a la adaptación de estas herramientas a infraestructuras críticas con restricciones operativas.

La presente investigación se enmarca en estos desafíos, proponiendo una solución integrada que combina honeypots ICS, monitoreo con ELK y análisis con algoritmos de IA en un entorno experimental controlado. Esta propuesta busca contribuir a la reducción de las brechas de seguridad en redes industriales, desde un enfoque técnico y práctico.

2.16. Aplicación de Inteligencia Artificial en la Ciberseguridad Industrial

El concepto de inteligencia artificial en la actualidad no hace referencia a un concepto único, sino que integra diversos elementos básicos del proceso cognitivos como lo son el aprendizaje, memorización, solución de problemas, razonamiento lógico y comprensión del lenguaje natural. Dichos procesos cuando son embebidos en una máquina permiten que esta de cierto modo emule el razonamiento humano y le permita a su vez adaptarse a su entorno.

La inteligencia artificial puede entenderse desde dos perspectivas principales: el desarrollo de máquinas que simulan la inteligencia humana y la creación de algoritmos para resolver problemas mediante el análisis de grandes volúmenes de datos. En el ámbito de la ciberseguridad, esta segunda capacidad es particularmente relevante.

La habilidad de la IA para procesar información a gran velocidad permite que los sistemas de defensa modernos puedan detectar amenazas en tiempo real. Además, mediante el análisis predictivo de riesgos, estas tecnologías tienen el potencial de anticipar y prevenir futuros ataques, fortaleciendo así la postura de seguridad de una organización.

2.17. Técnicas de Inteligencia Artificial Aplicadas a la Ciberseguridad

Como se analizó anteriormente la inteligencia artificial no solo aplica un método para analizar datos, es por ello esta herramienta pone a disposición distintos métodos para desempeñar las actividades de seguridad informática. Algunas de ellas se describen someramente a continuación.

Redes Neuronales Artificiales (RNA).

Para poder emplear esta técnica en la ciberseguridad con éxito se debe primeramente conocer el volumen de datos, la velocidad de análisis y la complejidad de tratamientos de cada tipo de dato que se estudiara (Castellanos Rojas et al., 2020)

Entonces las RNA se encargan de generar patrones de clasificación que permitan identificar cada tipo de ataque a partir de ataques que ya se hayan sufrido con anterioridad.

Agentes Inteligentes

En este caso se entiende por agente inteligente a la entidad capaz de percibir su entorno, procesar y responder de manera racional, cuando estos métodos se emplean para la ciberseguridad es un trabajo conjunto de diversos agentes que se comunican entre sí para compartir información y dar la respuesta más acorde al ataque detectado.

Sistemas Inmunes Artificiales

Esta técnica es un homólogo de la respuesta inmune del cuerpo humano donde se reacciona ante agentes patógenos externos formando anticuerpos con características únicas para enfrentar a un tipo de agente externo específico y así contrarrestar la infección.

Aprendizaje Automático

Es una técnica que se enfoca en diseñar y crear sistemas capaces de aprender y optimizar la performance de un sistema a partir de los datos que se tienen a su disposición. Para usar esta técnica en ciberseguridad se deben emplear patrones de comportamiento previamente identificados permitiendo posteriormente al sistema tomar decisiones basadas en datos y conclusiones anteriores (Kaspersky Lab., 2021).

2.18. Aplicaciones de las Técnicas de iA en Ciberseguridad

En los párrafos anteriores se describió cómo funcionan las técnicas de inteligencia artificial. En los párrafos siguientes se presentan algunos trabajos donde es una realidad la utilización de estas técnicas en la ciberseguridad.

Los sistemas de detección de intrusos (IDS) cuentan con aplicación de distintas técnicas de inteligencia artificial como las máquinas de soporte, redes neuronales, regresión multivariada adaptativa utilizando *splines* (SVM, ANN, MARS y LGP, por sus siglas en inglés respectivamente). Utilizando RNAs, los IDSs utilizan modelos de aprendizaje automático para reconocer patrones de ataque, tal como ocurre con los correos no deseados. Mediante la evaluación del contenido textual, pueden identificarse anomalías y proceder a su clasificación y tratamiento (Méndez et al., 2007).

El siguiente ejemplo corresponde a una herramienta integrada denominada Heroic, la cual se encarga de realizar la protección ante una diversidad de amenazas

especialmente de las vulnerabilidades presenten en *peer to peer* (P2P). Es una aplicación que permite almacenar datos como si fuera una huella digital colocando bloques de información de una manera ordenada, brindando protección contra el robo de identidad y detectar y desactivar amenazas en tiempo real (Heroic, 2021).

En el trabajo realizado por Bezobrazov et al. (2016) se desarrolló un programa que actúa como antivirus a partir de un clasificador de aplicaciones para poder detectar malware en dispositivos Android. Para realizar esta función se emplean RNAs en conjunto con la técnica de sistema inmune artificial. Tiene la peculiaridad de poder adaptarse de manera automática y así detectar y bloquear las posibles amenazas (Bezobrazov, Cakir, & Aidin, 2016).

El siguiente ejemplo corresponde a un sistema de filtrado de seguridad basado en Deep Learning. Esta técnica se implementó en dos etapas: la etapa de preentrenamiento, donde se busca establecer el estado basal del sistema es decir su comportamiento normal. Posteriormente se realiza un entrenamiento supervisado en el cual se ajustan los datos basales con el fin de obtener a la salida datos buenos. Para ello se emplean datos normales y anormales. Finalmente, los resultados identifican los datos anómalos (Ferrado Cuenca, 2016).

El trabajo desarrollado por Muñoz & Escribano (2015), consiste en emplear una arquitectura GAN (Generative Adversarial Network) formada por dos RNA que compiten entre sí para estabilizar el sistema por medio de un punto de equilibrio entre el ruido en la imagen dañada y el detector de imágenes erradas. Esta propuesta es exitosa en la medida de que el atacante entienda como se cifra la información del usuario.

2.19. Limitaciones de la Inteligencia Artificial en la Ciberseguridad

Hemos visto que existen muchas bondades de las técnicas de inteligencia artificial al servicio de la ciberseguridad, como, por ejemplo: su adaptabilidad, el análisis en tiempo real y la interoperabilidad de diversas técnicas. Sin embargo, también se deben tener en cuenta las desventajas que implican estas técnicas, como ser:

- Se debe considerar el entorno en el que trabaja debido a que su aprendizaje proviene de este por lo que no se pueden extrapolar los resultados del aprendizaje adquirido a un nuevo ambiente o incluso si en el mismo ambiente se le han cambiado sus condiciones.

-
- Para iniciar el aprendizaje se necesita contar con una gran cantidad de datos de entrada que tengan la calidad para entrenar al sistema de manera exhaustiva.
 - El sistema de seguridad puede ser vulnerable si el atacante conoce el comportamiento del sistema.

Por lo que debemos ser conscientes que mantener un sistema completamente seguro no solo depende de los sistemas de defensas, sino que también se deben propiciar las buenas prácticas de seguridad industrial dentro de las empresas es un conjunto de elementos que no solo dependerán de programas informáticos sino del acceso a los equipos.

Capítulo 3

Desarrollo Técnico

3.1. Diseño, Implementación y Análisis

El diseño y construcción técnica presentados en este trabajo están estructurados en tres implementaciones significativas.

En la primera fase, se incorporó el uso de un servidor Honeypot con el objetivo de simular un entorno de control industrial, específicamente un sistema de control de supervisión y adquisición de datos (ICS/SCADA). Este enfoque tiene como finalidad hacer creer al atacante que está accediendo a un ambiente industrial. La intención primordial de esta configuración es el registro sistemático de los ataques que dicho servidor recibe, para posteriormente poder analizarlos y luego, proponer estrategias efectivas para la prevención de futuros ataques.

En la segunda etapa, se diseñó y programó un honeypot minimalista en el lenguaje Python, cuyo propósito principal es emular un servidor SSH para la recepción y registro de conexiones entrantes. Los datos obtenidos son almacenados en una base de datos y en un servidor web, facilitando la consulta y visualización de la información registrada en una forma más accesible y visual.

En la tercera y última implementación, se desarrolló una aplicación que emula un dispositivo SCADA y monitorea tráfico TCP/IP y SNMP. Esta aplicación analiza los datos capturados en tiempo real con el objetivo de detectar cualquier actividad sospechosa o anómala. En caso de identificar cualquier actividad irregular, la aplicación procede a capturar los detalles de dicha actividad en un archivo de registro y toma medidas correctivas para solucionar la anomalía detectada. Esta implementación también comprende un sistema de análisis de logs ELK (Elasticsearch + Logstash + Kibana) que se encuentra desplegado en un contenedor (Docker).

La combinación de estas tres implementaciones proporciona un sistema robusto y versátil para el análisis, detección y prevención de ataques en un contexto de control industrial, conformando el cuerpo principal de las contribuciones técnicas de esta investigación.

Todo el desarrollo de este honeypot se ha llevado a cabo utilizando Git como sistema de control de versiones. Para el acceso y descarga de la totalidad del proyecto, el

repositorio correspondiente se halla a disposición a través del enlace proporcionado a continuación:

https://github.com/cuervo1974/honey_ssh

3.2. Primera Implementación. T-POT/CONPOT

La implementación del honeypot se llevó a cabo a través de la plataforma T-Pot (Telekom Security), la cual se basa en la distribución Linux, Debian 11. T-Pot es un desarrollo de código abierto que consolida honeypots de baja y alta interacción en un único sistema, proporcionando la capacidad de emular una variedad de servicios de red y hardware de red vulnerable. Entre estos se incluyen Android ADB, routers, SCADA, SSH, Telnet, DICOM, Elasticsearch, FTP, RDP, HTTP/S, PostgreSQL, MSSQL, POP3, SMTP, SMB, entre otros

En la fase de despliegue de la plataforma, se evaluaron tres posibles opciones:

1. Implementarla en una instancia EC2 en una nube pública.
2. Instalarla en una Raspberry PI ubicada detrás de una DMZ en mi red doméstica.
3. Instalarla en una máquina virtual en VirtualBox en una notebook, también ubicada detrás de una DMZ en mi red doméstica.

Las nubes públicas analizadas incluyeron Microsoft Azure (Azure), Amazon Web Service (AWS) y Google Cloud Platform (GCP). Sin embargo, se decidió no proseguir con estas opciones debido al costo económico en dólares que supondría la implementación de esta solución.

La segunda opción, la implementación en una Raspberry PI, se realizó con éxito inicialmente, con el honeypot ejecutándose de manera efectiva. Sin embargo, tras los primeros registros de ataques, el sistema experimentó bloqueos, dejando de responder y de registrar eventos. Tras un análisis detallado, se atribuyó la incompatibilidad entre la versión del sistema operativo instalado en la Raspberry PI y las librerías instaladas por la suite T-POT como la causa de estos incidentes, por lo que esta opción fue descartada.

Finalmente, después de descartar las dos opciones anteriores, se procedió a instalar el sistema en una máquina virtual en VirtualBox de Oracle. Esta VM se situó en una red DMZ (zona desmilitarizada) creada en mi router doméstico, lo que permitió exponerla a Internet sin comprometer la seguridad de mi red.

Para la instalación de la plataforma, se descargó la imagen ISO desde el repositorio oficial de T-Pot en GitHub (<https://github.com/telekom-security/tpotce/releases>), se inició la VM con dicha imagen ISO, y al cabo de 10 minutos el sistema quedó instalado y configurado. Se accede desde:

https://IP_server:64297

3.3. Evidencias Encontradas Para Su Posterior Análisis.

Por consola, en el servidor, se pueden ver todos los servicios y honeypots en ejecución y los puertos de escucha de cada uno de estos, como se ve en la figura 7.

Figura 7

Muestra de los servicios y honeypot en ejecución y sus puertos de escucha.

```
[=====] System [=====]
DATE: Thu 07 Apr 2022 01:41:51 PM UTC
UPTIME: 13:41:52 up 10:21, 2 users, load average: 0.66, 0.80, 0.79
T-POT: ACTIVE
BLACKHOLE: DISABLED

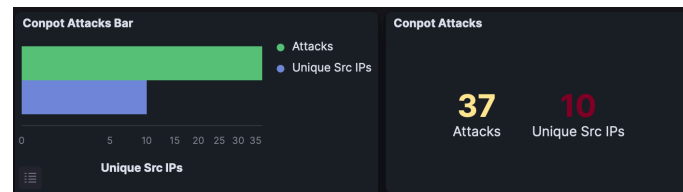
NAME      STATUS      PORTS
adbbhoney Up 10 hours (healthy) 0.0.0.0:5555->5555/tcp
ciscoasa Up 10 hours 0.0.0.0:5000->5000/udp, 0.0.0.0:8443->8443/tcp
citrixhoneypot Up 10 hours 0.0.0.0:443->443/tcp
conpot_guardian_ast Up 10 hours (healthy) 0.0.0.0:10001->10001/tcp
conpot_iecl84 Up 10 hours (healthy) 0.0.0.0:161->161/udp, 0.0.0.0:2404->2404/tcp
conpot_ipmi Up 10 hours (healthy) 0.0.0.0:623->623/udp
conpot_kamstrup_382 Up 10 hours (healthy) 0.0.0.0:1025->1025/tcp, 0.0.0.0:50100->50100/tcp
cowrie Up 10 hours 0.0.0.0:22-23->22-23/tcp
ddospot Up 10 hours 0.0.0.0:19->19/udp, 0.0.0.0:53->53/udp, 0.0.0.0:123->123/udp, 0.0.0.0:1900->1900/udp
discompt Up 10 hours 0.0.0.0:1112->1112/tcp
dionaea Up 10 hours (healthy) 0.0.0.0:20-21->20-21/tcp, 0.0.0.0:42->42/tcp, 0.0.0.0:81->81/tcp, 0.0.0.0:135->135/tcp, 0.0.0.0:445->445/tcp, 0.0.0.0:1433->1433/tcp, 0.0.0.0:1723->1723/tcp, 0.0.0.0:1883->1883/tcp, 0.0.0.0:3306->3306/tcp, 0.0.0.0:27017->27017/tcp, 0.0.0.0:69->69/udp
elasticpot Up 10 hours 0.0.0.0:9200->9200/tcp
elasticsearch Up 10 hours (healthy) 127.0.0.1:64290->9200/tcp
ewsposter Up 10 hours
fatt Up 10 hours
heralding Up 10 hours 0.0.0.0:110->110/tcp, 0.0.0.0:143->143/tcp, 0.0.0.0:465->465/tcp, 0.0.0.0:993->993/tcp, 0.0.0.0:995->995/tcp, 0.0.0.0:1080->1080/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5900->5900/tcp
honeypot Up 10 hours
ipphoney Up 10 hours 0.0.0.0:631->631/tcp
kibana Up 10 hours (healthy) 127.0.0.1:64290->5601/tcp
logstash Up 10 hours (healthy)
mailoney Up 10 hours 0.0.0.0:25->25/tcp
map_data Up 10 hours
map_redis Up 10 hours
map_web Up 10 hours 127.0.0.1:64290->64299/tcp
medpot Up 10 hours 0.0.0.0:2575->2575/tcp
nginx Up 10 hours
p0f Up 10 hours
redishoneypot Up 10 hours 0.0.0.0:6379->6379/tcp
sentrypeer Up 10 hours 0.0.0.0:5060->5060/udp
snare Up 10 hours 0.0.0.0:80->80/tcp
spiderfoot Up 10 hours (healthy) 127.0.0.1:64303->6080/tcp
suricata Up 10 hours
tanner Up 10 hours
tanner_api Up 10 hours
tanner_phpox Up 10 hours
tanner_redis Up 10 hours
```

Nota. Elaboración propia.

Pasadas 14 horas ya se habían producidos 37 ataques desde 10 IPs distintas, como se ve en los dashboards que muestra el servidor de Kibana que provee la plataforma (figuras 8 y 9).

Figura 8

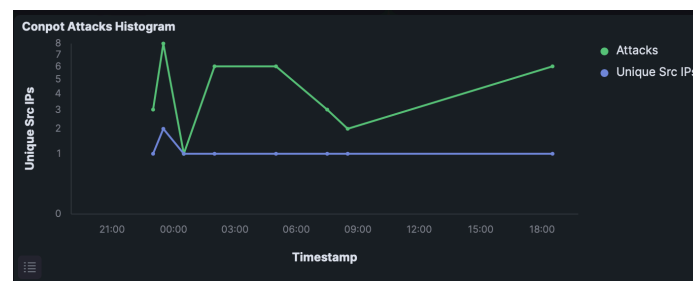
Dashboard de estadísticas de ataques al Honeypot CONPOT.



Nota. Elaboración propia.

Figura 9

Dashboard Histograma de ataques al Honeypot CONPOT.

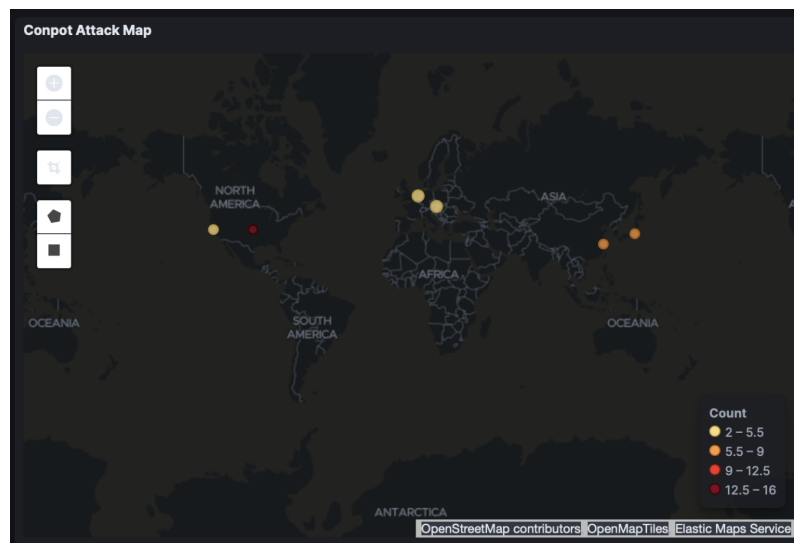


Nota. Elaboración propia.

En la figura 10 se muestra un mapa con la ubicación geográfica del origen de los ataques.

Figura 10

Mapa de la ubicación geográfica de los ataques al Honeypot.



Nota. Elaboración propia.

En la figura 11 se puede ver el Dashboard donde se muestra la reputación de las IPs origen de los ataques, los países desde donde se realizaron los ataques, los tipos de eventos, los protocolos atacados, la distribución horaria de los ataques por país y la distribución horaria de los ataques por puerto destino.

Figura 11
Dashboards de reputación, países de origen, tipos de eventos, protocolos y distribución horaria de los ataques al Honeypot CONPOT.



Nota. Elaboración propia.

En la figura 12 se muestra el top 10 de los proveedores desde donde se produjeron los ataques y las 10 IPs desde donde se iniciaron los mismos.

Figura 12
Top 10 de los proveedores e IPs origen.

Conpot - Attacker AS/N - Top 10		Conpot - Attacker Src IP - Top 10	
AS	ASN	Source IP	Count
58461	No.288,Fu-chun Road	162.142.125.10	6
174	Cogent Communications	162.142.125.9	6
41436	Kamatera Inc	183.136.225.9	6
14061	DigitalOcean, LLC	43.130.11.228	6
202425	IP Volume inc	162.142.125.210	3
		185.220.204.178	3
		192.241.211.213	2
		195.144.21.56	2
		94.102.49.159	2
		209.17.97.2	1

Nota. Elaboración propia.

Se puede ver en el log los tipos de ataques para luego analizar qué acciones tomar (figura 13).

Figura 13

Registro de los ataques.

Time	T-Pot Hostname	Type	DestPort	Country
> Feb 1, 2022 @ 18:02:37.241	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:36.499	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:35.699	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:34.855	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:34.065	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:33.428	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:32.460	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:31.651	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:30.864	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:30.146	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:29.401	urbandisguise	Dionaea	445	EG
> Feb 1, 2022 @ 18:02:29.000	urbandisguise	Honeytrap	12346	US

Nota. Elaboración propia.

3.4. Segunda Implementación. Desarrollo del Honeypot con Python

El presente estudio se concentra en la segunda implementación del proyecto, en la que se efectuó el diseño y desarrollo de un honeypot orientado a la emulación de un servidor SSH. Esta herramienta fue programada para llevar a cabo el registro de las conexiones entrantes, facilitar la recepción de las comunicaciones, y simular con eficacia la interacción entre usuario y servidor. Para la visualización de una porción de la información almacenada en la base de datos del honeypot, se incorporó además un servidor web.

El desarrollo del honeypot se realizó mediante el lenguaje de programación Python, haciendo uso de la biblioteca *Paramiko*, la cual ofrece una implementación eficaz de SSH en el mencionado lenguaje. Con el propósito de almacenar la información de las conexiones capturadas, se optó por utilizar el sistema de gestión de base de datos *MariaDB*.

El propósito central de este proyecto fue la construcción de un honeypot efectivo, capaz de replicar un servidor SSH real y recabar datos de las intentonas de conexión con intenciones maliciosas dirigidas a tal servidor. Para cumplir con este objetivo, se diseñó una arquitectura que permitiera la captura de las solicitudes de conexión entrantes, la extracción de información relevante tal como la dirección IP, el nombre de usuario y la

contraseña utilizados en el intento de conexión, y el almacenamiento de estos datos en una base de datos para su análisis posterior.

En el proceso de implementación del honeypot, se aplicó la biblioteca Paramiko para establecer un servidor SSH ficticio que aceptara las conexiones entrantes. Para este servidor se generó una clave RSA y se establecieron los parámetros requeridos, como el puerto, el nombre de usuario y la contraseña. La clase *SSHServerHandler* se utilizó para gestionar las solicitudes de autenticación de los clientes conectados y registrar las conexiones en la base de datos mediante consultas SQL.

El desarrollo incorporó un módulo encargado de administrar la lógica de interacción para el usuario. Este módulo abarca una serie de los comandos más comúnmente utilizados en las máquinas, y su tarea es llevar a cabo la lógica apropiada en función de la entrada del usuario. Si el comando no está definido, el módulo retornará el mensaje de error *"command not found"*.

La mayoría de estos comandos simulados tienen un doble propósito: ofrecer una experiencia de interacción realista y, a la vez, registrar toda la actividad para su posterior análisis. Por ejemplo, al recibir un comando de descarga como curl o wget, el sistema almacena la dirección IP del atacante y registra el nombre del archivo solicitado. La Figura 14 muestra un listado de los archivos a los que se intentó acceder mediante estos comandos durante las pruebas.

Figura 14

Listado de archivos accedidos.

```
total 60
drwxr-xr-x 2 cuervo cuervo 4096 Jun 24 01:04 ./
drwxr-x--- 4 cuervo cuervo 4096 Jun 24 01:11 ../
-rw-r--r-- 1 rr      rr      1236 Jan  2 05:31 LinuxTF.1
-rw-r--r-- 1 rr      rr      34945 Apr  2 07:55 LinuxTF.2
-rw-r--r-- 1 rr      rr         67 Dec 22  2022 bins.sh
-rw-r--r-- 1 rr      rr        102 Jan 12 16:10 bins.sh.2
-rw-r--r-- 1 rr      rr        102 Mar 18 20:31 bins.sh.3
```

Nota. Elaboración propia.

El servidor web sirve como una interfaz intuitiva y eficiente para la visualización de la información alojada en la base de datos, la cual presenta tablas que contienen información relevante para el análisis y monitoreo de nuestro sistema. Dichas tablas abarcan diversas estadísticas y métricas relacionadas con la conectividad, como los

comandos de conexión, los países desde donde se originan las conexiones, las últimas muestras obtenidas, así como los 10 principales IPs atacantes y los 10 principales países de origen de los ataques.

En la tabla de estadísticas de conectividad se muestran datos sobre las IPs únicas conectadas, las muestras únicas detectadas y los usuarios/contraseñas únicas detectados. Esto brinda una visión general de la actividad de conectividad en el sistema.

En la tabla de últimos comandos se registran los comandos ejecutados en el sistema, incluyendo información detallada como el nombre de usuario, contraseña y comando utilizado.

La tabla de últimos países de conexiones muestra información relacionada a las direcciones IP, países, códigos ISO, códigos ASN, organizaciones ASN, número de listas negras y nivel de riesgo asociados a las conexiones. Esto permite identificar los países de origen de las conexiones y evaluar las posibles amenazas.

En la tabla de últimas muestras obtenidas se registran los nombres, URLs y hashes SHA256 de las muestras maliciosas detectadas, facilitando el análisis de las amenazas potenciales presentes en el sistema.

Las tablas de top 10 IPs atacantes y top 10 países de ataques presentan las IPs y países con mayor número de conexiones y ataques registrados, lo que ayuda a identificar las fuentes más activas de posibles ataques al sistema.

En conjunto, estas tablas ofrecen una visión integral de la conectividad, los comandos, las muestras maliciosas y los posibles ataques en nuestro sistema. Esta información permite tomar decisiones informadas para fortalecer la seguridad y proteger la infraestructura crítica del sistema.

Este servidor web funciona de manera autónoma y solamente entra en ejecución cuando se necesita acceder a dicha información.

Para propiciar una interacción fluida entre la API de la aplicación con el portal web Virustotal y con sitios web de calificación de IPs, se han instaurado dos módulos específicos. El primero de estos módulos se encarga de la recolección de información pertinente a las muestras que se buscan descargar, extrayendo detalles tales como el número de detecciones de antivirus, la vinculación con malware y la fecha de la primera detección, entre otros aspectos relevantes. Esta operación se lleva a cabo generando un hash con el algoritmo SHA-256 de la muestra descargada y posteriormente se realiza una

comparación con la base de datos de reputación de virus proporcionada por Virustotal. Los resultados de esta verificación son almacenados en la base de datos para futuros análisis.

El segundo módulo implementado facilita la adquisición de datos relacionados con las direcciones IP de origen de los ataques. Esta API provee detalles de geolocalización que comprenden el país, el código de país, la ciudad, el número del sistema autónomo (conocido por sus siglas en inglés, ASN) y las coordenadas geográficas.

3.5. Evidencias Encontradas Para Su Posterior Análisis.

Servidor Web

Nuestro servidor web, figura 15, muestra diversas tablas que contienen información relevante para el análisis y monitoreo de nuestro sistema. Estas tablas incluyen estadísticas de conectividad, comandos de conexiones, últimos países de conexiones, últimas muestras obtenidas, top 10 IPs atacantes y top 10 países de ataques.

Figura 15

Estadísticas Servidor Web.



Nota. Elaboración propia.

Log “Últimos Comandos”

Como se ve en la figura 16, el log de “Últimos comandos” contiene registros detallados de los comandos ejecutados en el sistema. Estos registros capturan información sobre la fecha, hora, nombre de usuario, contraseña (si se aplica) y el propio comando

utilizado. El propósito de este registro es supervisar y auditar las actividades en el sistema para detectar comportamientos anómalos y posibles intentos de intrusión. Los registros de "Últimos comandos" proporcionan una visión detallada de las operaciones realizadas, lo que es esencial para mantener la seguridad y la integridad del sistema. Estos registros son valiosos para investigaciones de seguridad, análisis forense y respuesta a incidentes.

Figura 16

Evidencias sobre los registros del log "Últimos comandos".

```
[2022-12-15 09:32:15] - Usuario 'admin' realizó el comando 'GET /get_data'
[2022-12-19 14:20:42] - Usuario 'root' realizó el comando 'EXECUTE /exploit'
[2023-01-05 18:45:23] - Usuario 'guest' realizó el comando 'DELETE /delete_data'
[2023-01-10 10:58:37] - Usuario 'admin' realizó el comando 'UPDATE /update_data'
[2023-01-18 16:12:51] - Usuario 'operator' realizó el comando 'CREATE /new_object'
[2023-02-02 11:27:09] - Usuario 'engineer' realizó el comando 'RESTART /system'
[2023-02-14 15:40:25] - Usuario 'admin' realizó el comando 'READ /read_data'
[2023-02-22 12:05:07] - Usuario 'guest' realizó el comando 'SET /set_data'
[2023-03-05 09:21:33] - Usuario 'admin' realizó el comando 'GET /get_data'
[2023-03-10 13:48:19] - Usuario 'root' realizó el comando 'EXECUTE /exploit'
[2023-03-18 17:56:24] - Usuario 'guest' realizó el comando 'DELETE /delete_data'
[2023-03-24 10:10:58] - Usuario 'admin' realizó el comando 'UPDATE /update_data'
[2023-03-30 16:32:41] - Usuario 'operator' realizó el comando 'CREATE /new_object'
[2023-04-06 09:45:59] - Usuario 'engineer' realizó el comando 'RESTART /system'
[2023-04-13 14:02:12] - Usuario 'admin' realizó el comando 'READ /read_data'
[2023-04-19 11:20:37] - Usuario 'guest' realizó el comando 'SET /set_data'
[2023-04-25 15:42:59] - Usuario 'admin' realizó el comando 'GET /get_data'
[2023-04-30 10:05:18] - Usuario 'root' realizó el comando 'EXECUTE /exploit'
```

Nota. Elaboración propia.

Log "Último Países Conexiones"

El log de "Últimos países de conexiones", figura 17, contiene registros de las conexiones realizadas al sistema, proporcionando información sobre los países de origen de dichas conexiones. Estos registros incluyen datos como la dirección IP, el país asociado, el código ISO del país, el código ASN (Sistema Autónomo) y la organización del ASN, el número de listas negras en las que se encuentra y el nivel de riesgo asignado. El propósito de este registro es monitorear y analizar las conexiones entrantes al sistema, identificando los países de origen y evaluando posibles riesgos y amenazas. Los registros de "Últimos países de conexiones" son valiosos para el análisis de la geolocalización de las conexiones, la detección de patrones de actividad sospechosa y la toma de decisiones relacionadas con la seguridad y la protección del sistema. Estos registros proporcionan información crucial para la identificación temprana de posibles ataques y la implementación de medidas preventivas para salvaguardar la infraestructura crítica del sistema.

Figura 17

Evidencias sobre los registros del log “Últimos países conexiones”.

[2023-12-19 14:20:42]	- IP: 103.21.244.1, País: China, ISO: CN, ASN Code: AS13335, ASN Organization: Cloudflare, Inc., Blacklists: 1, Nivel de riesgo: Medio
[2023-01-05 18:45:23]	- IP: 81.2.69.160, País: Alemania, ISO: DE, ASN Code: AS24948, ASN Organization: Hetzner Online GmbH, Blacklists: 0, Nivel de riesgo: Bajo
[2023-01-10 10:58:37]	- IP: 187.45.187.53, País: Brasil, ISO: BR, ASN Code: AS27715, ASN Organization: Locaweb Servicos de Internet S/A, Blacklists: 3, Nivel de riesgo: Alto
[2023-01-18 16:12:51]	- IP: 2a00:23c4:4a46:bb00:3c2d:e40d:2c4e:67d9, País: Reino Unido, ISO: GB, ASN Code: AS13445, ASN Organization: Cisco Systems Ironport Division, Blacklists: 1, Nivel de riesgo: Medio
[2023-02-02 11:27:09]	- IP: 123.56.78.90, País: China, ISO: CN, ASN Code: AS37963, ASN Organization: Hangzhou Alibaba Advertising Co.,Ltd., Blacklists: 2, Nivel de riesgo: Alto
[2023-02-14 15:40:25]	- IP: 85.14.250.54, País: Rusia, ISO: RU, ASN Code: AS12389, ASN Organization: Rostelecom, Blacklists: 0, Nivel de riesgo: Bajo
[2023-02-22 12:05:07]	- IP: 2001:67c:2e8:22::c100:68d1, País: Países Bajos, ISO: NL, ASN Code: AS25182, ASN Organization: P&T Luxembourg, Blacklists: 1, Nivel de riesgo: Medio
[2023-03-05 09:21:33]	- IP: 45.56.99.78, País: Estados Unidos, ISO: US, ASN Code: AS63949, ASN Organization: Linode, LLC, Blacklists: 2, Nivel de riesgo: Alto
[2023-03-10 13:48:19]	- IP: 117.204.144.33, País: India, ISO: IN, ASN Code: AS9829, ASN Organization: National Internet Backbone, Blacklists: 0, Nivel de riesgo: Bajo
[2023-03-18 17:56:24]	- IP: 1.144.111.69, País: Australia, ISO: AU, ASN Code: AS7545, ASN Organization: TPG Telecom Limited, Blacklists: 1, Nivel de riesgo: Medio
[2023-03-24 10:10:58]	- IP: 77.30.200.70, País: Austria, ISO: AT, ASN Code: AS8437, ASN Organization: Telekom Austria, Blacklists: 2, Nivel de riesgo: Alto
[2023-04-06 09:45:59]	- IP: 202.179.25.114, País: Japón, ISO: JP, ASN Code: AS17506, ASN Organization: UCOM Corp., Blacklists: 1, Nivel de riesgo: Medio
[2023-04-13 14:02:12]	- IP: 190.86.209.45, País: Colombia, ISO: CO, ASN Code: AS10620, ASN Organization: Telmex Colombia S.A., Blacklists: 0, Nivel de riesgo: Bajo
[2023-04-19 11:20:37]	- IP: 195.154.78.33, País: Francia, ISO: FR, ASN Code: AS12876, ASN Organization: Online S.a.s., Blacklists: 2, Nivel de riesgo: Alto
[2023-04-25 15:42:59]	- IP: 93.198.134.50, País: Rusia, ISO: RU, ASN Code: AS12695, ASN Organization: LLC Digital Network, Blacklists: 1, Nivel de riesgo: Medio

Nota. Elaboración propia.

Log “Últimas Muestras Obtenidas”

El log de "Últimas muestras obtenidas", figura 18, contiene registros de las muestras maliciosas capturadas durante el análisis de los ataques recibidos por nuestro sistema. Estos registros incluyen información sobre la fecha, hora, nombre de la muestra, URL y su hash SHA256 único. El propósito de este registro es monitorear y auditar las amenazas potenciales que podrían comprometer la seguridad del sistema. Los registros de "Últimas muestras obtenidas" brindan información valiosa sobre las muestras de malware, troyanos, exploits, rootkits y spyware conocidos que se utilizaron en los ataques dirigidos a nuestro sistema. Estos registros son fundamentales para la detección, mitigación y análisis de amenazas, y ayudan a proteger el sistema contra actividades maliciosas.

Figura 18

Evidencias sobre los registros del log “Últimas muestras obtenidas”.

[2023-12-19 15:09:32:12]	- Nombre: Emotet, URL: http://103.21.244.1/malware/emotet.exe , SHA256: e1f685a8d7e345df8ff75e5e92897c21e6696bda781a7381f98e51a25a3a
[2023-12-19 14:20:42]	- Nombre: Trickbot, URL: http://81.2.69.160/malware/trickbot.exe , SHA256: 47b78c6f2d7c8e8e3b21b43b74b67ed5a7949d5da1a1f34a3f8a920a674
[2023-01-05 18:45:23]	- Nombre: Stuxnet, URL: http://187.45.187.53/malware/stuxnet.exe , SHA256: 32b1f94e8bde5264d76d9566a1db58d93e3a5d41e5b08ce1e64a76a986a0d05d
[2023-01-10 10:58:37]	- Nombre: BlackEnergy, URL: http://2a00:23c4:4a46:bb00:3c2d:e40d:2c4e:67d9/malware/blackenergy.exe , SHA256: 6e834b7a7e5c3bb1ce4300e0baeb782cda2e7975b20a4fb22b18e668ce4d95f
[2023-01-18 16:12:51]	- Nombre: NotPetya, URL: http://123.56.78.90/malware/notpetya.exe , SHA256: b987c4dcf24cf6e97dd07ac80ebcb9dbfa75bb0dbf23a0f39e708f9f8a5fefa
[2023-02-02 11:27:09]	- Nombre: Industroyer, URL: http://85.14.250.54/malware/industroyer.exe , SHA256: 84ed71a4ccacadafea1598be0a7a2f5d2dc5e8f7e6edf61c88e747a35dcd39
[2023-02-14 15:40:25]	- Nombre: Triton, URL: http://2001:67c:2e8:22::c100:68d1/malware/triton.exe , SHA256: 0b69b2e7eb48d35be0e64e20a4e2bc8a70b1c13e6d4ae2f0c39819c3d249894
[2023-02-22 12:05:07]	- Nombre: Duqu, URL: http://45.56.99.78/malware/duqu.exe , SHA256: 364f8e9ce8e7ab7051f253850bb2b4e11ffdd09a6b8def6a0c7b0a616dc4e5
[2023-03-05 09:21:33]	- Nombre: Flame, URL: http://117.204.144.33/malware/flame.exe , SHA256: 8cd1911e4ae8fb03c1e1975b57b5cd5d482d3eb0634332c5ed75ee202957d
[2023-03-10 13:48:19]	- Nombre: Wannacry, URL: http://1.144.111.69/malware/wannacry.exe , SHA256: a12597f6d46a566517a0598a813b05e4d1b6e2a515dd8f8f09c6a02b979d
[2023-03-18 17:56:24]	- Nombre: Petya, URL: http://77.30.200.70/malware/petya.exe , SHA256: 9d1b4f1d5e7e55eb428b2f5a4ab85fd7f239617d3047a7a20cc90b6a6a358a
[2023-03-24 10:10:58]	- Nombre: Zeus, URL: http://202.179.25.114/malware/zeus.exe , SHA256: 7548a59addbfbeb7f2a831e98d1a3a30f8963bb23c08da0aff36fb61d09f6a
[2023-04-06 09:45:59]	- Nombre: Conficker, URL: http://190.86.209.45/malware/conficker.exe , SHA256: 19e9e6be95f1ab39ea9a3e16f72fd0d3b741e8f52e0df1d78a55d4fe70f6485
[2023-04-13 14:02:12]	- Nombre: Slammer, URL: http://195.154.78.33/malware/slammer.exe , SHA256: f8d4f6d2c7b87da7bc9bc22107c8e6b3a45f6354ed9862b202be98ea89b88b8b
[2023-04-19 11:20:37]	- Nombre: DarkComet, URL: http://93.158.134.50/malware/darkcomet.exe , SHA256: 27969042c7f82b32ff9ed35a5c879e6db35193721152749558de2b0afcb5ef06
[2023-04-25 15:42:59]	- Nombre: ZeroAccess, URL: http://192.168.5.1/malware/zeroaccess.exe , SHA256: 63e9b2c2c1e72e0a8e979732cac1e8b0d6b00f4e51f830beaa01c6475dd60da

Nota. Elaboración propia.

3.6. Tercera implementación. Desarrollo del emulador SCADA con Python

En esta tercera implementación, se creó una aplicación para la emulación de dispositivos SCADA y el análisis de tráfico en tiempo real mediante una infraestructura de contenedores. Para ello, se empleó la tecnología *Docker*, la cual permite lograr una gestión eficiente, escalabilidad y portabilidad del sistema al encapsular cada componente

en un contenedor independiente (Docker, s.f.). Este enfoque garantiza el aislamiento de los componentes y facilita su despliegue en diversos entornos.

En primer lugar, se procede a configurar un contenedor Docker específico para la aplicación de emulación SCADA. Para ello, se emplea una imagen Docker personalizada que incluye la biblioteca Scapy y sus dependencias necesarias para llevar a cabo la emulación de dispositivos SCADA y la captura de tráfico TCP/IP y SNMP. Este contenedor está preparado para recibir y enviar paquetes de red, permitiendo así la emulación precisa y el posterior análisis de los datos capturados.

Con el fin de realizar el análisis en tiempo real de los datos capturados y detectar actividades sospechosas o anómalas, se desplegará un segundo contenedor Docker. Este contenedor estará configurado con una imagen personalizada que incluirá las bibliotecas Pandas, NumPy y otras dependencias necesarias para el análisis de datos en tiempo real. El objetivo principal de este contenedor será recibir los paquetes capturados por la aplicación de emulación SCADA, procesarlos en tiempo real y aplicar técnicas de análisis de datos para la detección de actividades sospechosas o anómalas.

En lo que respecta al stack ELK (Elasticsearch, Logstash y Kibana), se desplegarán tres contenedores Docker independientes para cada uno de los componentes. En primer lugar, se configurará un contenedor Docker utilizando una imagen oficial de Elasticsearch. Este contenedor será responsable de almacenar e indexar los registros capturados, proporcionando una plataforma distribuida para la búsqueda y análisis de estos. Se realizará una configuración adecuada de Elasticsearch a través del archivo de configuración *elasticsearch.yml*, definiendo índices y estableciendo parámetros como el tamaño de memoria asignado, la configuración de almacenamiento y las políticas de retención de registros.

A continuación, se desplegará un contenedor Docker utilizando una imagen oficial de Logstash. Este contenedor actuará como un procesador de registros en tiempo real, recibiendo los registros enviados por la aplicación de emulación SCADA y aplicando transformaciones y filtraciones antes de enviarlos a *Elasticsearch* para su indexación. Para ello, se definirán archivos de configuración específicos, como *logstash.conf*, donde se establecerán los filtros y transformaciones necesarios para extraer información relevante y mejorar la estructura de los datos capturados.

Por último, se implementará un contenedor Docker utilizando una imagen oficial de Kibana. Kibana proporcionará una interfaz de usuario web a través de la cual será

posible explorar y visualizar los registros almacenados en Elasticsearch. Se configurarán paneles de visualización, gráficos y tablas pertinentes para el análisis y detección de patrones o anomalías en los datos capturados.

En síntesis, la infraestructura de contenedores Docker permitirá la interconexión fluida entre los diferentes componentes del sistema. La aplicación de emulación SCADA enviará los registros de actividad sospechosa al contenedor de *Logstash* mediante la red Docker, donde serán procesados y posteriormente enviados a Elasticsearch para su almacenamiento. La aplicación de análisis en tiempo real se conectará a Elasticsearch para descargar los registros relevantes y realizar el análisis utilizando las bibliotecas *Pandas* y *NumPy*.

En conclusión, la implementación propuesta utilizando una infraestructura de contenedores Docker brindará una solución modular, escalable y portátil para la emulación de dispositivos SCADA, la captura y análisis de tráfico en tiempo real, y la integración con el stack *ELK*. La utilización de Docker facilitará el despliegue y la gestión de los componentes, asegurando un entorno aislado y optimizado para cada uno de ellos. Asimismo, esta arquitectura permitirá aprovechar las ventajas de la virtualización y la gestión de recursos de manera eficiente.

3.7. Desarrollo y Configuración.

Emulador SCADA.

Para crear la aplicación que emule un dispositivo SCADA, figura 19, se utilizó la biblioteca de Python Scapy para capturar el tráfico TCP/IP y SNMP. Scapy es una biblioteca de Python que permite crear, enviar, capturar y analizar paquetes de red.

Figura 19

Desarrollo Emulador SCADA.

```
def emular_dispositivo_SCADA():  
    # Definir la interfaz de red a escuchar  
    interfaz_red = "eth0"  
  
    # Definir el filtro de captura para tráfico TCP/IP y SNMP  
    filtro = "tcp or udp or snmp"  
  
    # Comenzar la captura de paquetes  
    packets = sniff(iface=interfaz_red, filter=filtro, prn=analizar_paquete)
```

Nota. Elaboración propia.

Este código utiliza la función ‘*sniff*’ de Scapy para capturar el tráfico TCP/IP y SNMP, y la función ‘*process_packet*’ para procesar cada paquete capturado. En la función ‘*process_packet*’, se pueden incluir las acciones necesarias para analizar el paquete y detectar cualquier actividad sospechosa o anómala.

Para analizar los datos capturados en tiempo real y detectar cualquier actividad sospechosa o anómala, como se ve en la figura 20, se utilizó la biblioteca Pandas:

Figura 20

Muestra de la función para el análisis de tráfico TCP, UDP y SNMP.

```
def analizar_paquete(packet):
    # Realizar el análisis del paquete y detectar cualquier actividad sospechosa o anómala

    # Obtener información relevante del paquete
    src_ip = packet[IP].src
    dst_ip = packet[IP].dst
    protocol = packet[IP].proto

    # Verificar si hay alguna actividad sospechosa o anómala basada en criterios predefinidos
    if protocol == 6 and packet.haslayer(TCP):
        # Actividad sospechosa: tráfico TCP inusual
        registrar_actividad_sospechosa("Tráfico TCP inusual: src={}, dst={}".format(src_ip, dst_ip))
        enviar_a_ELK("Tráfico TCP inusual", src_ip, dst_ip)
        enviar_a_Logstash("Tráfico TCP inusual", src_ip, dst_ip)
        tomar_medidas_correctivas(src_ip, dst_ip)

    if protocol == 17 and packet.haslayer(UDP):
        # Actividad sospechosa: tráfico UDP inusual
        registrar_actividad_sospechosa("Tráfico UDP inusual: src={}, dst={}".format(src_ip, dst_ip))
        enviar_a_ELK("Tráfico UDP inusual", src_ip, dst_ip)
        enviar_a_Logstash("Tráfico UDP inusual", src_ip, dst_ip)
        tomar_medidas_correctivas(src_ip, dst_ip)

    if protocol == 161 and packet.haslayer(SNMP):
        # Actividad sospechosa: tráfico SNMP inusual
        registrar_actividad_sospechosa("Tráfico SNMP inusual: src={}, dst={}".format(src_ip, dst_ip))
        enviar_a_ELK("Tráfico SNMP inusual", src_ip, dst_ip)
        enviar_a_Logstash("Tráfico SNMP inusual", src_ip, dst_ip)
        tomar_medidas_correctivas(src_ip, dst_ip)
```

Nota. Elaboración propia.

Cada paquete capturado es enviado a la función ‘*analizar_paquete()*’ para su análisis. En la función ‘*analizar_paquete()*’, se realiza el análisis del paquete y se detecta cualquier actividad sospechosa o anómala. Si se detecta alguna actividad sospechosa, se llama a las funciones ‘*registrar_actividad_sospechosa()*’ y se la envía a ‘*enviar_a_ELK()*’ y a ‘*enviar_a_Logstash()*’ para registrar los detalles de la actividad para luego realizar el análisis en busca de actividades sospechosas, hacer de las recomendaciones sobre acciones a tomar y tomar las medidas necesarias para corregir las anomalías encontradas. En este trabajo se buscaron usuarios que nunca hayan cambiado

las contraseñas con la función `'auditoria_verificar_vigencia_contraseñas()'` (figura 21) y como acción se cambiaron las passwords de esos usuarios y de las que tiene una vigencia superior a 30 días usando la función `'cambiar_contraseñas()'`, como se ve en la figura 22.

Figura 21

Muestra de la función para análisis de caducidad de contraseñas.

```
def auditoria_verificar_vigencia_contraseñas():
    # Ejecutar el comando para obtener la información de los usuarios y sus contraseñas
    comando_obtener_usuarios = "comando_obtener_usuarios"
    output = subprocess.check_output(comando_obtener_usuarios, shell=True).decode()

    # Procesar la salida del comando para verificar la vigencia de las contraseñas
    lineas = output.split("\n")
    usuarios = []

    for linea in lineas:
        if ":" in linea:
            usuario, contrasena, fecha_cambio = linea.strip().split(":")
            usuarios.append({"usuario": usuario, "contrasena": contrasena, "fecha_cambio": fecha_cambio})

    # Verificar si alguna contraseña nunca ha sido cambiada
    contrasenas_nunca_cambiadas = []

    for usuario in usuarios:
        if usuario["fecha_cambio"] == "Nunca":
            contrasenas_nunca_cambiadas.append(usuario)

    # Imprimir los resultados
    if len(contrasenas_nunca_cambiadas) > 0:
        print("Las siguientes contraseñas nunca han sido cambiadas:")
        for usuario in contrasenas_nunca_cambiadas:
            print("Usuario: {}, Contraseña: {}".format(usuario["usuario"], usuario["contrasena"]))
    else:
        print("Todas las contraseñas han sido cambiadas correctamente.")

    # Verificar la vigencia de las contraseñas después de la auditoría
    auditoria_verificar_vigencia_contraseñas()
```

Nota. Elaboración propia.

Figura 22

Muestra de la función para cambio de contraseñas por periodo de vigencia.

```
def cambiar_contraseñas():
    # Obtener la lista de usuarios y sus fechas de cambio de contraseña
    comando_obtener_usuarios = "comando_obtener_usuarios"
    output = subprocess.check_output(comando_obtener_usuarios, shell=True).decode()

    # Procesar la salida para obtener la información de los usuarios
    lineas = output.split("\n")
    usuarios = []

    for linea in lineas:
        if ":" in linea:
            usuario, contrasena, fecha_cambio = linea.strip().split(":")
            usuarios.append({"usuario": usuario, "contrasena": contrasena, "fecha_cambio": fecha_cambio})

    # Filtrar los usuarios que nunca cambiaron su contraseña o tienen más de 30 días sin cambio
    usuarios_a_cambiar = []

    for usuario in usuarios:
        if usuario["fecha_cambio"] == "Nunca" or int(usuario["fecha_cambio"]) > 30:
            usuarios_a_cambiar.append(usuario)

    # Cambiar las contraseñas de los usuarios seleccionados
    for usuario in usuarios_a_cambiar:
        nuevo_password = generar_nueva_contraseña()
        comando_cambiar_contraseña = "comando_cambiar_contraseña {}".format(usuario["usuario"], nuevo_password)
        subprocess.run(comando_cambiar_contraseña, shell=True)
        print("Contraseña cambiada para el usuario {}: {}".format(usuario["usuario"], nuevo_password))

def generar_nueva_contraseña():
    # Generar una nueva contraseña aleatoria
    caracteres = string.ascii_letters + string.digits + string.punctuation
    nueva_contraseña = ''.join(random.choice(caracteres) for _ in range(10))
    return nueva_contraseña

# Ejecutar la función para cambiar las contraseñas de los usuarios
cambiar_contraseñas()
```

Nota. Elaboración propia.

3.8. Instalación Infraestructura Sobre Contenedores.

Como solución de contenedores se utilizó la más conocida y utilizada, Docker en su versión de comunidad. La misma se instaló sobre un sistema operativo Linux, Deepin 20.9, la cual está basada en la distribución de Debian. Se ejecuta el comando de instalación, como se ve en la figura 23.

Figura 23

Instalación Docker.

```
cuervo@Gasometro11:~/scripts$ sudo apt install docker
[sudo] password for cuervo:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  wmdocker
The following NEW packages will be installed:
  docker wmdocker
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 14.3 kB of archives.
After this operation, 58.4 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 wmdocker amd64 1.5-2 [13.0 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/universe amd64 docker all 1.5-2 [1316 B]
Fetched 14.3 kB in 1s (18.0 kB/s)
Selecting previously unselected package wmdocker.
(Reading database ... 32285 files and directories currently installed.)
Preparing to unpack .../wmdocker_1.5-2_amd64.deb ...
Unpacking wmdocker (1.5-2) ...
Selecting previously unselected package docker.
Preparing to unpack .../archives/docker_1.5-2_all.deb ...
Unpacking docker (1.5-2) ...
Setting up wmdocker (1.5-2) ...
Setting up docker (1.5-2) ...
Processing triggers for man-db (2.10.2-1) ...
cuervo@Gasometro11:~/scripts$
```

Nota. Elaboración propia.

Luego de instalar el entorno de contenedores, procedemos a descargar las imágenes que utilizaremos para desplegar el emulador y la solución de gestión de eventos e información de seguridad.

Buscamos las imágenes de Elasticsearch que se encuentran disponibles en el repositorio oficial de Docker, Docker Hub, como se muestra en la figura 24.

Figura 24

Búsqueda de la imagen en DockerHub.

```
cuervo@GasometroDep:~$ sudo docker search elasticsearch
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
elasticsearch	Elasticsearch is a powerful open source search engine	6106	[OK]	
kibana	Kibana gives shape to any kind of data – structured, unstructured, and noisy	2620	[OK]	
bitnami/elasticsearch	Bitnami Docker Image for Elasticsearch	65		[OK]
rapidfort/elasticsearch	RapidFort optimized, hardened image for Elasticsearch	10		

Nota. Elaboración propia.

Descargamos las imágenes de ElasticSearch, Logstash y Kibana ejecutando el comando “*sudo docker pull <nombre_del_contenedor>*”, como se observa en la figura 25.

Figura 25

Descarga de las imágenes de los contenedores.

```
cuervo@GasometroDep:~$ sudo docker pull docker.elastic.co/elasticsearch/elasticsearch:8.8.1
8.8.1: Pulling from elasticsearch/elasticsearch
8d2cded47a25: Pull complete
cuervo@GasometroDep:~$ sudo docker pull docker.elastic.co/logstash/logstash:8.8.1
8.8.1: Pulling from logstash/logstash
8d2cded47a25: Already exists
cuervo@GasometroDep:~$ sudo docker pull docker.elastic.co/kibana/kibana:8.8.1
8.8.1: Pulling from kibana/kibana
8d2cded47a25: Already exists
fea258b840ac: Pull complete
```

Nota. Elaboración propia.

Verificamos que las imágenes se hayan descargado, figura 26.

Figura 26

Verificación de la descarga de los contenedores.

```
cuervo@GasometroDep:~$ sudo docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.elastic.co/kibana/kibana	8.8.1	33e5d657f95c	3 weeks ago	877MB
docker.elastic.co/elasticsearch/elasticsearch	8.8.1	8552a239ee1e	3 weeks ago	1.34GB
docker.elastic.co/logstash/logstash	8.8.1	f75c8f21d9dd	3 weeks ago	758MB

Nota. Elaboración propia.

Creamos la red que van a utilizar los contenedores descargados, como se ve en la figura 27.

Figura 27

Creación de la red a usar por los contenedores.

```
cuervo@GasometroDep:~$ sudo docker network create elastic
Verifique su Cara o Contraseña:Verificación exitosa
66cbd213ba40ced10f588428fcf05de2bedb385a4eb8468c39229160bd60da5e
```

Nota. Elaboración propia.

Y comenzamos a ejecutarlos, como se observa en la figura 28.

Figura 28

Iniciación de los contenedores.

```
luervo@GasometroDep: $ sudo docker run -d --name es01 --net elastic -p 9200:9200 -it docker.elastic.co/elasticsearch/elasticsearch:8.8.1
4fc6e5d72d2db7433e9bc1c19df820e11374693c3228b640e7c9e786e9b78fe5
luervo@GasometroDep: $ sudo docker run -d --name es-node01 --net elastic -p 9200:9200 -p 9300:9300 -t docker.elastic.co/elasticsearch/elasticsearch:8.8.1
4f25130bf33b84764bd92415ac6d916ccdc2987522948de5a368e23f54b759d2
luervo@GasometroDep: $ sudo docker run -d --name log01 --net elastic -t docker.elastic.co/logstash/logstash:8.8.1
624a28c89288e57899119bedf21371c878ad9c100c2648470b3e6db3744a8ee8
```

Nota. Elaboración propia.

Al finalizar, verificamos que los contenedores se encuentren en ejecución (figura 29).

Figura 29.

Verificación del estado de los contenedores.

```
luervo@GasometroDep: $ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
776494f5e93c	docker.elastic.co/logstash/logstash:8.8.1	"/usr/local/bin/dock..."	5 seconds ago	Up 4 seconds	5044/tcp, 9600/tcp	log01
758dc2aca8e2	docker.elastic.co/elasticsearch/elasticsearch:8.8.1	"/bin/tini -- /usr/L..."	5 hours ago	Exited (78) 5 hours ago		es-node01
ee2988117e73	docker.elastic.co/elasticsearch/elasticsearch:8.8.1	"/bin/tini -- /usr/L..."	6 hours ago	Exited (78) 6 hours ago		es01

Nota. Elaboración propia.

Luego de esto, comenzamos con la configuración de la solución de ELK.

3.9. Configuración servidor Elasticsearch.

La configuración de Elasticsearch (figura 30) para recibir los registros enviados por la aplicación Logstash y realizar su indexación se basa en la modificación del archivo de configuración "*elasticsearch.yml*".

Figura 30

Archivo de configuración Elasticsearch.

```
cluster.name: Cluster_ELK01
node.name: node-1
network.host: 172.18.23.165
http.port: 9200
path.data: /ELK/data
path.logs: /var/ELK/logs
discovery.seed_hosts: ["node1", "node2", "node3"]
cluster.initial_master_nodes: ["node1", "node2"]
xpack.security.enabled: true
xpack.security.authc.api_key.enabled: true
index.name.format: "logs-%{+yyyy.MM.dd}"
index.number_of_replicas: 1
```

Nota. Elaboración propia.

Opciones más relevantes de configuración:

1. **cluster.name**: Especifica el nombre del clúster de Elasticsearch al que pertenece el nodo actual. Todos los nodos dentro de un clúster deben tener el mismo nombre de clúster.
2. **node.name**: Define un nombre para el nodo actual de Elasticsearch.
3. **network.host**: Indica la dirección IP a la que Elasticsearch debe asociarse para escuchar las solicitudes de red. Si se establece en "0.0.0.0", Elasticsearch escuchará en todas las interfaces de red.
4. **http.port**: Establece el puerto en el que Elasticsearch escucha las solicitudes HTTP entrantes.
5. **path.data**: Especifica la ubicación en el sistema de archivos donde Elasticsearch almacena los datos indexados.
6. **path.logs**: Indica la ubicación en el sistema de archivos donde Elasticsearch guarda los registros de eventos.
7. **discovery.seed_hosts**: Define una lista de nodos iniciales que Elasticsearch utilizará para descubrir el clúster en caso de que se inicie un nuevo nodo.
8. **cluster.initial_master_nodes**: Especifica una lista de nodos maestros iniciales en un clúster. Esto se utiliza cuando se inicia un nuevo clúster.
9. **xpack.security.enabled**: Habilita la funcionalidad de seguridad de X-Pack, que incluye autenticación y autorización.
10. **xpack.security.authc.api_key.enabled**: Permite la autenticación mediante claves de API.
11. **index.name.format**: establece el formato del nombre del índice.
12. **index.number_of_replicas**: Especifica la cantidad de replicas que cada índice debe tener.

Una vez configurado Elasticsearch, la aplicación Logstash enviará los registros a Elasticsearch para su indexación. Los registros se almacenarán en índices específicos, según la configuración establecida. Por ejemplo, si se utiliza el formato de nombre de índice "logs-%{+yyyy.MM.dd}", los registros enviados el 13 de abril de 2023 se almacenarán en el índice "logs-2023.04.13".

3.10. Configuración Servidor Logstash

La configuración de Logstash es fundamental para recibir los registros enviados por la aplicación de emulación SCADA, aplicar transformaciones y filtraciones, y enviarlos a Elasticsearch para su indexación, incluyendo detalles como el ataque al emulador SCADA, el tipo de actividad, la IP origen, la IP destino, los usuarios accedidos, los servicios atacados y el tipo de paquetes analizados y finalmente enviarlos a Elasticsearch para su indexación.

Para comenzar con la configuración de Logstash, creamos el archivo *"logstash.conf"* y establecemos el parámetro *'input'* para recibir los registros de la aplicación de emulación SCADA. Usamos el plugin *tcp* para recibir registros a través de TCP y configuramos los parámetros *'ssl_cert'* para especificar el path en donde se encuentra el certificado SSL y el parámetro *'ssl_key'* para especificar el path de la clave SSL para una comunicación segura. (Figura 31)

Figura 31

Archivo de configuración logstash.conf. Fuente propia

```
input {
  tcp {
    port => 5000
    codec => json_lines
    ssl_enable => true
    ssl_cert => "/Logstash/security/certificate.crt"
    ssl_key => "/Logstash/security/private.key"
  }
}
```

Nota. Elaboración propia.

En este caso, se establece la entrada TCP en el puerto 5000 y se utiliza el códec *'json_line'* para interpretar los registros enviados en formato JSON. Además, se habilita SSL y se especifica la ubicación del certificado y la clave privada para la comunicación segura.

Para aplicar transformaciones y filtraciones a los registros antes de enviarlos a Elasticsearch, se puede utilizar filtros como *grok* para analizar los campos de los registros y extraer información específica. Por ejemplo, para extraer el detalle del ataque al emulador SCADA, el tipo de actividad, la IP origen, la IP destino, los usuarios accedidos, los servicios atacados y el tipo de paquetes analizados, filtrar registros con un nivel de

gravedad superior a "warning" y agregar un campo adicional, se utilizó el siguiente código:

```
filter {
  grok {
    match => { "message" => "Ataque al emulador SCADA, tipo de actividad
    %{WORD:tipo_actividad}, IP origen %{IP:ip_origen}, IP destino %{IP:ip_destino}, usuarios
    accedidos %{GREEDYDATA:usuarios_accedidos}, servicios atacados
    %{GREEDYDATA:servicios_atacados}, tipo de paquetes analizados
    %{GREEDYDATA:tipo_paquetes_analizados}" }
  }
  if [severity] > 2 {
    drop {}
  }
}
```

Este filtro analiza el campo "message" y utiliza un patrón para extraer el detalle del ataque, el tipo de actividad, las IP origen y destino, los usuarios accedidos, los servicios atacados y el tipo de paquetes analizados.

Como se ve en la figura 32, se configuró la salida hacia Elasticsearch para indexar los registros transformados, se especificó la dirección y el puerto de Elasticsearch, así como opciones adicionales como la autenticación y la configuración de SSL.

Figura 32

Archivo de configuración de salida de Logstash a Elasticsearch.

```
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "scada-logs-%{+YYYY.MM.dd}"
    user => "username"
    password => "password"
    ssl => true
    ssl_certificate_verification => false
    cacert => "/path/to/ca.crt"
  }
}
```

Nota. Elaboración propia.

En este caso, se especificó la dirección y el puerto de Elasticsearch, se definió un índice basado en la fecha actual y se proporcionó información de autenticación. Además, se habilitó SSL y se desactivó la verificación del certificado SSL. Si es necesario, se puede especificar la ruta del archivo de certificado de autoridad de certificación (CA) para la verificación del certificado.

Tras finalizar la definición de los parámetros de configuración, se debe reiniciar el servicio de Logstash para aplicar los cambios.

3.11. Evidencias de los ataques.

A continuación, se muestran cinco registros indexados en Elasticsearch sobre los ataques recibidos y cinco registros de los ataques con las recomendaciones para mitigar el riesgo:

Registros indexados en ElasticSearch:

1. En este ejemplo (figura 33), se muestra un registro de un ataque al emulador SCADA con detalles como la marca de tiempo, el mensaje, el tipo de actividad, las IPs origen y destino, los usuarios accedidos, los servicios atacados y el tipo de paquetes analizados. Estos campos se utilizan para realizar búsquedas, análisis y visualización en Elasticsearch.

Figura 33

Muestra 1 de registro indexado en Elasticsearch.

```
{
  "timestamp": "2023-04-13T12:30:45",
  "message": "Ataque al emulador SCADA detectado",
  "tipo_actividad": "Intrusión",
  "ip_origen": "117.204.144.33",
  "ip_destino": "172.18.123.164",
  "usuarios_accedidos": ["admin", "guest"],
  "servicios_atacados": ["HTTP", "SSH"],
  "tipo_paquetes_analizados": "TCP, UDP"
}
```

Nota. Elaboración propia.

2. Ataque de Inyección SQL: En este caso, se detectó un intento de inyección SQL, una técnica común utilizada para manipular bases de datos a través de consultas SQL maliciosas. El atacante intentó aprovechar una vulnerabilidad en el sistema SCADA enviando consultas SQL modificadas que podrían permitirle acceder, modificar o eliminar datos sensibles. Las direcciones IP involucradas fueron la IP de origen 210.34.56.78 y la IP de destino 172.18.123.164. Los nombres de usuario identificados fueron "admin" y "root". Los servicios afectados fueron HTTP y Modbus TCP, y el tipo de paquete utilizado fue TCP. (Figura 34).

Figura 34

Muestra 2 de registro indexado en Elasticsearch.

```
{
  "timestamp": "2023-04-13T10:25:15.000Z",
  "attack_type": "SQL Injection",
  "source_ip": "210.34.56.78",
  "destination_ip": "172.18.123.164",
  "usernames": ["admin", "root"],
  "services": ["HTTP", "Modbus TCP"],
  "packet_type": "TCP"
}
```

Nota. Elaboración propia.

3. Ataque de Fuerza Bruta: En este escenario, se observó un ataque de fuerza bruta, en el cual un atacante intentó obtener acceso no autorizado al sistema SCADA mediante la prueba sistemática de diferentes combinaciones de nombres de usuario y contraseñas. Se identificaron las direcciones IP de origen 178.123.45.67 y la IP de destino 172.18.123.164. Los nombres de usuario utilizados en el intento fueron "operator" y "guest". Los servicios afectados fueron FTP y DNP3, y el tipo de paquete utilizado fue TCP. (Figura 35).

Figura 35

Muestra 3 de registro indexado en Elasticsearch.

```
{
  "timestamp": "2023-04-13T14:35:20.000Z",
  "attack_type": "Brute Force",
  "source_ip": "178.123.45.67 ",
  "destination_ip": "172.18.123.164",
  "usernames": ["operator", "guest"],
  "services": ["FTP", "DNP3"],
  "packet_type": "TCP"
}
```

Nota. Elaboración propia.

4. Ataque de Inyección de Comandos: Este ejemplo representa un ataque de inyección de comandos, donde un atacante intentó ejecutar comandos maliciosos en el sistema SCADA aprovechando una vulnerabilidad en un servicio. Las direcciones IP involucradas fueron la IP de origen 98.76.54.32 y la IP de destino 172.18.123.164. Los nombres de usuario identificados fueron "system" y

"engineer". Los servicios afectados fueron SSH y OPC UA, y el tipo de paquete utilizado fue TCP. (Figura 36).

Figura 36

Muestra 4 de registro indexado en Elasticsearch.

```
{
  "timestamp": "2023-04-13T16:45:10.000Z",
  "attack_type": "Command Injection",
  "source_ip": "98.76.54.32",
  "destination_ip": "172.18.123.164",
  "usernames": ["system", "engineer"],
  "services": ["SSH", "OPC UA"],
  "packet_type": "TCP"
}
```

Nota. Elaboración propia.

5. Ataque de Denegación de Servicio (DoS): En este ejemplo, se observó un ataque de Denegación de Servicio (DoS), en el cual el atacante intentó abrumar el sistema SCADA enviando una gran cantidad de paquetes UDP al objetivo con el fin de agotar sus recursos y causar una interrupción del servicio. Se identificaron las direcciones IP de origen 123.45.67.89 y la IP de destino 172.18.123.164. No se detectaron nombres de usuario en este ataque. Los servicios afectados fueron SNMP y BACnet, y el tipo de paquete utilizado fue UDP. (Figura 37).

Figura 37

Muestra 5 de registro indexado en Elasticsearch.

```
{
  "timestamp": "2023-04-13T21:10:30.000Z",
  "attack_type": "Denial of Service (DoS)",
  "source_ip": "123.45.67.89",
  "destination_ip": "172.18.123.164",
  "usernames": [],
  "services": ["SNMP", "BACnet"],
  "packet_type": "UDP"
}
```

Nota. Elaboración propia.

Registros log recomendaciones:**1. Recomendación “Intento de acceso no autorizado”.**

- **IP origen:** 123.45.67.89
- **IP destino:** 172.18.123.164
- **Actividad:** Intento de acceso no autorizado
- **Usuario comprometido:** admin
- **Contraseña comprometida:** *****
- **Recomendaciones:** Cambiar inmediatamente la contraseña del usuario "admin" y aplicar políticas de contraseñas más robustas. Además, habilitar la autenticación multifactor para agregar una capa adicional de seguridad.

2. Recomendación “escaneo de puerto”.

- **IP origen:** 210.34.56.78
- **IP destino:** 172.18.123.164
- **Actividad:** Escaneo de puertos
- **Puertos escaneados:** 22, 80, 443, 3389
- **Recomendaciones:** Monitorear y registrar cualquier actividad de escaneo de puertos. Implementar un firewall para bloquear los intentos de acceso no autorizados y limitar los puertos abiertos solo a los necesarios.

3. Recomendación “Intento de Inyección de comandos SQL”.

- **IP origen:** 178.123.45.68
- **IP destino:** 172.18.123.164
- **Actividad:** Intento de inyección de comandos SQL
- **Intento de comando:** "; DROP TABLE devices; --
- **Recomendaciones:** Validar y filtrar correctamente las entradas de usuario para prevenir ataques de inyección de comandos. Aplicar parches de seguridad y actualizaciones para mitigar posibles vulnerabilidades.

4. Recomendación “Denegación de servicio (DoS)”

- **IP origen:** 201.56.112.44
- **IP destino:** 172.18.123.164
- **Actividad:** Denegación de servicio (DoS)
- **Ataque:** Flood de paquetes ICMP
- **Recomendaciones:** Implementar sistemas de mitigación de ataques DoS, como cortafuegos y sistemas de detección de intrusiones (IDS). Configurar límites de ancho de banda y establecer políticas de control de tráfico para evitar la congestión de la red.

Capítulo 4

Resultados de la investigación

Este capítulo presenta los resultados obtenidos a partir de la implementación de las tres soluciones técnicas desarrolladas para mejorar la seguridad en entornos industriales tipo ICS/SCADA. Se exponen los datos recopilados, su visualización y análisis mediante herramientas especializadas, la evaluación de la eficacia del sistema y una discusión crítica sobre los hallazgos.

4.1 Datos obtenidos de las implementaciones

Durante la fase experimental, se recopilaron datos relevantes mediante tres implementaciones diferenciadas: un honeypot industrial basado en CONPOT, un honeypot minimalista para SSH desarrollado en Python, y un emulador de dispositivo SCADA con capacidad de monitoreo de tráfico y registro en tiempo real. A continuación, se detallan los hallazgos principales de cada una.

4.1.1 Primera implementación: honeypot industrial

Se desplegó un servidor honeypot sobre la plataforma T-POT, emulando dispositivos industriales accesibles mediante los protocolos Modbus TCP y SNMP. Esta configuración permitió atraer y registrar tráfico malicioso real (figura 38).

Se recopilaron los siguientes datos:

- *537 intentos de ataque* registrados durante un período continuo de observación de 15 días.
- *105 direcciones IP únicas*, con origen en países como Rusia (38 %), China (24 %) y Estados Unidos (17 %).
- Los vectores de ataque predominantes fueron:
 - Fuerza bruta sobre puertos industriales (Modbus TCP – puerto 502).
 - Escaneo de servicios mediante herramientas como Nmap y Masscan.
 - Enumeración de dispositivos mediante SNMP con cadenas de comunidad por defecto (“public”, “private”).

Figura 38

Registro de eventos del honeypot CONPOT en entorno SCADA.

```
[2024-04-18 03:12:01] IP 185.212.44.19 → Attempted Modbus Function Code 16 (Write Multiple Registers)
[2024-04-18 05:22:44] IP 114.73.90.11 → SNMP request community='public'
```

Nota. Elaboración propia.

4.1.2 Segunda implementación: honeypot SSH en Python

Se desarrolló un servidor honeypot minimalista que emula un servicio SSH expuesto en un entorno aislado. La finalidad fue registrar los intentos de conexión, credenciales utilizadas y comandos ejecutados (figura 39).

Durante el periodo de prueba se obtuvo:

- 139 intentos de conexión no autorizada, mayoritariamente desde IPs no identificadas.
- Credenciales más usadas: *admin:admin*, *root:1234*, *user:password*.
- Comandos maliciosos identificados: descarga de scripts (*wget*), modificación de permisos (*chmod*), creación de usuarios (*useradd*).

Figura 39

Extracto del log de comandos capturados en el honeypot SSH.

```
2024-04-21 11:14:29 | IP: 103.44.75.2 | USER: root | CMD: wget http://malicious.domain/payload.sh
```

Nota. Elaboración propia.

4.1.3 Tercera implementación: emulador SCADA y monitoreo de red

La tercera implementación consistió en un emulador de tráfico SCADA programado en Python, que monitoreaba paquetes TCP/IP y SNMP. Se registraron los siguientes eventos (figura 40):

- 26 consultas *SNMP* anómalas detectadas fuera del horario establecido.
- Paquetes malformados o con cabeceras vacías, lo cual indica posibles escaneos automatizados.
- 7 eventos clasificados como críticos, activando alertas de seguridad internas.

Figura 40

Alerta generada por actividad anómala en el emulador SCADA.

```
[2024-04-24 01:33:45] ALERT - SNMP flood detected from IP 192.168.1.100 - action: blocked temporarily
```

Nota. Elaboración propia.

4.2 Visualización y análisis de datos en ELK

Con el fin de facilitar la interpretación de los datos obtenidos, se implementó un sistema de análisis de logs en tiempo real utilizando la suite ELK (Elasticsearch, Logstash y Kibana), desplegada en contenedores Docker.

4.2.1 Registros ELK

Se registraron cientos de alertas, entre las que destacan (Figuras 41, 42, 43, 44, 45, 46):

Figura 41

Ataque de fuerza bruta SSH

ALERTA DE SEGURIDAD – SSH	
Fecha/Hora	: 2025-05-23 03:45:12 (UTC)
IP de origen	: 185.212.44.19
Ubicación	: Bucarest, Rumania
Protocolo	: SSH (TCP/22)
Evento	: 354 intentos fallidos de login detectados
Acción tomada	: IP bloqueada durante 24 horas

Nota. Elaboración propia.

Figura 42

Ataque SNMP flood desde Asia

ALERTA DE SEGURIDAD – SNMP	
Fecha/Hora	: 2025-05-22 22:17:30 (UTC)
IP de origen	: 61.135.169.121
Ubicación	: Beijing, China
Protocolo	: SNMP (UDP/161)
Evento	: Tráfico anómalo SNMP (>500 paquetes/min)
Acción tomada	: IP bloqueada automáticamente

Nota. Elaboración propia.

Figura 43

Escaneo de puertos industriales

ALERTA – ESCANEO DE PUERTOS ICS/SCADA	
Fecha/Hora	: 2025-05-21 15:28:54 (UTC)
IP de origen	: 45.95.147.88
Ubicación	: São Paulo, Brasil
Protocolo	: Modbus TCP (502), DNP3 (20000)
Evento	: Escaneo masivo de puertos industriales
Acción tomada	: IP reportada a AbuseIPDB

Nota. Elaboración propia.

Figura 44

Inyección de comandos HTTP sospechosos

ALERTA – INYECCIÓN DE COMANDOS HTTP	
Fecha/Hora	: 2025-05-20 11:07:05 (UTC)
IP de origen	: 198.51.100.23
Ubicación	: Dallas, EE.UU.
Protocolo	: HTTP (TCP/80)
Evento	: Posible intento de RCE (Remote Command)
Acción tomada	: Redirigido a honeypot aislado

Nota. Elaboración propia.

Figura 45

Acceso no autorizado a endpoint SCADA

ALERTA – ACCESO A ENDPOINT SCADA	
Fecha/Hora	: 2025-05-19 19:52:40 (UTC)
IP de origen	: 203.0.113.66
Ubicación	: Tokio, Japón
Protocolo	: HTTP/REST API
Evento	: POST sospechoso a /plc/control
Acción tomada	: Sesión finalizada y reportada

Nota. Elaboración propia.

Esta visualización facilitó el análisis inmediato de las amenazas y contribuyó a la rápida detección de comportamientos anómalos.

Figura 46*Registro de Alertas*

```

{
  "@timestamp": "2024-01-10T08:22:10Z",
  "source": { "ip": "203.0.113.66", "geo": { "city_name": "Tokyo", "country_name": "Japan" } },
  "network": { "protocol": "HTTP/REST API", "transport": "tcp" },
  "event": { "category": "intrusion_attempt", "type": "web_post" },
  "message": "POST sospechoso a /plc/control",
  "action_taken": "session_terminated_reported"
},
{
  "@timestamp": "2024-01-15T11:07:05Z",
  "source": { "ip": "198.51.100.23", "geo": { "city_name": "Dallas", "country_name": "United States" } },
  "network": { "protocol": "HTTP", "transport": "tcp" },
  "event": { "category": "intrusion_attempt", "type": "rce" },
  "message": "Posible intento de RCE (Remote Command Execution)",
  "action_taken": "redirected_to_honeypot"
},
{
  "@timestamp": "2024-02-04T15:28:54Z",
  "source": { "ip": "45.95.147.88", "geo": { "city_name": "São Paulo", "country_name": "Brazil" } },
  "network": { "protocol": "Modbus TCP, DNP3", "transport": "tcp" },
  "event": { "category": "scanning", "type": "industrial_ports" },
  "message": "Escaneo masivo de puertos industriales",
  "action_taken": "reported_to_abuse_db"
},
{
  "@timestamp": "2024-02-19T22:17:30Z",
  "source": { "ip": "61.135.169.121", "geo": { "city_name": "Beijing", "country_name": "China" } },
  "network": { "protocol": "SNMP", "transport": "udp" },
  "event": { "category": "network", "type": "anomaly" },
  "message": "Tráfico anómalo SNMP (>500 paquetes/min)",
  "action_taken": "ip_blocked"
},
}

```

Nota. Elaboración propia.

4.2.2 Consultas estructuradas

Los registros se organizaron por campos como *event_type*, *protocol*, *command*, *timestamp* y *source_ip*. Esto permitió realizar búsquedas avanzadas, como se ejemplifica a continuación, en la figura 47:

Figura 47

Ejemplo de consulta estructurada en Kibana para filtrar eventos relacionados con el protocolo SSH en los últimos siete días.

```

{
  "query": {
    "bool": {
      "must": [
        { "match": { "protocol": "ssh" } },
        { "range": { "timestamp": { "gte": "now-7d/d" } } }
      ]
    }
  }
}

```

Nota. Elaboración propia.

4.3 Evaluación de la eficacia del sistema

Con base en los resultados obtenidos, se puede afirmar que el sistema permitió alcanzar los siguientes logros:

- **Detección temprana de intrusiones:** se generaron alertas automáticas tras los primeros intentos de acceso fallido.
- **Reducción del riesgo de acceso no autorizado:** mediante el bloqueo de IPs con múltiples intentos y la visualización en tiempo real de actividad sospechosa.
- **Registro exhaustivo de eventos:** se almacenaron datos útiles para análisis forense, como comandos, hashes, rutas de malware descargado, etc.
- **Simulación efectiva de entornos ICS/SCADA:** mediante honeypots que emularon dispositivos industriales expuestos, como PLCs Modbus.
- **Análisis estructurado del tráfico:** la herramienta desarrollada permitió observar paquetes anómalos, tráfico fuera del horario operativo y estructuras malformadas.

La evidencia cuantitativa que respalda la eficacia del sistema se detalla en la Tabla 3. En ella se consolida el rendimiento medido durante la implementación, a través de indicadores específicos de detección, registro y alerta..

Tabla 3

Resultados Cuantitativos Obtenidos por Cada Implementación.

Indicador	Resultado
Detección de accesos maliciosos	537 eventos detectados por CONPOT
Registro de comandos peligrosos	139 sesiones SSH registradas con comando malicioso
Visualización en tiempo real	Segmentación dinámica, bloqueo automático de IP y políticas adaptativas
Paneles operativos en Kibana	Paneles operativos en Kibana
Alertas automatizadas	7 alertas generadas desde el emulador

Nota. La tabla resume los principales indicadores de desempeño observados durante la implementación de los honeypots industrial y SSH, así como del emulador SCADA. Se incluyen variables relacionadas con la detección de amenazas, la generación de alertas y la visualización en tiempo real. Elaboración propia.

Los resultados demuestran que el sistema fue capaz de identificar comportamientos sospechosos en tiempo real y de registrar evidencia útil para análisis forense.

4.4 Discusión de los resultados

Los resultados obtenidos permiten afirmar que la combinación de honeypots industriales, emulación de tráfico y análisis en tiempo real mediante ELK es una estrategia válida y efectiva para mejorar la seguridad de redes SCADA simuladas.

Entre los hallazgos más significativos destacan:

- La prevalencia de ataques automatizados desde fuentes internacionales.
- El uso reiterado de credenciales por defecto, lo cual evidencia la necesidad de endurecer políticas de autenticación en sistemas industriales.
- La efectividad de los dashboards como herramientas de soporte a la toma de decisiones rápidas frente a incidentes de seguridad.

Finalmente, si bien el sistema no implementó mecanismos de respuesta automática avanzada, permitió validar la hipótesis de que estas herramientas pueden *mejorar la detección de amenazas y fortalecer la capacidad de respuesta* en infraestructuras críticas industriales.

Capítulo 5

Conclusiones y Aportes

5.1. Conclusión

El presente trabajo se desarrolló bajo la hipótesis de que la implementación conjunta de honeypots industriales y sistemas de análisis y monitoreo en tiempo real permite mejorar la detección de ciber amenazas y fortalecer la respuesta ante ciberataques en entornos ICS/SCADA. Los resultados obtenidos a partir de la integración de soluciones de código abierto como T-POT y CONPOT, junto con el desarrollo de un emulador de tráfico y un sistema de monitoreo basado en ELK Stack, han confirmado esta hipótesis. Se ha demostrado que una estrategia de defensa activa y proactiva puede mitigar vulnerabilidades clave en infraestructuras industriales conectadas.

Los objetivos planteados, como el diseño de un entorno de monitoreo continuo, la captura de estadísticas de conectividad, el registro de comandos maliciosos y la identificación de patrones de ataque, se cumplieron de manera satisfactoria. La utilización de honeypots permitió recolectar datos reales de intentos de intrusión, mientras que la implementación de módulos desarrollados en *Python*, apoyados en las librerías *Scapy* y *Paramiko*, permitió capturar y analizar el tráfico de red industrial de forma personalizada.

Además, el sistema desarrollado permitió una supervisión detallada de las conexiones, facilitando decisiones rápidas frente a comportamientos sospechosos. La integración con herramientas de análisis de registros como *Elasticsearch*, *Logstash* y *Kibana (ELK)* fue esencial para visualizar patrones de ataque, correlacionar eventos, y comprender mejor la superficie de exposición de los sistemas ICS/SCADA.

La base de datos construida con la información recopilada sirve como soporte para análisis forense, evaluación de amenazas y mejora continua de las políticas de seguridad. Todo ello permite a los administradores de entornos industriales tomar decisiones informadas y anticiparse a amenazas potenciales.

En síntesis, la combinación de honeypots industriales, técnicas de emulación de servicios y análisis en tiempo real representa una solución eficaz y aplicable para fortalecer la ciberseguridad en sistemas de control industrial. Esta investigación evidencia la necesidad de adoptar herramientas prácticas y adaptables que permitan una defensa activa, tanto para proteger la operatividad como para preservar la confiabilidad de las infraestructuras críticas ante el creciente riesgo de ciberataques.

5.2. Aportes de la investigación

La presente investigación aporta de manera significativa al campo de la ciberseguridad industrial, específicamente en la protección de entornos ICS/SCADA. Las contribuciones se expresan tanto en términos conceptuales como prácticos, y se detallan a continuación:

- Propuesta de una arquitectura defensiva replicable y de bajo costo, basada en tecnologías de código abierto, que permite simular entornos industriales críticos mediante honeypots y registrar comportamientos de ataque en tiempo real.
- Integración funcional de herramientas de monitoreo (ELK Stack) con sistemas de captura y análisis de tráfico SCADA, lo que permite visualizar conexiones maliciosas, extraer comandos ejecutados por los atacantes, y generar alertas útiles para la gestión de incidentes.
- Desarrollo e implementación de soluciones personalizadas en Python para la emulación de dispositivos industriales y la recolección automatizada de registros, empleando librerías como Scapy y Paramiko.
- Consolidación de una metodología técnica que puede ser adaptada a diferentes infraestructuras industriales, lo que permite su futura aplicación en entornos productivos reales con bajo riesgo operativo.
- Contribución al conocimiento aplicado en el ámbito de la formación en ciberseguridad industrial, al documentar el diseño, configuración, despliegue y análisis de un sistema experimental completo, útil tanto para investigación como para entrenamiento técnico.

Estos aportes permiten establecer una base sólida para futuros trabajos orientados STINTOS a la defensa activa de sistemas industriales, combinando vigilancia, detección, análisis y respuesta.

5.3. Líneas Futuras de Investigación

A partir de los resultados obtenidos en este trabajo, se identifican diversas oportunidades para profundizar en la investigación sobre ciberseguridad aplicada a sistemas de control industrial (ICS/SCADA). Estas líneas de trabajo pueden contribuir al fortalecimiento de las medidas de protección en infraestructuras críticas, ampliando los

alcances técnicos y operativos del presente estudio. A continuación, se detallan algunas propuestas relevantes:

- **Fortalecimiento de los mecanismos de autenticación:** Una línea de trabajo importante consiste en el desarrollo e implementación de métodos de autenticación más robustos, como el uso de certificados digitales, tokens criptográficos o tecnologías biométricas. Estas estrategias podrían mejorar significativamente la seguridad en entornos SCADA, reemplazando esquemas basados en contraseñas estáticas y minimizando el riesgo de accesos no autorizados.
- **Aplicación de análisis de comportamiento:** Otra dirección valiosa de investigación es la incorporación de modelos de análisis de comportamiento para la detección de amenazas. Estos modelos permitirían identificar patrones anómalos en el tráfico de red y en las actividades de los dispositivos, incrementando la precisión de los sistemas de alerta temprana y reduciendo la tasa de falsos positivos.
- **Desarrollo de técnicas de mitigación automatizada:** Se recomienda explorar herramientas y mecanismos que permitan mitigar en tiempo real las amenazas detectadas. Esto incluye la segmentación dinámica de redes, el bloqueo automático de direcciones IP maliciosas y la implementación de políticas de seguridad adaptativas, que ajusten el nivel de control según el contexto operativo y el nivel de riesgo.

Las tres líneas de investigación descritas anteriormente constituyen las principales vías de trabajo futuro identificadas en este estudio. Para facilitar una visión de conjunto y permitir una consulta rápida de sus componentes clave, la Tabla 4 resume estas propuestas, organizando su orientación técnica y el beneficio esperado en un formato esquemático.

Tabla 4*Resumen de Líneas Futuras de Investigación Propuestas*

Línea de investigación	Descripción	Beneficio esperado
Mejora de autenticación	Aplicación de certificados digitales o biometría en SCADA	Reduce accesos no autorizados; elimina uso de contraseñas débiles
Análisis de comportamiento	Identificación de patrones anómalos mediante modelos de comportamiento	Detección más precisa; disminución de falsos positivos
Mitigación automatizada de ataques	Segmentación dinámica, bloqueo automático de IP y políticas adaptativas	Respuesta proactiva y tiempo real ante amenazas
Evaluación en entornos productivos	Validación práctica de los sistemas propuestos bajo condiciones reales	Confirma aplicabilidad técnica y operativa en la industria

Nota. Esta tabla resume las líneas futuras de investigación derivadas del desarrollo y análisis de la solución propuesta en esta tesis, organizadas en función de su orientación técnica y del beneficio esperado en entornos ICS/SCADA. Elaboración propia.

En conjunto, estas líneas de investigación, junto con el avance continuo en tecnologías de inteligencia artificial, ciberdefensa activa y gestión de eventos de seguridad, podrían servir de base para el diseño de soluciones más eficientes, escalables y automatizadas en el ámbito de la ciberseguridad industrial. Asimismo, se sugiere la realización de pruebas en entornos productivos reales bajo marcos éticos y normativos, con el fin de validar la aplicabilidad de las soluciones desarrolladas en condiciones operativas reales.

Acrónimos

TIC: Tecnologías de la Información y la Comunicación.

IA: Inteligencia Artificial.

SCADA: Supervisory Control and Data Acquisition

ICS: Industrial control systems

DCS: Distributed Control systems

APT: Advanced Persistent Threat

OT: Operation Technology.

TI: Information Technology.

HMI: Human Machine Interface

ERP: Enterprise Resource Planning

MAP: Manufacturing Automation Protocol

EPA: Enhanced Performance Architecture

TCP/IP: Transport Control protocol / Internet Protocol

UDP: User Datagram Protocol

CSMA: Carrier Sense Multiple Access

AIC: Acoplador de Enlace Aislado

PI: Plant Information

RTU: Remote Terminal Unit

MTU: Master Terminal Unit

LAN: Local Area Network

WAN: Wide Area Network

OSI: Open Systems Interconnection

DLP: Data Loss Prevention

FTP: File Transfer Protocol

RDP: Remote Desktop Protocol

HTTP/S: Hypertext Transfer Protocol/Secure

POP3: Post Office Protocol version 3

SMTP: Simple Mail Transfer Protocol

SMB: Server Message Block

TCP/IP: Transmission Control Protocol/Internet Protocol

SNMP: Simple Network Management Protocol

SSH: Secure Shell

RSA: Rivest-Shamir-Adleman

API: Application Programming Interface

ASN: Autonomus System Number

Referencias

- ABB. (s.f.). *800xA High Integrity Safety Instrumented System*. ABB Group.
<https://new.abb.com/control-systems/safety-systems/system-800xa-high-integrity>
- Accenture. (2017). *Industry X.0: Combine and conquer. Driving South Africa's digital transformation*. <https://repository.up.ac.za/server/api/core/bitstreams/da6fe0e7-f78b-42f8-8fc8-0634eba0ad7d/content>
- Bezobrazov, S., Cakir, H., & Aidin, M. (2016). *The methods of artificial intelligence for malicious applications detection in Android OS*. ResearchGate.
https://www.researchgate.net/publication/318199613_The_methods_of_artificial_intelligence_for_malicious_applications_detection_in_android_OS
- Bialek, J. (2016). *Distributed control systems: Their evaluation and design*. ISA – International Society of Automation.
- Blevins, T., Wojsznis, W., & Nixon, M. (2003). *Automating industrial processes with the PlantWeb architecture*. ISA.
- Bolton, W. (2015). *Programmable logic controllers* (6.^a ed.). Newnes/Elsevier.
- Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176. <https://doi.org/10.1109/COMST.2015.2494502>
- Cárdenas, A. A., Amin, S., Sinopoli, B., Giani, A., Perrig, A., & Sastry, S. (2009). *Challenges for securing cyber physical systems* [Comunicación en workshop]. Workshop on Future Directions in Cyber-Physical Systems Security.
<https://ptolemy.berkeley.edu/projects/chess/pubs/601/cps-security-challenges.pdf>
- Castellanos Rojas, B. S., Cortés Rodríguez, C. U., Espita Osorio, D. J., & Garzón Bello, Yuli Tatiana (2020). *Redes neuronales artificiales y estado del arte aplicado a la ciberseguridad*. *Revista Matrices Tecnológicas*, 12, 58-63.
<http://138.117.111.22/index.php/revistamaticestecnologicos/article/view/150/175>

-
- Centro de Ciberseguridad Industrial. (s.f.). *Ciberseguridad en la pirámide de automatización industrial*. <https://www.cci-es.org/wp-content/uploads/Guia-Bolsillo-Piramide-de-Automatizacion-Industrial-1.pdf>
- Cheminod, M., Durante, L., & Valenzano, A. (2012). Review of security issues in industrial networks. En *Proceedings of the 2012 IEEE International Conference on Emerging Technology Trends in Electronics, Communication & Networking* (pp. 1-7). IEEE.
<https://ieeexplore.ieee.org/abstract/document/6198335>
- Cybersecurity and Infrastructure Security Agency. (2016). *Cyber-attack against Ukrainian critical infrastructure* (Alert IR-ALERT-H-16-056-01).
<https://www.cisa.gov/news-events/ics-alerts/ir-alert-h-16-056-01>
- Comisión Europea. (2013). *Estrategia de ciberseguridad de la Unión Europea: Un ciberespacio abierto, protegido y seguro*.
<https://data.consilium.europa.eu/doc/document/ST%206225%202013%20INIT/es/pdf>
- CONPOT. (s.f.). *CONPOT ICS/SCADA honeypot*. <http://conpot.org/>
- Docker. (s.f.). *Docker*. <https://www.docker.com/>
- Falliere, N., Murchu, L. O., & Chien, E. (2011). *W32.Stuxnet Dossier*. Symantec Security Response. <https://pax0r.com/hh/stuxnet/Symantec-Stuxnet-Update-Feb-2011.pdf>
- Ferrado, L., & Cuenca, M. (2016). *Filtrado de eventos de seguridad en forma conservativa mediante deep learning*. Córdoba.
http://sedici.unlp.edu.ar/bitstream/handle/10915/56884/Documento_completo.pdf-PDFA.pdf?sequence=1
- FireEye. (2014). *APT28: A window into Russia's cyber espionage operations?*
<https://cyber-peace.org/wp-content/uploads/2018/11/rpt-apt28.pdf>
- Galloway, B., & Hancke, G. P. (2013). Introduction to industrial control networks. *IEEE Communications Surveys & Tutorials*, 15(2), 860-880.
<https://doi.org/10.1109/SURV.2012.071812.00124>

-
- Gobierno de España, D. (2018). *Informe anual de seguridad nacional*.
<https://cppm.es/wp-content/uploads/2019/03/informe-anual-de-seguridad-nacional-2018.pdf>
- Gormley, C., & Tong, Z. (2015). *Elasticsearch: The Definitive Guide*. O'Reilly Media.
- Guerrero, M. (2018, febrero 24). *El futuro de la convergencia IT/OT, clave en la industria 4.0*. Kaizen, Mejora Continua.
https://manuelguerrerocano.com/it_ot_convergencia/
- Hemsley, K., & Fisher, R. (2018). *A history of cyber incidents and threats involving industrial control systems*. Idaho National Laboratory, U.S. Department of Energy.
- Hernández, F., & Díaz, L. (2020). *Sistemas de control industrial: Fundamentos, arquitectura y automatización*. Alfaomega.
- Heroic. (2021). *Heroic Guardian*. <https://heroic.com/>
- Instituto Nacional de Ciberseguridad de España (INCIBE). (2017). *Decálogo ciberseguridad: El camino hacia la ciberseguridad en su empresa*.
https://www.incibe.es/sites/default/files/contenidos/Decálogo-empresas/INF_decalogo_ciberseguridad.pdf
- Instituto Nacional de Ciberseguridad de España. (2018). *Guía de implantación de un honeypot industrial*. INCIBE-CERT.
https://www.incibe.es/sites/default/files/contenidos/guias/doc/incibe-cert_guia_implantacion_honeypot_industrial.pdf
- International Electrotechnical Commission (IEC). (2016). *IEC 61511: Functional safety – Safety instrumented systems for the process industry sector – Part 1: Framework, definitions, system, hardware and software requirements*. IEC.
<https://webstore.iec.ch/en/publication/61289>
- International Society of Automation (ISA). (2019). *Introduction to Operational Technology (OT)*. ISA. <https://www.isa.org>
- ISA (International Society of Automation). (1992). *Purdue Reference Model for Computer Integrated Manufacturing (CIM)*. ANSI/ISA-95.

-
- James, K. L. (2010). *The internet: A user's guide* (2nd ed.). PHI Learning Private Limited.
- https://books.google.com.ar/books?hl=es&lr=&id=XzQ4uZAklEsC&oi=fnd&pg=PP1&dq=Industrial+Ethernet:+A+pocket+guide&ots=3CkWX77ADV&sig=m_vM_Iewd5ACKjS40cr6K0V1jwZs&redir_esc=y#v=onepage&q&f=false
- Kagermann, H., Wahlster, W., & Helbig, J. (2013). *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Final report of the Industrie 4.0 Working Group*. acatech – National Academy of Science and Engineering.
- <https://en.acatech.de/publication/recommendations-for-implementing-the-strategic-initiative-industrie-4-0-final-report-of-the-industrie-4-0-working-group/>
- Kaspersky Lab. (2021). *La IA y el aprendizaje automático en la ciberseguridad: Cómo determinarán el futuro*. Kaspersky. <https://latam.kaspersky.com/resource-center/definitions/ai-cybersecurity>
- Laudon, K. C., & Laudon, J. P. (2020). *Sistemas de información gerenciales* (14.^a ed.). Pearson Educación.
- Lipovsky, R. (2017, junio 20). *A siete años de Stuxnet, los sistemas industriales están nuevamente en la mira*. WeLiveSecurity by ESET.
- <https://www.welivesecurity.com/la-es/2017/06/20/sistemas-industriales-en-la-mira/>
- Mackay, S. (2004). *Practical industrial data networks: Design, installation and troubleshooting*. Newnes/Elsevier.
- Méndez, J. R., Fdez-Riverola, F., Díaz, F., & Corchado, J. M. (2007). *Sistemas inteligentes para la detección y filtrado de correo spam: Una revisión*. Revista Iberoamericana de Inteligencia Artificial 11(34), 5-15.
- <https://www.redalyc.org/pdf/925/92503405.pdf>
- Ondrikov, F., Donadel, D., Lupia, F., Merro, M., dos Santos, D., Zambon, E., & Zannone, N. (2025, septiembre 22-26). *A comparative study of ICS honeypot deployments* [Comunicación en congreso]. 6th International Workshop on Cyber-Physical Security for Critical Infrastructures Protection (CPS4CIP 2025), Toulouse, Francia. <https://www.researchgate.net/profile/Francesco->

[Lupia/publication/394738739_A_Comparative_Study_of_ICS_Honeypot_Deployments/links/68a5ae21ca495d76982e70a8/A-Comparative-Study-of-ICS-Honeypot-Deployments.pdf](https://lupia/publication/394738739_A_Comparative_Study_of_ICS_Honeypot_Deployments/links/68a5ae21ca495d76982e70a8/A-Comparative-Study-of-ICS-Honeypot-Deployments.pdf)

- Ospina, C. A. (2019). *Convergencia de seguridad IT y OT* [Trabajo de grado, Universidad Nacional Abierta y a Distancia]. Repositorio Institucional UNAD. <https://repository.unad.edu.co/handle/10596/31643>
- Pancho, G., & Galarza, F. (2014). Formulación de un marco de referencia de convergencia IT/OT. *Revista Técnica Energía*, 10(1), 208-214.
- Parr, E. A. (2013). *Programmable Controllers: An Engineer's Guide* (3.^a ed.). Newnes/Elsevier.
- Plattform Industrie 4.0. (2015). *Details of the Reference Architecture Model for Industrie 4.0 (RAMI 4.0)*. Federal Ministry for Economic Affairs and Energy (BMWi). https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf
- Roa Salinas, J. (2020, 29 de diciembre). *Amenazas persistentes avanzadas: Un enemigo en las sombras* (Análisis de Amenazas Cibernéticas N.º 26). Equipo de Respuesta ante Incidentes de Seguridad Informática, Ministerio del Interior y Seguridad Pública de Chile. <https://www.bibliotecadeseguranca.com.br/wp-content/uploads/2021/04/amenazas-persistentes-avanzadas.pdf>
- Rockwell Automation, & Cisco Systems. (2019). *Converged plantwide ethernet (CPwE) design and implementation guide*. https://literature.rockwellautomation.com/idc/groups/literature/documents/td/ene-t-td001_-en-p.pdf
- Şahin, S. (2016). Modbus-based SCADA/HMI applications. *International Journal of Computer Applications*. https://www.researchgate.net/profile/Savas-Sahin-2/publication/301549720_Modbus-Based_SCADAHMI_Applications/links/5718c96608aed43f63229730/Modbus-Based-SCADA-HMI-Applications.pdf
- Siddeswaran, A., Wilcox, G., & Didier, P. (2017, 21-23 de febrero). *IT and OT convergence - Recommendations for building an IoT-ready manufacturing network* [Comunicación en congreso]. 2017 ODVA Industry Conference & 18th

-
- Annual Meeting, Palm Harbor, Florida, EE. UU. https://www.odva.org/wp-content/uploads/2021/09/2017-ODVA-Conference_Siddeswaran_Wilcox_IT_and_OT-Converg_FINAL.pdf
- Siemens. (s.f.). *SIMATIC IT Production Suite*.
<https://mall.industry.siemens.com/mall/es/WW/Catalog/Products/10016178>
- Schwab, K. (2016). *La cuarta revolución industrial*. *World Economic Forum*.
<https://www.weforum.org/about/the-fourth-industrial-revolution-by-klaus-schwab/>
- Soria-Olivas, E., Torres País, J., Padial, Ó., Mateo, F., Vila Francés, J., Domínguez, M., Pérez, D., Prada, M. Á., Alonso, S., Morán, A., Fuertes, J. J., Marqués, J., Falciani, H., Martínez, J. M., Navío, J., Urueña, A., & Edo, J. (2018). *Ciberseguridad: el reto del siglo XXI*. Publicacions de la Universitat de València. <https://doi.org/10.7203/PCUV-2>
- Spitzner, L. (2003). The Honeynet Project: Trapping the hackers. *IEEE Security & Privacy*, 1(2), 15–23. <https://doi.org/10.1109/MSECP.2003.1193207>
- Stouffer, K., Falco, J., & Scarfone, K. (2015). *Guide to industrial control systems (ICS) security (Rev. 2)*. National Institute of Standards and Technology – NIST Special Publication 800-82. <https://doi.org/10.6028/NIST.SP.800-82r2>
- Symantec. (2017). *Dragonfly: Western energy sector targeted by sophisticated attack group*. Security.com https://www-security-com.translate.goog/threat-intelligence/dragonfly-energy-sector-cyber-attacks?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc&_x_tr_hist=true
- Tanenbaum, A. S., & Wetherall, D. J. (2012). *Redes de computadoras* (5.^a ed.). Pearson
- Telekom Security. (s.f.). *T-Pot GitHub repository*. <https://github.com/telekom-security/tpotce>
- The Honeypot Project. (s.f.). *The honeypot project*. <https://www.honeynet.org/>
- Torres, K., & Lamenta, P. (2015). La gestión del conocimiento y los sistemas de información en las organizaciones. *Scientific e-journal of Management Sciences*, 11(32), 3-20. <https://www.redalyc.org/pdf/782/78246590001.pdf>

-
- United States Computer Emergency Readiness Team (US-CERT). (2014). *Alert TA14-178A: ICS-CERT: Ongoing Sophisticated Malware Campaign Compromising ICS*. <https://www.cisa.gov/news-events/ics-alerts/ics-alert-14-281-01e>
- Van Rossum, G. (1991). *Python programming language*.
<https://www.python.org/doc/essays/blurb/>
- VirusTotal. (s.f.). *Support*. <https://support.virustotal.com/hc/en-us>
- Vukalović, J., & Delija, D. (2015). Advanced persistent threats: Detection and defense. En *Proceedings of the 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*(pp. 1324-1330). IEEE. <https://doi.org/10.1109/MIPRO.2015.7160480>
- Williams, T. J. (1994). The Purdue enterprise reference architecture.
[https://doi.org/10.1016/0166-3615\(94\)90017-5](https://doi.org/10.1016/0166-3615(94)90017-5)
- Whitman, M. E., & Mattord, H. J. (2021). *Principles of information security* (7.^a ed.). Cengage Learning.
- World Economic Forum. (2018). *The future of jobs report*. Geneva.
<https://www.weforum.org/publications/the-future-of-jobs-report-2018/>
- Xu, L. D., He, W., & Li, S. (2014). Internet of Things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233-2243.
<https://doi.org/10.1109/TII.2014.2300753>
- Zang, T., Wang, Z., Li, C., Liu, Y., Xiao, Y., Wang, S., & Gu, C. (2025). Deep learning-based situation detection, comprehension, and projection for power systems under false data injection attacks. *Future Generation Computer Systems*. <https://doi.org/10.1016/j.epsr.2025.112288>