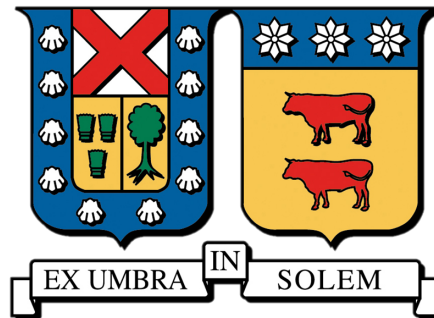


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



**Identificación de Riesgos de Ciberseguridad en Redes de
Servidores Mediante la Detección No Supervisada de
Anomalías.**

Benjamín Gabriel Pino Olivares

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO
CIVIL TELEMÁTICO**

PROFESOR GUÍA:

Mauricio Araya

PROFESOR CORREFERENTE:

Berioska Contreras

Diciembre 2024



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Identificación de Riesgos de Ciberseguridad en Redes de Servidores Mediante la Detección No Supervisada de Anomalías.

Nombre del candidato(a): [Benjamín Pino Olivares](#)

Carrera / Grado: Ingeniería Civil Telemática

Campus: Campus Casa Central Valparaíso Departamento: Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Mauricio Araya, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

X El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 01/12/2025 Firma:

Estudiante o Candidato(a):

Fecha: 01/12/2025 Firma:

Agradecimientos

Agradezco al equipo de seguridad del operador de telecomunicaciones nacional que propuso el desafío y nos apoyó en el desarrollo del proyecto. También, agradezco a mis compañeros de equipo Gonzalo Soto y Sebastián Jara por sus contribuciones.

Me gustaría agradecer de forma personal a mi profesor guía, Mauricio Araya, por su guianza en todo lo relacionado con minería de datos y ciencia de datos. Igualmente agradezco a Berioska Contreras por su tiempo y su dedicación a enseñar el campo de la seguridad en redes.

También, agradezco a mi familia: Nelson, Marisol, Javiera, Edilia y Nacor. Es gracias a su ayuda que he podido lograr todo lo que me he propuesto. Finalmente, agradezco a Yennifer, Sofía, Maria Jose y Catalina por su apoyo durante el desarrollo de esta carrera.

Resumen

Resumen - Esta investigación evalúa la eficacia de modelos de aprendizaje no supervisados, en particular Isolation Forest, al identificar casos de riesgo medio y alto en datos de detecciones de nodos de una red interna de un operador de telecomunicaciones en Chile, con el propósito de generar una mayor distinción entre casos de riesgo normal y casos de características similares a aquellos de riesgo alto. El modelo fue aplicado en conjunto con factores de riesgo relacionados al lado defensivo para crear una distinción entre casos de alto y bajo riesgo para casos de detecciones de vulnerabilidades individuales y vistas generales de activos. Se descubrió que el problema más frecuente de seguridad es aquel donde se detectan servicios con localizador de recursos uniformes (URLs) sospechosos, sucediendo en un 44 % de los casos detectados. En contraste, la configuración errónea de contraseñas tiene solo una ocurrencia del 15 % de los casos, pero es considerado uno de los casos más riesgosos. Si bien la herramienta obtuvo resultados favorables cuando se aplicó a datos de detecciones de red individuales, su desempeño fue menos preciso en el caso donde se evaluó el riesgo por activo de la red.

Palabras Clave: *Árboles de clasificación y regresión, Isolation Forest, Aprendizaje de máquinas, Seguridad en redes, Detección de intrusión/anomalía*

Abstract - This research evaluates the effectiveness of unsupervised learning models, in particular Isolation Forest, in identifying medium and high risk cases in node detection data from an internal network of a telecommunications operator in Chile, with the purpose of generating a greater distinction between normal risk cases and cases with characteristics similar to those of high risk. The model was applied in conjunction with risk factors related to the defensive side to create a distinction between high and low risk cases for individual vulnerability detection cases and asset overviews. It was discovered that the most frequent security problem is the one where services with suspicious Uniform Resource Locators (URLs) are detected, occurring in 44 % of the detected cases. In contrast, the configuration of incorrect passwords has only an occurrence in 15 % of the cases, but is considered one of the riskiest cases. While the tool performed favorably when applied to individual network detection data, its performance was less accurate when risk was assessed per network asset.

Keywords: *Classification and Regression Trees, Isolation Forest, Machine Learning, Network Security, Intrusion/Anomaly Detection*

Índice de figuras

3.1. Diagrama de arquitectura del sistema.	16
3.2. Prototipo de vista de inventario.	17
4.1. Comparación entre el Outlier Score entregado por IF y el riesgo calculado con la formula de riesgo actual	24
5.1. Efecto de IF en datasets con una contaminación del 3 %	28
5.2. Efecto de IF en datasets con una contaminación del 10 %	29

Índice de cuadros

4.1. Vectores de ataque considerados actualmente	20
4.2. Vectores incluidos en la medida de riesgo para activos	21
4.3. Ventajas de los métodos de detección de anomalías	22
4.4. Desventajas de los métodos de detección de anomalías	23
4.5. Peso por vector de ataque.	25
5.1. Mediciones de activos en la primera fuente de datos	28
5.2. Mediciones de activos en la segunda fuente de datos	29
6.1. Objetivos y riesgo abordado por vector	36

Índice general

1. Introducción	7
1.1. Objetivos	8
1.1.1. Objetivo General	8
1.1.2. Pregunta de investigación	8
1.1.3. Objetivos Específicos	9
2. Situación Actual	11
2.1. Pentesting	11
2.2. Aprendizaje de máquinas	12
3. Contexto de desarrollo	15
3.1. Trabajo previo	15
3.1.1. Diagrama de arquitectura	16
3.2. Componentes	16
3.2.1. Herramientas de detección	16
3.2.2. API y Frontend	17
3.2.3. Herramienta de Predicción de riesgo	17
3.3. Cálculo de riesgo	17
3.4. Datos	18
4. Diseño experimental	19
4.1. Objetivo	19

4.2. Esquema de datos	20
4.2.1. Selección de modelo	21
4.2.2. Parámetros	23
4.3. Evaluación de Riesgo	24
4.3.1. Riesgo evaluado por IF	24
4.3.2. Implementación	24
5. Análisis de Resultados	27
5.1. Discusión	29
5.2. Trabajo futuro	30
6. Conclusiones	33

Glosario

Open Web Application Security Project (OWASP): Proyecto de código abierto orientado a identificar causas de inseguridad en sistemas comunes.

NMAP: Herramienta de código abierto capaz de efectuar rastreo de puertos.

ZAP (Zed Attack Proxy): Herramienta de escaneo de OWASP para pruebas de penetración web.

SQLMAP: Herramienta especializada en automatizar el proceso de detectar vulnerabilidades de inyección SQL.

WPSCAN: Herramienta diseñada por WordPress para buscar vulnerabilidades en sitios web construidos por WordPress.

ACUNETIX: Plataforma comercial que ofrece escaneos de vulnerabilidades automatizados orientados a API's de empresas.

Invicti (Netsparker): Plataforma comercial que ofrece escaneos de vulnerabilidades automatizados orientados a API's de empresas.

Burp Suite: SHs de seguridad web de PortSwigger.

Aprendizaje supervisado: El aprendizaje supervisado, es un tipo de aprendizaje de máquinas que ocupa datos etiquetados para entrenar a un modelo con el propósito de clasificar elementos.

Aprendizaje no supervisado: El aprendizaje no supervisado, es un tipo de aprendizaje de máquinas que ocupa datos sin etiquetar para entrenar a un modelo con el propósito de agrupar elementos.

Redes neuronales: Modelos de aprendizaje basados en neuronas y capas que buscan emular el comportamiento del cerebro humano.

Datos tabulares: Datos que pueden ser presentados como una serie de columnas y filas en una tabla.

Capítulo 1

Introducción

Hoy en día existe una gran dificultad que enfrentan las empresas de todos los niveles y tamaños que es la dificultad de dar abasto para realizar todas las pruebas de seguridad a todos sus aplicativos e infraestructura para asegurarse de esta manera que no existan brechas de seguridad y puedan ser atacados y existan pérdidas millonarias tanto de nivel monetario, como de información. Por ejemplo: en la red interna de Entel, que posee sobre 25000 servidores en Chile¹, se concentran una cantidad significativa de servidores y aplicativos en su infraestructura que deben ser revisados. Debido a esto, no se logra completar la tarea de verificar la condición de cada activo dentro de un tiempo crítico donde se debe detectar una amenaza. Uno puede concluir que se requiere una herramienta que pueda obtener de forma eficiente las superficies de ataque existentes. Históricamente, la detección de amenazas se ha basado en enfoques reactivos, respondiendo a incidentes una vez que ya han ocurrido. Sin embargo, gracias a las nuevas tecnologías como la IA y el aprendizaje automatizado de máquinas, podemos avanzar hacia una estrategia preventiva y predictiva. Al correlacionar datos históricos y en tiempo real, estos sistemas pueden identificar patrones sutiles, anomalías y tendencias que podrían indicar la presencia de amenazas potenciales. Es fundamental comprender la importancia de esta innovación. No solo mejorará la eficiencia y eficacia de la respues-

¹<https://informacioncorporativa.entel.cl/>

ta ante incidentes de seguridad cibernética, sino que también proporcionará una capa adicional de defensa contra las amenazas cibernéticas emergentes, además de reducir la carga operativa al minimizar falsos positivos y automatizar procesos de análisis de datos complejos. Por otro lado, al aprovechar los avances tecnológicos recientes en IA y ML, podemos mejorar continuamente la precisión y la eficiencia de nuestros sistemas de detección. Las amenazas cibernéticas están en constante evolución, lo que requiere una solución adaptable y actualizable. Además, la solución debe integrarse sin problemas con la infraestructura existente para no interrumpir las operaciones y debe cumplir con las normativas y estándares de la industria para mantener la confidencialidad de los datos críticos. El uso de inteligencia artificial y aprendizaje automático (IA/ML) es llamativo en este enfoque. Estas tecnologías permiten la detección de anomalías mediante modelos de ML que identifican patrones inusuales en el tráfico de red. Además, los algoritmos de análisis predictivo pueden predecir posibles amenazas basándose en datos históricos y comportamientos conocidos. La automatización de respuestas es otra capacidad esencial, desarrollando sistemas que puedan tomar medidas preventivas y correctivas automáticamente cuando se detecten amenazas, lo que puede tener un impacto significativo en la rapidez de la respuesta en servicios esenciales o de infraestructura crítica como lo son las Telecomunicaciones.

1.1. Objetivos

1.1.1. Objetivo General

Obtener una visión del riesgo más precisa de la que existe actualmente en el sistema por medio del uso de herramientas de aprendizaje de máquinas.

1.1.2. Pregunta de investigación

Para esta investigación, nos interesa saber cuál es el efecto de los modelos no supervisados, en particular Isolation Forest, en la evaluación de riesgo de un sistema y

cómo puede incorporarse en un caso real.

1.1.3. Objetivos Específicos

- Identificar si es posible el uso de herramientas de aprendizaje de máquinas y, si este fuera el caso, identificar el modelo apropiado según las restricciones del ambiente y los requerimientos del sistema.
- Definir la estructura de los datos fuente que serán recolectados por el resto del sistema, considerando las distintas fuentes de información a ocupar.
- Usar el modelo o metodología de cálculo de riesgo elegida en datos de prueba. Revisar si el efecto es el deseado, sino volver a considerar el método ocupado.
- Medir el rendimiento del método ocupado en relación al método actual para el cálculo de riesgo. Comparando su costo contra su efectividad.

En el capítulo 2 se describe la situación actual en relación a métodos de Pentesting y aprendizaje de máquinas ocupados para análisis de riesgo; en el capítulo 3 se describe el contexto en el cual se desenvuelve el experimento considerando el trabajo previo que influenció el desarrollo; en el capítulo 4 se menciona el diseño experimental y la metodología implementada, en el capítulo 5 se analizan los resultados obtenidos por el experimento y sus posibles efectos en el sistema real, finalmente en el capítulo 6 se concluye el experimento.

Capítulo 2

Situación Actual

En este capítulo se describe el estado del arte y la situación actual referencia a métodos de detección de anomalías por medio de herramientas de aprendizaje de máquinas y las metodologías de pentesting que se aplicaron en el experimento.

2.1. Pentesting

La visión de la superficie de ataque de MITRE ATT&CK divide posibles vulnerabilidades en vectores de ataque, a los cuales se les puede categorizar y contrastar según su posible impacto en el sistema. Esta metodología ha resultado ser muy útil para pruebas de pentesting del lado ofensivo como defensivo ya que permite planificar alrededor de puntos conocidos[1].

El escaneo de vulnerabilidades es esencial para el proceso de pentesting, ya que permite recopilar información relevante sin alterar el sistema. Existen varias herramientas de uso abierto para recopilar información, entre ellas están las herramientas de NMAP, ZAP, SQLMAP y WPSCAN; como aquellas de uso comercial como ACUNETIX, In-victi y Burp Suite [2].

Para categorizar los riesgos conocidos, se utiliza como referencia el Open Web Application Security Project (OWASP). La calificación del riesgo según OWASP se calcula promediando la puntuación de los tres factores de probabilidad: prevalencia,

detectabilidad y explotabilidad; en una escala de 1 (bajo) a 3 (alto); luego, este valor se multiplica por el factor de impacto técnico, también en una escala de 1 (bajo o menor) a 3 (alto o crítico).[3].

2.2. Aprendizaje de máquinas

El aprendizaje de máquinas (ML) corresponde a un conjunto de técnicas y herramientas que permiten a un sistema de seguridad identificar patrones inusuales de comportamiento sin necesidad de que estas hayan sido programadas explícitamente. Considera técnicas como el aprendizaje supervisado, no supervisado y el uso de redes neuronales para evaluar el estado normal de una red y caracterizar aquellos casos que se desvían de este estándar. También, los árboles de decisión se pueden ocupar para dar prioridad a ciertas vulnerabilidades basándose en su severidad e impacto posible ocupando su historia como referencia[4].

Los árboles de decisión se refieren a métodos de aprendizaje de máquinas que se enfocan en separar los datos recursivamente en grupos cada vez más específicos. También se puede ocupar para separar grupos de acuerdo a aquellas características que los diferencian entre sí. Los algoritmos basados en árboles de decisión tienen la ventaja de tener una gran robustez ante datos de alta dimensionalidad y la capacidad de ocupar algoritmos de "pruning" que les permite reducir el número de grupos de acuerdo a alguna condición[5]. En estudios comparativos se ha encontrado que para datos tabulares, como aquellos ocupados frecuentemente para el análisis de seguridad, los árboles de decisión han sido capaces de sobrepasar la capacidad de clasificación de las redes neuronales en relación a su costo computacional relativo[6].

El Modelo de Mezcla Gaussianas (GMM) es una herramienta estadística basada en la suma ponderada de densidades de componentes gaussianos. Estos modelos tienen una variedad de aplicaciones diversas y pueden ser ajustados para resolver una variedad amplia de problemas. Su mayor debilidad es su linealidad intrínseca, lo que tiende a crear problemas de ajustes para datos no lineales. Para resolver este problema, se ha

propuesto el uso de Modelos de Mezcla Gaussiana Difusa (FGMM) que incorporan métodos basados en distancia y probabilidad[7].

Dentro de los métodos de aprendizaje de árboles de decisión existe un método conocido como Isolation Forest (IF) que se basa en detectar casos anómalos en vez de enfocarse en agrupar los datos por similitudes. El uso de aislamiento en vez de agrupamiento le permite a IF conseguir mejores resultados que aquellos conseguidos con otros algoritmos más costosos computacionalmente como Random Forest o métodos basados en distancia como Local Outlier Factor(LOF). IF también tiene el beneficio de funcionar de forma robusta incluso en casos de alta dimensionalidad e incluso en casos donde no existen datos anómalos dentro del grupo de entrenamiento[8].

La distancia de Mahalanobis es una métrica común ocupada para evaluar la similitud entre elementos de un grupo basándose en su distancia. Es ocupada de forma frecuente por algoritmos de clasificación y agrupamiento por su robustez al estimar la cercanía de elementos. Esto la vuelve una métrica ideal para descubrir la existencia de datos sobresalientes sin necesidad de entrenar un modelo. Por desgracia, en casos de alta dimensionalidad esta métrica tiende a fallar debido a las desviaciones más grandes que ocurren en esos datos[9].

Local Outlier Factor (LOF) es un algoritmo de detección de anomalías basado en densidad. Encuentra datos sobresalientes calculando la desviación local de un punto de datos dado. LOF es ocupado frecuentemente en minería de datos debido a su capacidad de detectar anomalías dentro de grupos localizados, aunque también posee una alta complejidad temporal. Se han propuesto soluciones compuestas que utilizan LOF en conjunto con Isolation Forest (IF) para detectar anomalías tanto en los grupos locales como en el grupo global, mitigando las debilidades de ambos algoritmos[10].

ABOD (Detección de Anomalías Basada en Ángulos) es un método para identificar valores atípicos en conjuntos que utiliza el ángulo entre los puntos en vez de la distancia entre los puntos, como se ocupa en métodos detección de anomalías por distancia tradicionales. Esto le ofrece una resistencia mayor a la llamada "maldición de la dimensionalidad", ya que no depende en que la selección de parámetros afecte la calidad de la

detección. Este modelo tiende a ofrecer mejores resultados que otros factores comunes como LOF, pero lo hace a cambio de un mayor costo computacional en aquellos casos donde el número de muestras incrementa[11].

La divergencia de Kullback-Leibler (KLD) es una medida utilizada para determinar la diferencia entre dos distribuciones de probabilidad por medio de un único número entero. Se ha demostrado en experimentos ocupando datos de señales que es útil para seleccionar anomalías bajo múltiples condiciones diversas. Incluso es capaz de conseguir buenos resultados cuando existen pocas muestras a las cuales analizar. Ya que no es un modelo tradicional de aprendizaje de máquinas sino que una medida matemática probabilística, no requiere mayores costos computacionales para implementar[12].

Las Redes Neuronales Auto-Organizadas (SOM) son un tipo de red neuronal definida por su arquitectura con forma de hoja de papel donde cada célula se especializa en detectar una entrada específica de datos. Esto es con el propósito de mejor simular el comportamiento de las neuronas encontradas en los cerebros de los animales. Esto las hace muy robustas para la detección de patrones complejos que ocupan señales ruidosas[13]. La detección de anomalías por medio de SOM se logra al detectar las neuronas que destacan dentro de la estructura o detectar datos que no entran a ninguna neurona. Estas neuronas se forman cuando existen concentraciones relativamente pequeñas de datos separados del resto de las muestras. Puede suceder entonces que un punto está separado y no introduce nuevas neuronas al sistema[14].

Para concluir, podemos decir con certeza que existe una amplia gama de métodos, modelos, métricas, etc. a nuestra disposición para poder resolver el problema de detección de anomalías y determinar su efecto sobre la percepción de riesgo.

Capítulo 3

Contexto de desarrollo

Este capítulo aborda el contexto del experimento. Se discute el trabajo previo realizado anteriormente por el equipo de seguridad de una empresa nacional de telecomunicaciones y se menciona la estructura básica del sistema al que pertenece nuestro modelo.

3.1. Trabajo previo

Para comenzar con el proyecto se nos ha presentado con un sistema básico de reconocimiento de vulnerabilidades de sondas llamado: Recon. Este sistema es capaz de detectar varios factores conocidos de vulnerabilidad, llamados Vectores de Ataque, para formar una Superficie de Ataque que representa la visión de riesgo acumulado en el sistem. Hasta ahora, el sistema ocupaba una forma simple para calcular el riesgo ocupando un factor de importancia o peso para cada Vector de ataque presente. La visión de riesgo de Recon esta enfocada únicamente a una dirección IP, un puerto y una fecha y hora específica, no considera en el cálculo de riesgo su relación con otras detecciones ni el comportamiento del Activo que puede contener múltiples direcciones IP y puertos.

3.1.1. Diagrama de arquitectura

El módulo que representa la herramienta de Machine Learning es representado en este diagrama (Figura 3.1). Este recibe la superficie de ataque, esto sería las detecciones guardadas en la base de datos por las herramientas de escaneo implementadas por el equipo hasta este momento.

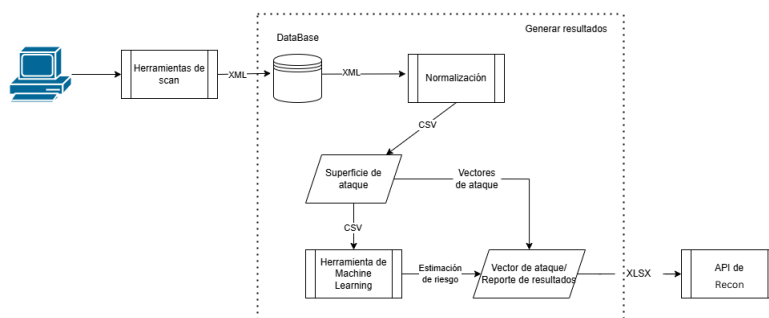


Figura 3.1: Diagrama de arquitectura del sistema.

3.2. Componentes

3.2.1. Herramientas de detección

Este módulo se basa en una serie de herramientas basadas alrededor de detecciones, que se organizan por indicadores de compromiso, específicamente pares e puerto y direcciones IP, que identifican puntos débiles en nodos de la red, consolidados como vectores de ataque siguiendo un esquema similar al ocupado por MITRE.

Las herramientas ocupadas para la detección de esta superficie de ataque se basan alrededor del resultado del reconocimiento o "fingerprint" de NMAP¹ que puede ser obtenido por medio de los parámetros que este ofrece. Estas mediciones se codifican en lo que nosotros conocemos como vectores de ataque.

¹<https://nmap.org/book/osdetect-fingerprint-format.html>

3.2.2. API y Frontend

Para poder visualizar los datos, se generó una API capaz de mostrar detecciones, activos y niveles de riesgo por medio de una interfaz simple de entender.

ID	NOMBRE DE SERVIDOR	NOMBRE DE USUARIO	IP	HOSTNAME	CPU	MEMORIA	DISCO	TAMAÑO	VECTORES DE ATAQUE	METADATOS
1	24.06	None			None	10			default_conditional_rule	
2	3.66	None			None	8			suspicious_url	
3	0	None			None	10			default_conditional_rule	
4	7.28	None			unknown	10			default_conditional_rule	

Figura 3.2: Prototipo de vista de inventario.

3.2.3. Herramienta de Predicción de riesgo

La Herramienta de Predicción de riesgo es un componente diseñado para identificar y evaluar posibles amenazas en una red interna de un operador de telecomunicaciones. Utiliza un algoritmo para asignar un valor de riesgo a cada detección y permite obtener una visión longitudinal del riesgo en el sistema. Maneja cantidades de datos grandes, considerando casi 25.000 sistemas que brindan servicio en Chile.

3.3. Cálculo de riesgo

El cálculo del riesgo para cada activo y detección se basa en una combinación ponderada de los vectores de ataque y las métricas de incidentes asociados. La fórmula empleada para calcular el riesgo es la siguiente:

$$R = \sum_{i=0}^n (P_n \cdot X_n) \quad (3.1)$$

Donde:

- P_n : Peso ponderado del vector de ataque n.

- X_n : Presencia del vector de ataque n , es 1 si existe y 0 si no.

La fórmula está diseñada para ser flexible, permitiendo incorporar nuevas métricas o ajustar los pesos en función de los requerimientos del sistema y los objetivos específicos del análisis de riesgos. El peso es determinado por el usuario del sistema dependiendo de sus necesidades. El umbral de riesgo mínimo varía, pero como regla general se considera una detección con una ponderación de alto riesgo al superar el 70 % de las otras mediciones.

3.4. Datos

Para realizar los experimentos, contamos con dos conjuntos de datos reales extraídos de las redes internas de la compañía mencionada anteriormente. El primer conjunto contiene 341 muestras obtenidas en 103 activos y el segundo contiene 317 muestras obtenidas en 108 activos distintos. Sin embargo, estos datos no están etiquetados, lo que representa un desafío significativo al momento de calcular métricas comunes como precisión y error. La ausencia de etiquetas dificulta la validación tradicional de los resultados obtenidos, ya que no se dispone de una referencia para determinar si las anomalías detectadas corresponden a eventos de alto riesgo.

Para concluir, podemos ver que este sistema maneja un volumen de datos grande, con nuestra muestra representando aproximadamente un 15 % de la tasa de detecciones diarias esperada en la herramienta Recon, con información tabular que permite envisionar el riesgo en el sistema. Nos interesa entonces mejorar este sistema por medio de la implementación de algoritmos de detección de anomalías sin aumentar demasiado la carga computacional del sistema.

Capítulo 4

Diseño experimental

En este capítulo se describe el diseño experimental, incluyendo la selección de un método de detección de anomalías y los vectores de ataque a considerar.

La herramienta funciona como un sistema predictivo que permite estimar la severidad de incidentes de seguridad y evaluar la posibilidad de riesgos futuros en la red. Esto se logra mediante la detección de anomalías y patrones inusuales en los datos recolectados, con el objetivo de anticipar vulnerabilidades o amenazas potenciales. El propósito principal es proporcionar una solución proactiva que permita a las organizaciones minimizar el impacto de incidentes de seguridad al abordar problemas antes de que se conviertan en brechas de datos o denegaciones del servicio.

4.1. Objetivo

El objetivo del proyecto es distinguir aquellos casos de alto riesgo en un sistema de redes internas ocupando un modelo de agrupamiento de aprendizaje de máquinas, e identificar casos limítrofes riesgosos que poseen características similares a los casos de alto riesgo pero que no son detectados por el sistema actual.

4.2. Esquema de datos

En el sistema actual de Recon se ocupa un esquema de vectores de ataque para definir la superficie de ataque del sistema. Estos vectores corresponden a vulnerabilidades comunes encontradas en sistemas de redes internas corporativas según el OWASP TOP 10 del 2021 a los cuales se le asocia un impacto máximo considerado por el equipo de seguridad (Cuadro 4.1). Basándose en el esquema ocupado por las partes del sistema en uso actualmente podemos definir cada punto de datos como una Detección. Esta corresponde a una medida de que vectores de riesgos predeterminados se pueden encontrar activos en ese nodo de la red, en un tiempo específico. Para poder obtener una visión completa del riesgo del sistema, se ha propuesto el uso de dos modelos distintos para calcular el riesgo a nivel de detección, esto sería obtener una medida de riesgo para cada detección obtenida, y calcular el riesgo a nivel del activo al que pertenecen las detecciones. Un activo se considera como un elemento de la red que puede poseer múltiples direcciones IP asociadas y tener incidentes reportados en relación a vulnerabilidades encontradas en el sistema.

Nombre vector	OWASP TOP 10 2021	Impacto
password_policy	A07:2021-Identification and Authentication Failures A02:2021-Cryptographic Failures	95
insecure_shared_folder	A05:2021-Security Misconfiguration	60
web_login	A04:2021-Insecure Design A03:2021-Injection A02:2021-Cryptographic Failures	35
suspicious_url	A05:2021-Security Misconfiguration A01:2021-Broken Access Control	20
default_credentials	A05:2021-Security Misconfiguration	75
possible_directory_traversal	A03:2021-Injection A01:2021-Broken Access Control	45
dbms	A05:2021-Security Misconfiguration	90

Cuadro 4.1: Vectores de ataque considerados actualmente

Se posee entonces como fuente de datos aquellas detecciones que obtienen los sensores del sistema Recon que está en uso actualmente y el uso de una planilla de inci-

dentes a los cuales miembros del equipo de seguridad agregan vulnerabilidades descubiertas en los activos. Tomamos entonces como estructura de datos para el modelo de detecciones la dirección IP de la detección, su puerto y las 6 columnas que representan la existencia o ausencia del vector de ataque correspondiente. Para el caso del modelo de activos, se ha decidido calcular cuál es la razón de la presencia de cada vector en cada detección asociada al activo, resultando en un número entre 0 y 1 y sumar el número de direcciones IP únicas y puertos únicos como campos. De la misma manera, para hacer uso de los datos de la planilla de incidentes se decidió agregar 5 métricas relacionadas con estos datos (Cuadro 4.2).

Vectores Nuevos	Definición	Impacto
num_incidents_active	Porcentaje de incidentes relacionados con el activo que están pendientes	30
num_incidents_mitigated	Porcentaje de incidentes relacionados con el activo que están mitigados	1
total_low_incidents	Porcentaje de incidentes pendientes de riesgo bajo	20
total_medium_incidents	Porcentaje de incidentes pendientes de riesgo medio	50
total_high_incidents	Porcentaje de incidentes pendientes de riesgo alto	80
total_critical_incidents	Porcentaje de incidentes pendientes de riesgo crítico	100

Cuadro 4.2: Vectores incluidos en la medida de riesgo para activos

4.2.1. Selección de modelo

Para calcular el riesgo de cada caso, se requirió un modelo capaz de agrupar casos particulares sin necesitar datos etiquetados para el entrenamiento debido a que etiquetar los datos con la ayuda de un experto sería muy costoso para el cliente. Con este propósito, se decidió ocupar Isolation Forest como herramienta ya que su funcionamiento es el que mejor se adapta a las características del desafío. Isolation Forest se centra exclusivamente en identificar puntos que son diferentes al resto de los datos,

en lugar de modelar patrones normales como otros algoritmos. Esto lo hace ideal para contextos de seguridad donde los valores atípicos pueden representar eventos críticos o amenazas, como detecciones inusuales o riesgos elevados. El rendimiento de Isolation Forest es robusto incluso en datos de alta dimensionalidad; esto es importante ya que las dimensiones actuales deben crecer si se implementan nuevos vectores de ataque a la superficie de ataque. A diferencia de otros modelos como las mezclas de gaussianas (GMM), Isolation Forest no asume distribuciones específicas en los datos, lo que lo hace más flexible para detectar anomalías en conjuntos heterogéneos y no lineales típicos de sistemas de seguridad. En el documento “A comparative evaluation of outlier detection algorithms: experiments and analyses.” (Pattern Recognition Vol. 74, 2018) [15] Se puede observar una comparación entre distintos métodos de identificación de datos que sobresalen de la norma, de los cuales nos interesan comparar aquellos más indicativos del comportamiento de cada clase de método (Cuadros 4.3 y 4.4).

Método	Ventajas
Gaussian Mixture Model (GMM)	- Uso de memoria bajo. - Tiempo de entrenamiento corto.
Isolation Forest	- Uso de memoria bajo. - Funciona bien independientemente de la distribución de los datos. - Tiempo de cálculo y entrenamiento corto.
Mahalanobis Distance Outlier Detection	- Ocupa pocos recursos computacionales. - Tiempo de cálculo y entrenamiento corto.
Local Outlier Factor (LOF)	- Adecuado para detectar anomalías locales.
Angle-Based Outlier Detection (ABOD)	- Eficiente para datasets de alta dimensión. - Tiempo de cálculo y entrenamiento corto.
Kullback-Leibler Divergence Outlier Detection (KL)	- Presenta resultados similares a GMM.
Redes Neuronales Auto-Organizadas (SOM)	- Puede detectar casos anomalos en datasets de distribución arbitraria. - En general, obtiene buenos resultados al medir su precisión.

Cuadro 4.3: Ventajas de los métodos de detección de anomalías

Método	Desventajas
Gaussian Mixture Model (GMM)	- No es útil cuando los datos siguen una distribución compleja.
Isolation Forest	- Menos eficaz para encontrar anomalías locales.
Mahalanobis Distance Outlier Detection	- Poca robustez ante datos de alta dimensionalidad.
Local Outlier Factor (LOF)	- No presenta buenos resultados en ambientes de alto ruido. - El numero de muestras afecta mucho su eficacia.
Angle-Based Outlier Detection (ABOD)	- Requiere de un mayor tiempo de cómputo cuando los datos son muy complejos - El numero de muestras afecta mucho su eficacia. - Poco robusto ante datos con alta dimensionalidad.
Kullback-Leibler Divergence Outlier Detection (KL)	- No presenta buenos resultados en datasets demasiado complejos. - No presenta buenos resultados en datasets con mucho ruido.
Redes Neuronales Auto-Organizadas (SOM)	- Costoso computacionalmente. - No se justifica su uso en casos más simples.

Cuadro 4.4: Desventajas de los métodos de detección de anomalías

4.2.2. Parámetros

Para obtener una visión amplia del efecto que tiene nuestro modelo sobre la percepción de riesgo en el sistema consideramos dos casos distintos. El primero involucra entrenar el modelo asumiendo que el número de detecciones y activos anormales representan aproximadamente 3 % de las muestras totales de la prueba, luego ampliamos este número a 10 % en el segundo caso. El resto de los parámetros se mantienen con los valores por defecto para simplificar el uso de la herramienta.

4.3. Evaluación de Riesgo

4.3.1. Riesgo evaluado por IF

En la siguiente figura, podemos ver una comparación simple entre los valores entregados de Outlier Score por Isolation Forest y por el valor de riesgo calculado con la fórmula original, ajustada a la misma escala. Podemos notar que a pesar de que los puntajes de Isolation Forest tienden a ser mayores, podemos notar que es capaz de identificar todos los casos de alto riesgo de la muestra y además seleccionar algunos casos no detectados con la fórmula.

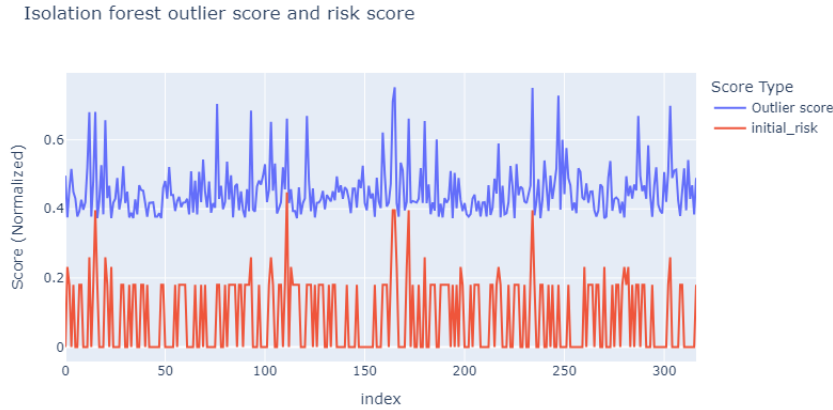


Figura 4.1: Comparación entre el Outlier Score entregado por IF y el riesgo calculado con la formula de riesgo actual

4.3.2. Implementación

Formula de ponderación de riesgo:

$$R = \frac{\sum_{i=0}^n (P_n \cdot X_n)}{\sum_{i=0}^n P_n} \cdot 100 \cdot O \quad (4.1)$$

Donde:

- P_n : Peso ponderado del vector de ataque n.

- X_n : Presencia del vector de ataque n, es 1 si existe y 0 si no.
- O : ponderación de anormalidad de Isolation Forest.

Vector de ataque	Probabilidad de Ocurrencia	Impacto	Peso
password_policy	0.15	95	14.003
cve	0.06	90	5.67
insecure_shared_folder	0.26	60	15.75
web_login	0.31	35	10.843
suspicious_url	0.44	20	8.714
default_credentials	0.26	75	19.6875
possible_directory_traversal	0.24	45	10.9845
dbms	0.26	90	23.625
num_incidents_active	0.06	30	1.65
num_incidents_mitigated	0.94	1	0.94
total_low_incidents	0.11	20	2.22
total_medium_incidents	0.28	50	13.85
total_high_incidents	0.33	80	26.64
total_critical_incidents	0.28	100	27.7

Cuadro 4.5: Peso por vector de ataque.

Para asegurar que el riesgo considere el lado defensivo y ofensivo del sistema, se ha diseñado una función que considere tanto los datos de la planilla de incidentes como el grado de anormalidad de las detecciones ocupando una fórmula (1) definida por el equipo de seguridad encargado del proyecto y modificada por nosotros para considerar el efecto de Isolation Forest en el riesgo. Esta considera una serie de pesos asociados a cada vector obtenidos al ponderar el porcentaje de incidentes relacionados al vector y multiplicarlos por su impacto en el sistema obtenido anteriormente. Esto se divide por el máximo riesgo posible para conseguir porcentaje total del riesgo y se ajusta a una escala del 1 al 100 (Cuadro 4.5). Finalmente esto se multiplica por la ponderación de IF que ofrece un Outlier Score que identifica el nivel de separación de la muestra del resto de las mediciones con un valor entre 0 y 1. Esto permite separar aquellos valores que destacan sin afectar significativamente a aquellos valores que presentan un riesgo aceptable.

Para concluir, podemos considerar que Isolation Forest es una solución efectiva a nuestro desafío principal de modificar el funcionamiento de Recon para que su visión de riesgo considere factores más dinámicos por medio de aprendizaje de máquinas. Nos interesa ver la capacidad de IF de identificar los casos de alto riesgo para responder la pregunta de investigación.

Capítulo 5

Análisis de Resultados

Este capítulo analiza los resultados obtenidos del experimento, evaluando su capacidad de identificar los casos de alto riesgo y los casos de riesgo limítrofe.

Para comprobar el funcionamiento de la herramienta, se han provisto dos fuentes de datos de parte del equipo de seguridad. Estas fuentes de datos son confidenciales, así que no se mostraran por razones de seguridad, pero se confirmó que el comportamiento de las fuentes de datos y del funcionamiento normal del sistema son congruentes. Comparando los resultados obtenidos con las medidas de riesgo obtenidas anteriormente por Recon podemos ver que Isolation Forest es capaz de separar aquellos casos de interés presentes en la muestra sin afectar los datos que se consideran normales. Esta propiedad se mantiene tanto en el caso de detecciones como en el caso de activos para las muestras (Figura 5.1 y 5.2). Podemos observar que en las detecciones (Figuras 5.1.A y 5.1.C) ocurre que aquellos casos más riesgosos se separan del comportamiento normal del sistema sin alterar el riesgo de los casos normales mayormente. En el caso de los activos no se puede notar tan claramente el efecto en la mayoría de los casos excepto aquellos que se encuentran al extremo derecho de la gráfica (Figuras 5.1.B y 5.1.D), estos corresponden a aquellos casos donde se encuentran incidentes abiertos. Estos puntos se pueden ver en la tabla (Cuadro 5.1 y 5.2). Podemos ver que el nivel de contaminación no presenta mayores efectos en la distinción de casos, también podemos

concluir que IF puede separar los casos poco comunes dentro de las detecciones con mayor facilidad que en el caso de los activos ya que el comportamiento común de las detecciones es más regular que en el segundo caso.

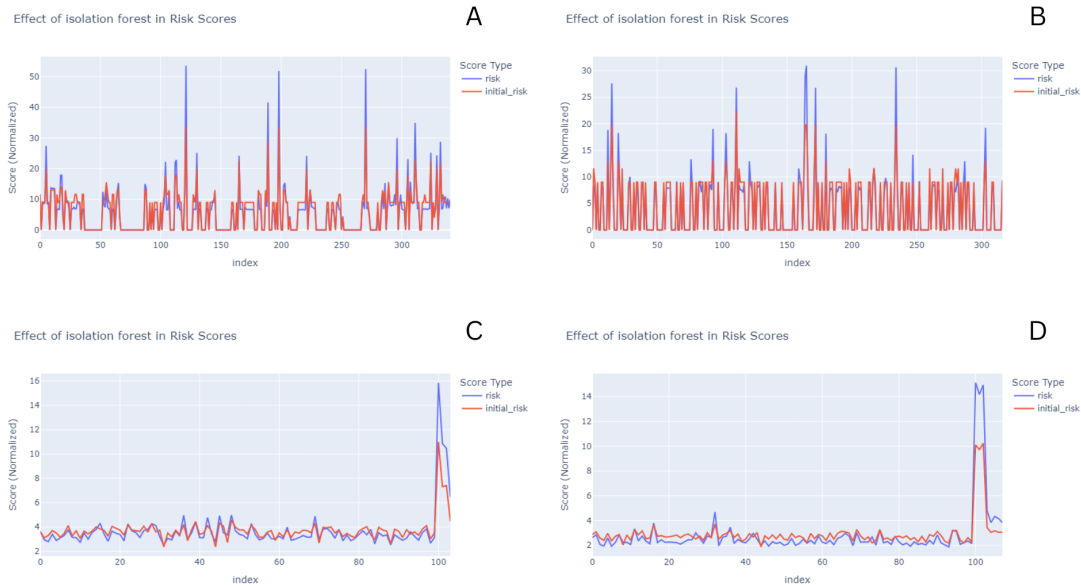


Figura 5.1: Efecto de IF en datasets con una contaminación del 3 %.A: Dataset 1 - Detecciones, B: Dataset 2 - Detecciones, C: Dataset 1 - Activos y D: Dataset 2 - Activos

Bajos	Medios	Altos	Críticos	Outlier score	risk	initial_risk
0.0	0.0	0.0	1.0	66.01 %	15.86	10.99
0.0	1.0	0.0	0.0	67.86 %	10.86	7.32
0.0	1.0	0.0	0.0	64.68 %	10.47	7.41
1.0	0.0	0.0	0.0	65.94 %	6.46	4.49

Cuadro 5.1: Mediciones de activos en la primera fuente de datos

Aunque podemos concluir con certeza que el sistema es capaz de separar aquellos casos de alto riesgo con certeza e identificar casos limítrofes que contienen características similares a aquellos casos de alto riesgo, el obtener una medida de precisión y de error se vuelve complejo ya que el dataset entregado no mencionaba aquellos casos de mayor riesgo o aquellos casos que querían ser identificados con certeza. La mayor comparación que podemos hacer es ver los casos específicos que se señalan como

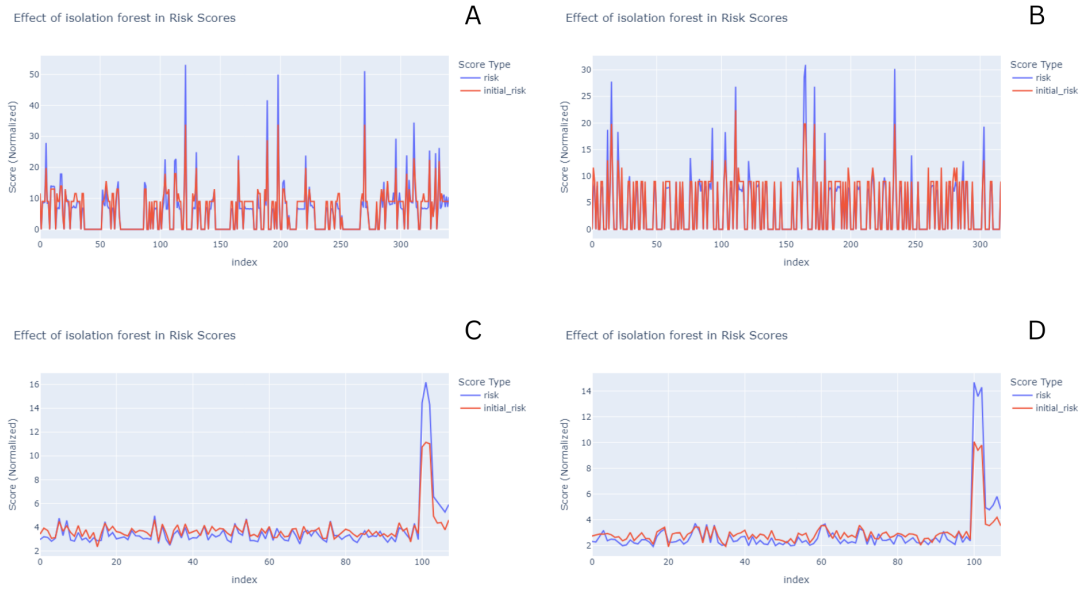


Figura 5.2: Efecto de IF en datasets con una contaminación del 10 %.A: Dataset 1 - Detecciones, B: Dataset 2 - Detecciones, C: Dataset 1 - Activos y D: Dataset 2 - Activos

Bajos	Medios	Altos	Críticos	Outlier score	risk	initial_risk
0.0	0.0	0.0	1.0	64.29 %	15.77	11.22
0.0	0.0	1.0	0.0	62.29 %	14.89	10.94
0.0	0.0	1.0	0.0	66.65 %	15.96	10.96
1.0	0.0	0.0	0.0	59.99 %	5.38	4.1
1.0	0.0	0.0	0.0	60.92 %	5.3	3.98
1.0	0.0	0.0	0.0	59.21 %	6.5	5.02
1.0	0.0	0.0	0.0	58.78 %	5.13	4
1.0	0.0	0.0	0.0	59.36 %	5.29	4.08

Cuadro 5.2: Mediciones de activos en la segunda fuente de datos

indicativos del comportamiento y validar que siguen lo esperado.

5.1. Discusión

Para esta investigación, nos interesa saber cual es el efecto de modelos no supervisados, en particular Isolation Forest, en la evaluación de riesgo de un sistema y cómo puede incorporarse en un caso real.

El uso de la herramienta ha sido capaz de identificar los casos de mayor riesgo como fue previsto. La diferencia entre el caso que no ocupa IF y el caso que si es tal que una propuesta posible de trabajo futuro puede ser el uso de IF por su cuenta, sin considerar los pesos, para identificar los casos de alto riesgo basándose únicamente en los datos de detección. Podemos concluir que los datos presentan comportamiento tal que los casos de alto riesgo en el sistema poseen características fácilmente distinguibles y estas características pueden ser usadas para llamar la atención a casos que en la superficie parecen inocuos pero poseen comportamientos similares a los casos de alto riesgo. Podemos ver esto en los resultados donde los casos ya conocidos de alto riesgo se elevan por sobre el comportamiento esperado, pero también hay casos que en la situación original no eran notables que se volvieron más riesgosos debido a esta relación establecida. La mayor debilidad del sistema viene al uso de activos, ya que estos no presentan un comportamiento normal base esperado como existe en las detecciones y, por lo tanto, solo se pueden distinguir los casos más claros en vez de aquellos casos limítrofes.

Los resultados obtenidos demuestran el potencial de la herramienta para identificar de manera efectiva los casos de mayor riesgo, tal como se previó. La capacidad de distinguir características distintivas en los casos de alto riesgo, incluso en aquellos que superficialmente parecen inocuos, resulta especialmente valiosa para empresas con redes internas. Al aplicar esta herramienta, estas organizaciones pueden detectar anomalías y comportamientos sospechosos que podrían pasar desapercibidos en análisis tradicionales. Esto permite a las compañías tomar medidas proactivas para proteger sus activos y minimizar el riesgo de incidentes de seguridad.

5.2. Trabajo futuro

Debido a las limitaciones de tiempo y disponibilidad de recursos, no se pudo demostrar la solución con fuentes de datos en tiempo real, sí fue posible con datos emulados. Para un trabajo futuro, es de gran interés medir el rendimiento de la herramienta

al recibir datos en tiempo real desde los nodos en una red interna. Es también posible reformular el uso de activos en el sistema para acomodar mejor la realidad del negocio y del comportamiento real de la red. En este momento, nuestra visión de Activos no presenta un comportamiento normalizado como las detecciones, así que la herramienta no ofrece la misma calidad de información como esta entrega cuando se ocupa de los datos referentes a las detecciones. Un factor limitante también fue la falta de acceso a medidas de validación debido a la falta de etiquetado en la fuente de datos. Vale la pena considerar un acercamiento supervisado para un posible trabajo futuro y sus ramificaciones en la calidad del entrenamiento. Existe también una gama amplia para incrementar el número de vectores de ataque considerados por medio de herramientas no consideradas en el estado actual, por ejemplo: no se hace ningún análisis longitudinal al tráfico de los nodos en el tiempo, solo se analiza su comportamiento por medio de fingerprints de NMAP.

Capítulo 6

Conclusiones

El uso de Isolation Forest (IF) ha demostrado ser eficaz para identificar casos de alto riesgo dentro de los datos proporcionados, cumpliendo con las expectativas iniciales. La herramienta es capaz de distinguir patrones anómalos en las detecciones, separando de manera consistente los casos críticos del comportamiento normal del sistema sin afectar significativamente los datos considerados normales. Este comportamiento destaca la utilidad de IF en datasets de detecciones, donde los casos de riesgo alto poseen características distinguibles que permiten una separación clara.

La implementación de IF tuvo un efecto en la visión de riesgo acotado debido a las limitaciones del ambiente de trabajo, pero se puede visualizar el uso de IF en combinación con otros elementos de ML para ampliar la superficie de ataque de formas no consideradas en la red de trabajo actualmente, como por ejemplo: el uso de historias de detecciones de nodos para la predicción de posibles riesgos futuros, o el uso de etiquetas para considerar alternativas supervisadas.

En general, los resultados refuerzan la utilidad de IF para detecciones, donde su capacidad de identificar casos atípicos es particularmente efectiva. Sin embargo, en el contexto de los activos, es necesario realizar ajustes para optimizar su desempeño. Esto podría incluir la normalización de los datos de activos o el desarrollo de estrategias específicas que complementen el análisis de IF en este caso.

Anexos

Nombre vector	Objetivos
password_policy	Validar que no hay contraseñas débiles en el sistema.
cve	Demostrar el impacto de vulnerabilidades activas conocidas.
insecure_shared_folder	Detectar exposición de datos confidenciales en SMB mal configurado.
web_login	Validar seguridad en login web.
suspicious_url	Analizar URLs detectadas en el web scan.
default_credentials	Detectar credenciales por defecto en tecnologías conocidas.
possible_directory_traversal	Detectar Directory Traversal y acceso a archivos.
dbms	Validar configuración segura de DBMS.

Nombre vector	Principal riesgo abordado
password_policy	Usuarios genéricos con contraseñas débiles.
cve	Vulnerabilidades críticas con RCE.
insecure_shared_folder	Exposición de PII y datos críticos.
web_login	Fuerza bruta, SQLi bypass, etc.
suspicious_url	Exposición de repositorios, backups, APIs, etc.
default_credentials	Acceso directo al sistema.
possible_directory_traversal	LFI para leer archivos críticos.
dbms	Configuración insegura y fuga de datos.

Cuadro 6.1: Objetivos y riesgo abordado por vector

Bibliografía

- [1] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. Mitre att&ck: Design and philosophy. In *Technical report*. The MITRE Corporation, 2018.
- [2] G Jayasuryapal, P Meher Pranay, Harpreet Kaur, et al. A survey on network penetration testing. In *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pages 373–378. IEEE, 2021.
- [3] Jinfeng Li. Vulnerabilities mapping based on owasp-sans: a survey for static application security testing (sast). *arXiv preprint arXiv:2004.03216*, 2020.
- [4] Ugochukwu Ikechukwu Okoli, Ogugua Chimezie Obi, Adebunmi Okechukwu Adewusi, and Temitayo Oluwaseun Abrahams. Machine learning in cybersecurity: A review of threat detection and defense mechanisms. *World Journal of Advanced Research and Reviews*, 21(1):2286–2295, 2024.
- [5] Michael LeBlanc and John Crowley. A review of tree-based prognostic models. *Recent advances in clinical trial design and analysis*, pages 113–124, 1995.
- [6] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520, 2022.
- [7] Zhaojie Ju and Honghai Liu. Fuzzy gaussian mixture models. *Pattern Recognition*, 45(3):1146–1158, 2012.

- [8] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth IEEE international conference on data mining*, pages 413–422. IEEE, 2008.
- [9] Xu Li, Songren Deng, Lifang Li, and Yunchuan Jiang. Outlier detection based on robust mahalanobis distance and its application. *Open Journal of Statistics*, 9(1):15–26, 2019.
- [10] Zhangyu Cheng, Chengming Zou, and Jianwei Dong. Outlier detection using isolation forest and local outlier factor. In *Proceedings of the conference on research in adaptive and convergent systems*, pages 161–168, 2019.
- [11] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452, 2008.
- [12] Mostafa Afgani, Sinan Sinanovic, and Harald Haas. Anomaly detection using the kullback-leibler divergence metric. In *2008 First International Symposium on Applied Sciences on Biomedical and Communication Technologies*, pages 1–5. IEEE, 2008.
- [13] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [14] Alberto Munoz and Jorge Muruzábal. Self-organizing maps for outlier detection. *Neurocomputing*, 18(1-3):33–60, 1998.
- [15] Rémi Domingues, Maurizio Filippone, Pietro Michiardi, and Jihane Zouaoui. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern recognition*, 74:406–421, 2018.