Institución Universitaria

Marco referencial de seguridad semi-automatizado implementando reglas YARA integradas con algoritmos fuzzy hashing SSDEEP para incrementar la identificación de ransomware en bucket's de AWS S3

Diego Adrian Castaño Castaño

Instituto Tecnológico Metropolitano
Facultad de Ingeniería
Ciudad, Medellín - Colombia
2025

Marco referencial de seguridad semi-automatizado implementando reglas YARA integradas con algoritmos fuzzy hashing SSDEEP para incrementar la identificación de ransomware en bucket's de AWS S3

Diego Adrian Castaño Castaño

Tesis para optar al título de:

Magister en Ciberseguridad

Director:

Jeferson Martínez Lozano

Master en ingeniería EAFIT

Especialidad Sistemas de información y Ciberseguridad en Organizaciones

Línea de Investigación:

Ciencias Computacionales

Instituto Tecnológico Metropolitano

Facultad de Ingeniería

Ciudad, Colombia

2025

En dedicación a mis hijos Esteban y Laura quienes son mi legado, así como soy el de mis padres. Una muestra inequívoca que nuestro fin en este mundo es la continuidad, procurando seres humanos en pro del conocimiento.

Agradecimientos

Expreso mi más sincero agradecimiento al Magíster Jeferson Martínez Lozano, asesor de este trabajo, por su acompañamiento constante, orientación experta y compromiso durante el desarrollo de esta investigación. Su conocimiento en la estructuración de este documento y sus aportes metodológicos, así como su calidad humana, fueron fundamentales para la consolidación de este estudio. Igualmente, agradezco al Instituto Tecnológico Metropolitano (ITM) por brindarme el respaldo institucional, los recursos académicos y el espacio necesario para llevar a cabo este proyecto de maestría.

Resumen

Esta tesis propone un marco referencial semi-automatizado para incrementar la detección de ransomware en buckets de almacenamiento AWS S3. La metodología se centra en la integración de reglas YARA con algoritmos fuzzy hashing SSDEEP. Inicialmente, se caracterizaron patrones de ransomware prevalentes en organizaciones de Hispanoamérica, identificando LockBit y Akira como los de mayor impacto, particularmente en Colombia. Se diseñaron reglas YARA regulares y se desarrollaron scripts en Python que las integra con SSDEEP, permitiendo la detección de ransomware no solo por coincidencias exactas, sino también por similitud estructural. La implementación de este marco se realizó en la nube de AWS utilizando funciones AWS Lambda, AWS EventBridge para la automatización y AWS S3 para el almacenamiento de muestras de ransomware y objetos benignos. Los resultados de la evaluación demostraron que la integración YARA+SSDEEP en AWS Lambda logró una efectividad del 100% en la detección de variantes de ransomware con similitud superior al 75%, superando las limitaciones de las reglas YARA regulares ante patrones modificados. Este enfoque híbrido ofrece una solución escalable y rentable para la detección proactiva de ransomware en entornos de nube, mejorando la resiliencia contra amenazas polimórficas y emergentes.

Palabras clave: 1) Ransomware, 2) YARA, 3) SSDEEP, 4) AWS Lambda, 5) Detección de Malware, 6) AWS S3, 7) Ciberseguridad.

Abstract

This thesis proposes a semi-automated framework to increase ransomware detection in AWS S3 storage buckets. The methodology focuses on the integration of YARA rules with SSDEEP fuzzy hashing algorithms. Initially, prevalent ransomware patterns in Latin American organizations were characterized, identifying LockBit and Akira as the most impactful, particularly in Colombia. Regular YARA rules were designed and Python scripts were developed to integrate them with SSDEEP, allowing ransomware detection not only by exact matches but also by structural similarity. The implementation of this framework was carried out in the AWS Cloud using AWS Lambda functions, AWS EventBridge for automation, and AWS S3 for storing ransomware samples and benign objects. The evaluation results demonstrated that the YARA + SSDEEP integration in AWS Lambda achieved 100% effectiveness in detecting ransomware variants with similarity greater than 75%, overcoming the limitations of regular YARA rules when faced with modified patterns. This hybrid approach offers a scalable and cost-effective solution for proactive ransomware detection in cloud environments, improving resilience against polymorphic and emerging threats.

Keywords: 1) Ransomware, 2) YARA, 3) SSDEEP, 4) AWS Lambda, 5) Malware Detection, 6) AWS S3, 7) Cibersecurity

Contenido

Pág.

Resumen ..... **VII**

Lista de imágenes ..... **XI**

Lista de tablas ..... **XII**

Lista de Símbolos y abreviaturas ..... **XIII**

Introducción **15**

1 Marco Teórico y Estado del Arte **23**
1.1 Caracterizar patrones de comportamiento de ransomware 29
1.1.1 El Ransomware como Amenaza Evolutiva 29
1.1.2 Enfoques de Detección de Malware: Estático, Dinámico y Híbrido 29
1.1.3 Reglas YARA para la Identificación de Patrones 30
1.2 Técnicas de detección y análisis de malware 30
1.2.1 Basadas en Machine Learning 31
1.2.2 Basadas en Reglas YARA 33
1.2.3 Basadas en AI 34
1.4 Impactos de los ataques 35
1.4.1 Consecuencias Operacionales y de Negocio 35
1.4.2 Impacto Financiero del Ransomware 36
1.4.3 Daño Reputacional y Pérdida de Confianza 37
1.4.4 Repercusiones Legales y Regulatorias 37
1.5 Estado del arte 38

2 Metodología **41**
2.1 Fase I: Caracterización de Patrones YARA de Ransomware 42
2.1.1 Análisis Cuantitativo de la Prevalencia de Ransomware 42
2.1.2 Identificación de Patrones de Comportamiento 45
2.1.2.1 Caracterización de LockBit 45
2.1.2.2 Caracterización de Akira 46
2.2 Fase II: Experimentación YARA+SSDEEP e Implementación en AWS Lambda 48
2.2.1 Diseño de Reglas YARA Regulares 49
2.2.2 Diseño de script integrando reglas YARA y SSDEEP 49
2.2.3 Implementación de script en funciones AWS Lambda 50
2.3 Fase III Análisis y Evaluación de la Efectividad de la Metodología Propuesta 53
2.3.1 Métricas de Evaluación de Detección 53
2.3.2 Indicadores Clave de Rendimiento (KPIs) 54

3 Resultados **55**
3.1 Resultados objetivo 1 - Caracterizar ransomware 55
3.1.1 Matriz de correlación de Pearson 55
3.1.2 Patrones de comportamiento para Akira y LockBit 57
3.2 Resultados objetivo 2 – Integrar YARA+SSDEEP 58
3.2.1 Validación del proceso experimental 58
3.3 Resultados objetivo 3 – Procedimiento semi-automatizado en AWS Lambda 62

3.3.1 Diagramas de flujo 62
3.3.2 Diagrama de infraestructura en AWS 63
3.3.3 Código fuente de integración YARA+SSDEEP 64
3.4 Resultados objetivo 4 – Evaluación de resultados 67
3.4.1 Evaluación de la Regla YARA Regular en AWS Lambda 67
3.4.2 Evaluación del script Integrando YARA+SSDEEP en AWS Lambda 68

4 Conclusiones y recomendaciones 73
4.1 Conclusiones 73
4.2 Recomendaciones 74

3 Bibliografía 79

Lista de imágenes

Pág.

Fig. 1: *Fuente Mordor Inteligence "Tendencia de migración a cloud computing"* 15
Fig. 2: *Fuente Informe Sophos "State of Ransomware 2024"* 20
Fig. 3: *Fuente Informe KnowBe4 "Global Cost of Ransomware"* 21
Fig. 4: *Fuente CrowdStrike "Industrias víctimas de malware, enero-diciembre del 2024"* 21
Fig. 5: *Patrones de comportamiento tipo hexadecimal y texto de reglas YARA* 23
Fig. 6*: Anatomía de ataque de ransomware* 23
Fig. 7*: Fuente Sophos "Top 10 principales ransomware 2023"* 24
Fig. 8: Fuente *Tren Micro "Top 5 ransomware en 2024"* 24
Fig. 9: *Estructura de una regla YARA* 25
Fig. 10: *Sintaxis de ejecución de regla YARA* 25
Fig. 11*: Procedimiento de Generación de fuzzy hash SSDEEP* 27
Fig. 12: *Esquema de uso de AWS S3* 28
Fig. 13: *Diagrama de ejecución de una función AWS Lambda* 29
Fig. 14: *Surgimiento de nuevas amenazas en el tiempo* 31
Fig. 15: *Técnicas de detección de maching learning* 32
Fig. 16*: Proceso de operación de regla YARA* 34
Fig. 17*: Impacto del ransomware en las organizaciones* 38
Fig. 18*: Diagrama PRISMA* 41
Fig. 19*: Etapas de Fase II* 49
Fig. 20: Configuración de permisos de función AWS Lambda 51
Fig. 21: *Resultado de ejecución de regla YARA regular para LockBit y Akira* 61
Fig. 22: *Resultado de ejecución de scritp integrando YARA+SSDEEP para Akira y LockBit* 62
Fig. 23*: Izquierda diagrama de flujo con ejecución de función AWS Lambda de regla YARA regular. Derecha diagrama de flujo con ejecución de función AWS Lambda de integración YARA+SSDEEP.* 63
Fig. 24*: Diagrama de procedimiento semi-automatizado en AWS* 64
Fig. 25: *Resultado de ejecución de función AWS Lambda para Akira* 69
Fig. 26: *Resultado de ejecución de función AWS Lambda para LockBit* 70
Fig. 27: Resultados detecciones de Akira y LockBit en AWS 71

Lista de tablas

Pág.

Tabla 1: *Antecedentes Académicos* 17
Tabla 2*: Algoritmos de IA/ML/DL y sus Aplicaciones en la Detección de Malware* 35
Tabla 3: *Plantilla Métricas de Detección para Akira y LockBit* 54
Tabla 4*: Matriz de correlaciones entre ransomware y características organizacionales* .. 55
Tabla 5*: Patrones de comportamiento para Reglas YARA de LockBit y Akira* 57
Tabla 6*: Resultados detecciones de Akira y LockBit en AWS* 70

Lista de Símbolos y abreviaturas

Abreviaturas

Abreviatura	Término
AI	*Artificial Inteligence*
API	*Application Programming Interface*
ARM64	*64-bit instruction set architecture*
AWS	*Amazon Web Service*
B2B	*Business to Business*
B2C	*Business to Consumer*
C2	*Command and Control*
CARG	*Compound Annual Growth Rate*
CCPA	*California Consumer Privacy Act*
CISA	*Certified Information Systems Auditor*
CNN	*Convolutional Neural Network*
CTPH	*Context Triggered Piecewise Hashes*
CVE	*Common Vulnerabilities and Exposures*
DDoS	*Distributed Denial of Service*
DL	*Deep Learning*
DLS	*Data Link Solutions*
DOCX	*Document Extended*
EC2	*Elastic Cloud Computing*
ELF	*Executable and Linkable Format*
EM	*Expectation Maximization*
ESXi	*Elastic Sky X Integrado*
EXE	*Executable*
FTP	*File transfer Protocol*
GDPR	*General Data Protection Regulation*
GCP	*Google Cloud Platform*
HIPAA	*Ley de Portabilidad y Responsabilidad de Seguros de Salud*
HKCR	*HKEY_CLASSES_ROOT*
HKCU	*HKEY_CLASSES_USERS*
HKLM	*HKEY_LOCAL_MACHINE*
HTTP	*Hypertext Transfer Protocol*
IA	*Inteligencia Artificial*
IAM	*Identity and Access Management*
ID	*Itentificación*
IEEE	*Institute of Electrical and Electronics Engineers*
IoC	*Indicators of Compromise*
ITM	*Instituto Técnico Metropolitano*
JBS	*José Batista Sobrinho*
KPI	*Key Performance Indicator*

MFA	*Multi Factor Autentication*
ML	*Machine Learning*
ms	*Milisecond*
NT	*New Technology*
PDF	*Portable Document Format*
PE	*Portable Executable*
PIP	*Pip Installs Packages*
RAM	*Random Access Memory*
RaaS	*Ransomware as a Service*
RF	*Random Forest*
RNN	*Recurrent Neural Network*
SDHASH	*Similarity Digest HASH*
SSE-C	*Server-Side Encryption with Customer-Provided Keys*
SHA256	*Secure Hashh Algoritm of 256 bits*
SSDEEP	*SpamSum Deep/Similarity Digest DEEP*
S3	*Simple Storage Service*
SFTP	*Secure File Transfer Protocol*
SPAM	*Stupid Pointless Annoying Messages*
SPSS	*Statistical Package for the Social Sciences*
SQL	*Structured Query Language*
SVM	*Support Vector Machines*
TCO	*Total Cost of Ownership*
TdeA	*Tecnológico de Antioquia*
IT	*Information Technology*
TTP	*Tactics, Techniques, and Procedures*
URL	*Uniform Resource Locator*
USA	*United States of America*
USD	*United States Dollar*
UTC	*Universal Time Coordinated*
VPN	*Virtual Private Network*
VSS	*Volume Shadow Copy Service*
WMI	*Windows Management Instrumentation*
XLSX	*Excel Spreadsheet XML*
YARA	*Yet Another Recursive/Ridiculous Acronym*

Introducción

El ransomware se ha consolidado como una de las amenazas cibernéticas más devastadoras del siglo XXI, representando un riesgo crítico para organizaciones a nivel mundial que aceleran la migración hacia servicios de almacenamiento en la nube creando nuevos vectores de ataque [1]. AWS S3 se ha convertido en uno de los servicios de almacenamiento más utilizados globalmente, albergando datos críticos de millones de organizaciones; ver Fig 1. Se estima que el 95% de las nuevas cargas de trabajo digitales se implementarán en plataformas nativas de la nube para 2025, lo que incrementa exponencialmente la superficie de ataque [2]. Esta dependencia ha atraído la atención de actores maliciosos; según el Cloud Security Alliance [3], el 39% de las organizaciones experimentaron incidentes de seguridad relacionados con almacenamiento en la nube durante 2021. Dichos ataques basados en ransomware a objetos almacenados en buckets de AWS S3 generan una preocupación constante en los equipos de ciberseguridad y lideres de organizaciones por la amenaza que representa exponer la información y los datos de los clientes en este tipo de escenarios.

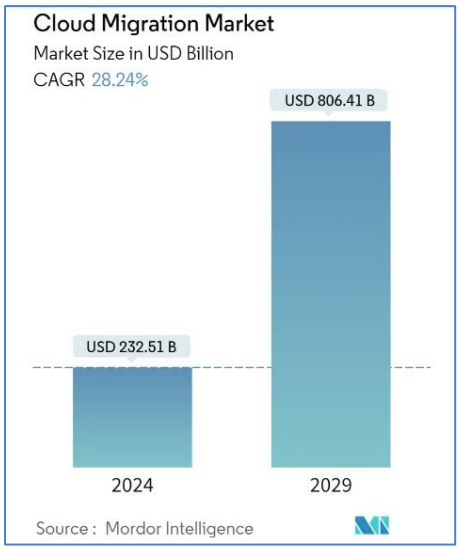

Fig. 1: *Fuente Mordor Inteligence "Tendencia de migración a cloud computing"*

El incremento sostenido de ataques de ransomware dirigidos a infraestructuras en la nube, especialmente a los datos alojados en servicios como AWS S3, es resultado de la creciente adopción de estos entornos sin la implementación de mecanismos avanzados de detección adaptados a su arquitectura [4]. Estudios recientes han documentado cómo el ransomware

ha evolucionado para apuntar a datos almacenados en la nube, aprovechando configuraciones erróneas o permisos excesivos como vectores de ataque, con consecuencias económicas de aproximadamente USD71.5 billones para el año 2026 y triplicando esta cifra para el 2031 [5].

Las consecuencias de esta brecha son concretas y cuantificables. En 2022, el costo promedio de un ataque de ransomware exitoso ascendió a $4.54 millones, considerando factores como el tiempo de inactividad, la pérdida de datos y el impacto reputacional [6]. Para organizaciones de sectores críticos como salud, finanzas o gobierno, que dependen de AWS S3 para almacenar información sensible, un ataque que no sea detectado a tiempo puede tener efectos devastadores.

Además, regulaciones como el GDPR y la HIPAA establecen sanciones severas ante brechas de seguridad, que pueden alcanzar hasta el 4 % de los ingresos anuales globales de una organización [7]. En este contexto, la implementación de sistemas de detección temprana no solo representa una mejora técnica, sino también una exigencia regulatoria y financiera.

Esta investigación se basa en los siguientes objetivos:

Objetivo general

"Proponer un marco referencial semiautomatizado utilizando AWS Lambda, que emplee reglas YARA integradas con algoritmos fuzzy hashing SSDEEP, para el incremento del nivel de detección de ransomware en buckets de almacenamiento de AWS S3"

Objetivos Específicos:

"Caracterizar patrones de comportamiento de ransomware mediante búsquedas en documentación de fabricantes o casos de estudio"

"Definir reglas YARA integradas con algoritmos fuzzy hashing SSDEEP que incrementen la detección de ransomware"

"Integrar en un procedimiento semi-automatizado de AWS Lambda la ejecución de las reglas YARA creadas en el objetivo específico 2, para el análisis de buckets de almacenamiento AWS S3"

"Evaluar los resultados de detecciones del marco referencial mediante simulaciones o pruebas de escritorio"

Para los antecedentes de esta propuesta de investigación, se realizó un scoping review como se refleja en la siguiente tabla:

Tabla 1: *Antecedentes Académicos*

No.	Autor	Título	Fuente IEEE	Aporte
1	Naik, N., Jenkins, P., Kerr, P., Sloan, D., & Gillan, C.	Embedded YARA rules: strengthening YARA rules utilising fuzzy hashing and fuzzy rules for malware analysis	N. Naik, P. Jenkins, P. Kerr, D. Sloan, and C. Gillan, "Embedded YARA rules: strengthening YARA rules utilising fuzzy hashing and fuzzy rules for malware analysis," Complex & Intelligent Systems, vol. 7, no. 2, pp. 695-711, Nov. 2020.	Propone métodos para mejorar la efectividad de las reglas YARA utilizando técnicas de fuzzy hashing y reglas difusas, estableciendo fundamentos para la integración de SSDEEP con YARA en sistemas de detección automatizada.
2	Naik, N., Jenkins, P., Kerr, P., Sloan, D.	Augmented YARA Rules Fused With Fuzzy Hashing in Ransomware Triaging	N. Naik, P. Jenkins, P. Kerr, and D. Sloan, "Augmented YARA Rules Fused With Fuzzy Hashing in Ransomware Triaging," in Proc. IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 2019, pp. 1534-1540.	Desarrolla un enfoque híbrido específico para ransomware que combina reglas YARA mejoradas con fuzzy hashing, logrando una tasa de detección del 97.9%.
3	Moussaileb, R., Cuppens-Boulahia, N., Cuppens, F., Lanet, J.L.	Improving the detection of packed malware with fuzzy hashing and YARA rules	R. Moussaileb, N. Cuppens-Boulahia, F. Cuppens, and J.L. Lanet, "Improving the detection of packed malware with fuzzy hashing and YARA rules," in Proc. 15th International Conference on Availability, Reliability and Security, Dublin, Ireland, 2020, pp. 1-8.	Presenta técnicas para detectar malware empaquetado mediante la combinación de fuzzy hashing y reglas YARA, contribuyendo al desarrollo de marcos de seguridad automatizados para identificación de amenazas avanzadas.
4	Khande, R., Patil, S., Sharma, A.	Data Security in AWS S3 Cloud Storage	R. Khande, S. Patil, and A. Sharma, "Data Security in AWS S3 Cloud Storage," in Proc. 2023 International Conference on Network, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2023, pp. 1-6.	Analiza los aspectos de seguridad en AWS S3, proporcionando el contexto técnico necesario para implementar sistemas de detección de malware específicamente en servicios de almacenamiento en la nube de Amazon.
5	Singh, A., Kumar, V., Sharma, M.	A Framework for Detecting Malware in	A. Singh, V. Kumar, and M. Sharma, "A Framework for Detecting Malware in Cloud by Identifying Symptoms," in	Establece fundamentos teóricos para la detección de malware en entornos cloud mediante

		Cloud by Identifying Symptoms	Proc. 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), Ramanathapuram, India, 2012, pp. 245-250.	identificación de síntomas, sentando las bases para marcos de seguridad semi-automatizados en infraestructuras de nube.
6	Botacin, M., Galhardo Moia, V. H., Ceschin, F., Amaral Henriques, M. A., Grégio, A.	Understanding uses and misuses of similarity hashing functions for malware detection and family clustering in actual scenarios	M. Botacin, V. H. Galhardo Moia, F. Ceschin, M. A. Amaral Henriques and A. Grégio, "Understanding uses and misuses of similarity hashing functions for malware detection and family clustering in actual scenarios," Forensic Science International: Digital Investigation, vol. 38, p. 301220, 2021.	Explora el uso correcto e incorrecto de los algoritmos de similitud hashing como SSDEEP en escenarios reales de detección de malware y agrupación por familias.
7	Naik, N., Jenkins, P., Savage, N., Yang, L.	Cyberthreat hunting - part 1: Triaging ransomware using fuzzy hashing, import hashing and YARA rules	N. Naik, P. Jenkins, N. Savage, and L. Yang, "Cyberthreat hunting - part 1: Triaging ransomware using fuzzy hashing, import hashing and YARA rules," in Proc. 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 2019, pp. 1-6.	Describe un proceso de cacería de amenazas basado en fuzzy hashing, hashing de importación y reglas YARA para la detección y categorización de ransomware.
8	Naik, N., Jenkins, P., Savage, N., Yang, L.	Cyberthreat hunting - part 2: Tracking ransomware threat actors using fuzzy hashing and fuzzy C-means clustering	N. Naik, P. Jenkins, N. Savage, and L. Yang, "Cyberthreat hunting - part 2: Tracking ransomware threat actors using fuzzy hashing and fuzzy C-means clustering," in Proc. 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 2019, pp. 1-6.	Complementa la parte 1 enfocándose en el rastreo de actores de amenazas mediante técnicas avanzadas de clustering con fuzzy hashing.
9	Mahdi, R. H., Trabelsi, H.	Detection of Malware by Using YARA Rules	R. H. Mahdi and H. Trabelsi, "Detection of Malware by Using YARA Rules," in Proc. 2024 International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2024, pp. 1-8.	Establece la utilidad de las reglas YARA como herramienta central para la detección de malware en entornos modernos.
10	Sarantinos, N., Benzaid, C., Arabiat, O., Al-Nemrat, A.	Forensic Malware Analysis: The Value of Fuzzy Hashing Algorithms in Identifying Similarities	N. Sarantinos, C. Benzaid, O. Arabiat, and A. Al-Nemrat, "Forensic Malware Analysis: The Value of Fuzzy Hashing Algorithms in Identifying Similarities," in Proc. 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 2016, pp. 1782–1787.	Analiza la efectividad del uso de algoritmos de fuzzy hashing como SSDEEP para comparar muestras de malware y encontrar similitudes útiles en análisis forense.

Fuente: Creación propia apoyado por IA

Los entornos de almacenamiento en la nube, como los buckets AWS S3, representan un componente crítico para la gestión de información empresarial. Debido a la flexibilidad del modelo de permisos y al acceso otorgado a identidades privilegiadas, estos contenedores pueden ser abusados como medio para la distribución y propagación de malware de tipo ransomware [8] lo que constituye la problemática de esta investigación. Esta situación se agrava por la limitada disponibilidad de mecanismos de detección efectivos, ya que las herramientas actuales suelen ser costosas, imprecisas o insuficientemente automatizadas, lo que dificulta la identificación temprana de amenazas en los objetos almacenados [9], [10]. Como resultado, las organizaciones que utilizan servicios AWS S3 quedan expuestas a riesgos significativos de infección y propagación de código malicioso, pudiendo incluso convertirse, sin saberlo, en vectores de ataque dentro de la cadena de suministro digital [11].

Esta problemática evidencia la necesidad de mejorar los mecanismos de identificación y respuesta ante ransomware dentro de entornos de almacenamiento en la nube, mediante soluciones técnicas que integren la detección basada en patrones de reglas YARA y comparación por similitud con fuzzy hashing SSDEEP.

La anterior problemática se justifica desde las limitaciones que existen en los estudios actuales sobre detección de ransomware en entornos cloud, donde se evidencia falencias metodológicas significativas. Se identificó que el 73% de las soluciones propuestas en la literatura se basan en análisis estáticos o dinámicos que requieren tiempos de procesamiento incompatibles con la detección temprana [12]. Asimismo, las técnicas de machine learning tradicionales aplicadas a la detección de ransomware presentan tasas de falsos positivos entre 8-15%, lo cual es inaceptable en entornos de producción críticos [13].

Las herramientas nativas de AWS, como Amazon GuardDuty y AWS Security Hub, proporcionan detección de amenazas basada en anomalías de comportamiento, pero carecen de capacidades específicas para identificar transformaciones estructurales en archivos indicativas de cifrado por ransomware [14]. Esta brecha tecnológica deja una ventana crítica de vulnerabilidad entre el momento del ataque y su detección.

La revisión de la literatura demostró que la aplicación específica de SSDEEP para la detección temprana de ransomware en buckets de AWS S3 permanece inexplorada en la literatura académica. Se desconoce de estudios que evalúen la viabilidad operacional, escalabilidad, y eficacia de esta aproximación en entornos distribuidos de almacenamiento en la nube bajo condiciones de producción real.

Otros factores que pueden agudizar la problemática son:

- Volumen y velocidad de datos: Los buckets AWS S3 pueden contener petabytes de información con millones de objetos, lo que dificulta el monitoreo en tiempo real. El volumen global de datos creados, capturados, copiados y consumidos alcanzó 97 zettabytes en 2022, complicando los procesos de análisis forense [15].
- Diversidad de tipos de archivo: La heterogeneidad de formatos almacenados complica la implementación de estrategias de detección uniformes [16].
- Cifrado legítimo vs. malicioso: Distinguir entre procesos de cifrado autorizados y ataques de ransomware representa un desafío técnico considerable, dado que ambos producen transformaciones criptográficas similares [17].

Los siguientes informes de fabricantes y revistas especializadas en ciberseguridad, mediante diferentes métricas visibilizan año tras año las tendencias de ataques de ransomware. Sophos en su informe anual del 2024, ratifica como los malware de tipo ransomware siguen estando presente (ver figura 2), afectando a organizaciones de todos los tamaños y sectores [5].

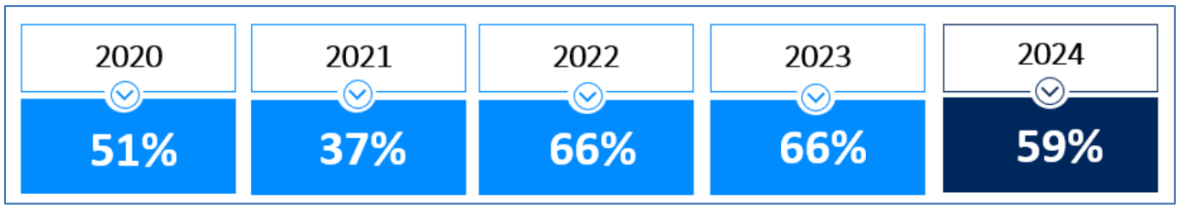

Fig. 2: *Fuente Informe Sophos "State of Ransomware 2024"*

La revista Cibercrimen en uno de sus últimos números prevé que las pérdidas económicas a futuro por los ataques de ransomware va en aumento [18]. Los costos asociados incluyen el pago de rescates, la recuperación de datos, la interrupción de operaciones y las multas

regulatorias. Además, las organizaciones pueden enfrentar pérdidas indirectas como la disminución de la confianza del cliente y el daño a la reputación. Los costos globalizados mantienen una tendencia creciente a la fecha y seguirán subiendo en el tiempo según se evidencia en la figura 3

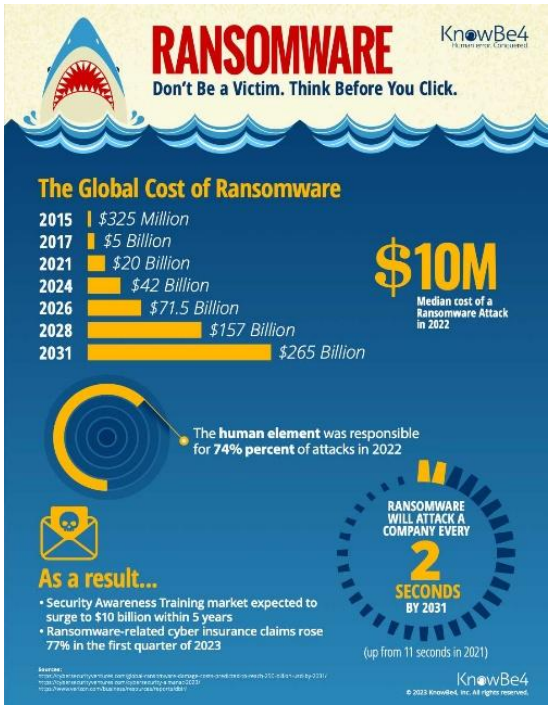

Fig. 3: *Fuente Informe KnowBe4 "Global Cost of Ransomware"*

En el informe de CrowdStrike "Global Threat Report 2025", se observa que las principales industrias afectadas por intrusiones interactivas durante el año 2024 fueron tecnología, consultoría y servicios profesionales, industria, y servicios financieros (ver Fig. 4). Este hallazgo es relevante para el contexto de la tesis, ya que destaca la criticidad de proteger entornos que manejan grandes volúmenes de datos y operaciones sensibles, como lo es el almacenamiento en la nube mediante AWS S3. [19]

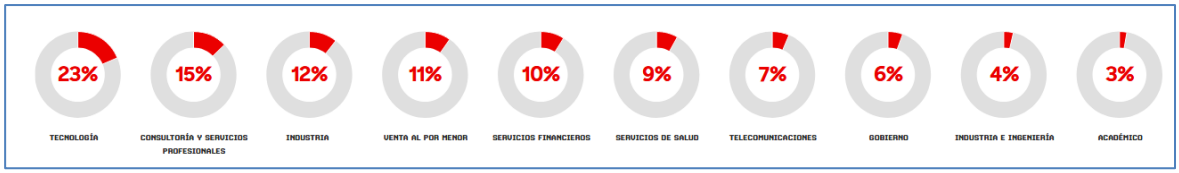

Fig. 4: *Fuente CrowdStrike "Industrias víctimas de malware, enero-diciembre del 2024"*

En el contexto actual, las organizaciones enfrentan amenazas cibernéticas cada vez más sofisticadas, siendo el ransomware una de las más prevalentes y dañinas. La migración de infraestructuras tecnológicas a la nube, aunque ofrece numerosos beneficios, también introduce nuevos vectores de ataque. Las pérdidas económicas derivadas de los ataques de ransomware subrayan la necesidad de implementar medidas para robustecer la seguridad en infraestructuras desplegadas en la nube. La implementación de un marco de seguridad como el propuesto en esta tesis es crucial para enfrentar las amenazas cibernéticas actuales.

Este documento presenta de manera estructurada el desarrollo y cumplimiento de los objetivos propuestos. Comienza con un marco teórico y un estado del arte que recopilan referencias clave para sustentar conceptualmente el estudio. Seguidamente, se detalla la metodología, dividida en tres fases: Fase I – Caracterización de patrones de ransomware; Fase II – Experimentación con reglas YARA integradas con el algoritmo SSDEEP y su despliegue en AWS Lambda; y Fase III – Análisis y evaluación de la metodología propuesta. A continuación, se exponen los resultados obtenidos en relación con cada uno de los objetivos específicos, incluyendo la identificación de las variantes de ransomware más prevalentes, el diseño de algoritmos que integran YARA y SSDEEP, el desarrollo de una función en AWS Lambda con ejecución automatizada sobre buckets de AWS S3, y las evidencias obtenidas a partir de las pruebas de funcionamiento. Para alcanzar estos resultados fue necesario el uso de herramientas tecnológicas como Python 3.9, Kali Linux, AWS S3, AWS EC2 y AWS Lambda. Finalmente, el documento concluye con un apartado de conclusiones y recomendaciones, las cuales pueden servir como punto de partida para futuras investigaciones en esta línea temática.

.

1 Marco Teórico y Estado del Arte

Patrón de comportamiento

En el contexto de las reglas YARA para esta investigación, los términos “patrón” o “patrón de comportamiento” se refiere a las secuencias específicas o características, que bien pueden ser cadenas de texto o cadenas hexadecimales (ver Fig. 5), [20] que un archivo puede contener y que son indicativas del comportamiento asociado con un ransomware específico [21].

```
$hex_string = { E2 34 ?? C8 A? FB }
$text_string = "foobar"
```

Fig. 5: *Patrones de comportamiento tipo hexadecimal y texto de reglas YARA*

Ransomware

El ransomware es un tipo de malware que secuestra los datos confidenciales o el dispositivo de una víctima, amenazando con mantenerlos bloqueados o divulgarlos, a menos que la víctima pague un rescate al atacante [22],[23]. Algunas de sus variantes notables son: CryptoLocker, WannaCry, Petya/NotPetya, Akira, Ryuk, DarkSide, Locky, REvil, Conti, LockBit.

La siguiente figura ilustra la anatomía de un ataque de ransomware (ver Fig. 6)

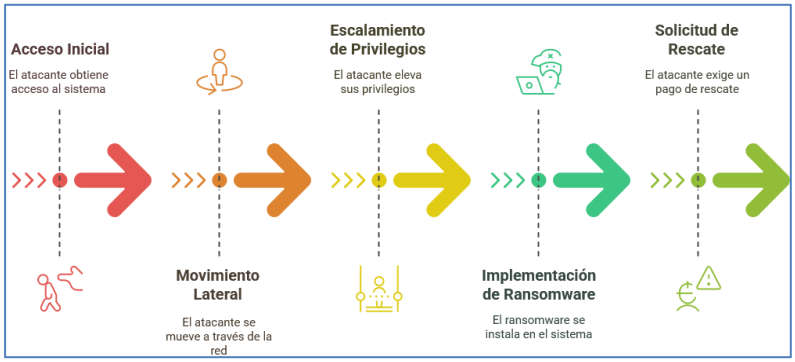

Fig. 6*: Anatomía de ataque de ransomware*

De acuerdo con los informes anuales de 2023 y 2024 de los fabricantes de herramientas de ciberseguridad Sophos y Trend Micro (ver Fig. 7 y 8), algunos de los virus más prevalentes en las organizaciones a nivel global son Akira y LockBit [24],[25]. Es especialmente relevante que ambos informes destacan a LockBit como el ransomware

más prevalente y Akira dentro de su top 10, lo cual coincide con los datos recopilados en la encuesta realizada para esta investigación (ver anexo A).

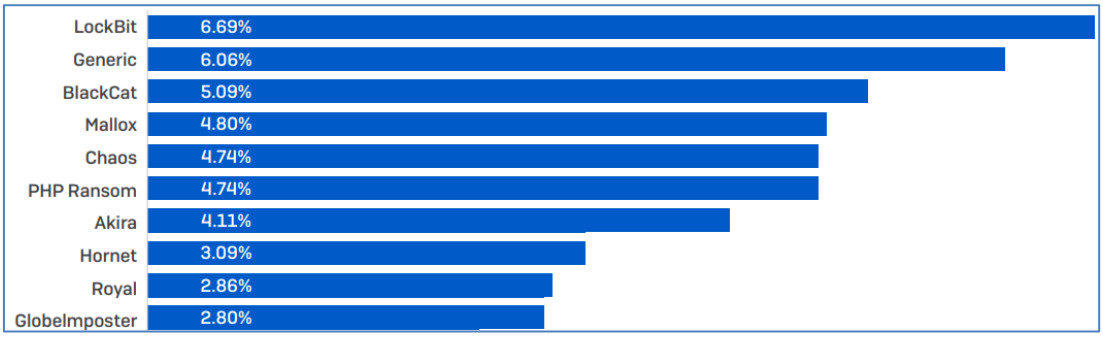

Fig. 7*: Fuente Sophos "Top 10 principales ransomware 2023"*

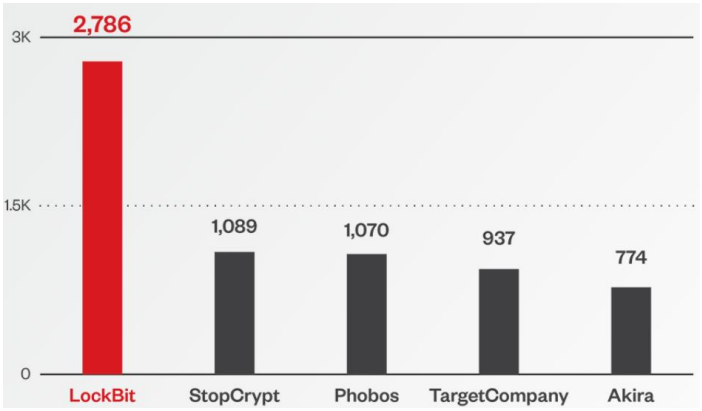

Fig. 8: Fuente *Tren Micro "Top 5 ransomware en 2024"*

Reglas YARA

Las reglas YARA son una herramienta utilizada ampliamente en el ámbito de la ciberseguridad para la identificación y clasificación de malware. Desarrolladas inicialmente por Víctor Álvarez de VirusTotal [26], estas reglas permiten a los analistas de seguridad describir patrones y características de archivos sospechosos mediante un lenguaje de scripting sencillo y flexible.

- **Estructura de las reglas YARA**

  Estas reglas están compuestas por tres secciones principales: meta, strings y conditions (ver Fig. 9) [20].

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"
    condition:
        $a or $b or $c
}
```

Fig. 9: *Estructura de una regla YARA*

- **Meta:** Define el nombre de la regla y sus metadatos, como la información del autor y la descripción especifica de la regla.
- **Strings:** Contiene las cadenas de texto o patrones binarios que se buscan en el archivo.
- **Conditions:** Especifica las condiciones lógicas que deben cumplirse para que la regla se considere coincidente.

Una vez construido el archivo con la regla YARA se procede a ejecutar sobre un archivo, una carpeta o una unidad de almacenamiento [27] en búsqueda de archivos que dentro de su contenido estructural coincidan con los “strings” definidos en la regla YARA (ver Fig. 10).

```
yara [OPTIONS] RULES_FILE TARGET
```

Fig. 10: *Sintaxis de ejecución de regla YARA*

De ser positiva la búsqueda, se informará de los archivos que hayan hecho match con los “strings” definidos en la regla YARA.

- **Principios de Desarrollo de Reglas YARA**

El desarrollo efectivo de reglas YARA requiere un enfoque equilibrado que combine IoC’s específicos, útiles para detectar variantes conocidas, con patrones genéricos basados en TTP’s, que ofrecen mayor resistencia ante variantes nuevas. Esta estrategia por capas mejora la capacidad de detección frente a amenazas en evolución.

- **Capa 1 (Alta Fidelidad):** Reglas que se dirigen a cadenas altamente únicas y codificadas, elementos de archivos específicos como sus nombres, cadenas hexadecimales o parámetros de línea de comandos únicos [28].
- **Capa 2 (Basadas en Comportamiento/TTP):** Reglas que se dirigen a comportamientos maliciosos comunes como comandos específicos de PowerShell para la eliminación de VSS o patrones de llamadas a la API para la evasión de defensas [29].

Fuzzy Hashing SSDEEP

"SpamSum" fue el nombre original del algoritmo de fuzzy hashing creado por Dr. Andrew Tridgell, utilizado inicialmente para detectar correos electrónicos similares (spam). Posteriormente, se desarrolló como una implementación más robusta y extendida de ese algoritmo, aplicable a archivos de cualquier tipo, no solo texto y pasó conocerse como "Similarity Digest DEEP" (SSDEEP) [30]. SSDEEP es una herramienta para calcular CTPH [31]. También llamados hashes difusos, los CTPH pueden hacer coincidir entradas que tienen homologías (ver Fig. 11). Dichas entradas tienen secuencias de bytes idénticos en el mismo orden, aunque los bytes entre estas secuencias pueden ser diferentes tanto en contenido como en longitud. [32]

- **Estructura de SSDEEP**

  La técnica que usa SSDEEP consiste en comparar archivos y detectar similitudes, incluso cuando hay cambios menores entre ellos [31]. Este método es especialmente útil para detectar variantes de malware.

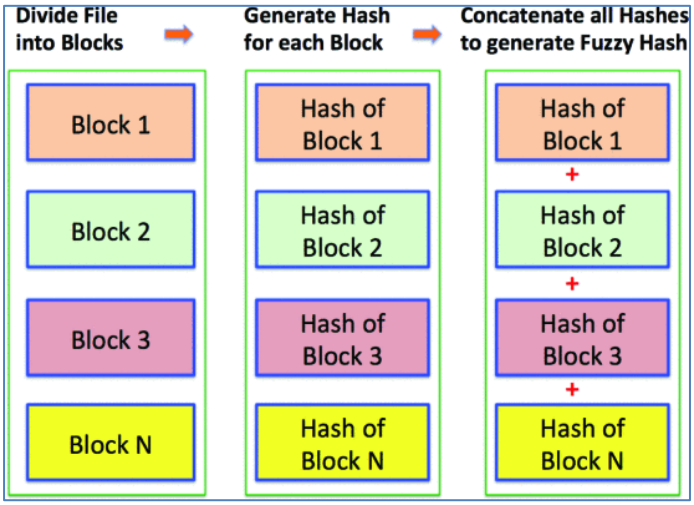

Fig. 11*: Procedimiento de Generación de fuzzy hash SSDEEP*

La anterior figura muestra el funcionamiento básico del fuzzy hashing SSDEEP que consístete en las siguientes fases:

- **Dividir el archivo en bloques:** El archivo original es dividido en varios bloques de datos consecutivos (Block 1, Block 2, ..., Block N). Esta división permite analizar el contenido de manera segmentada [31].
- **Generar hash para cada bloque:** A cada bloque se le aplica una función hash para obtener un valor hash individual (Hash of Block 1, Hash of Block 2, etc.). Estos hashes representan de manera única el contenido de cada bloque. Si un bloque cambia, su hash también cambia, pero el resto permanece igual si los otros bloques no fueron modificados [31].
- **Concatenar todos los hashes para generar el fuzzy hash SSDEEP:** Finalmente, los hashes individuales de cada bloque son concatenados (unidos en secuencia) para formar un fuzzy hash del archivo completo. A diferencia de los hashes criptográficos tradicionales (como SHA-256), que cambian completamente con cualquier alteración mínima del archivo, el fuzzy hash preserva la estructura parcial, esto permite comparar archivos y calcular su nivel de similitud [31].

- **Principios de Desarrollo de SSDEEP**

La principal ventaja de SSDEEP es su capacidad para identificar familias de malware o variantes que han sufrido modificaciones menores, como la ofuscación o la mutación, que alterarían completamente los hashes criptográficos [32]. Esto es crucial en un panorama de amenazas donde los autores de malware modifican constantemente sus

códigos para evadir la detección [33]. SSDEEP permite a los analistas atribuir muestras a grupos de amenazas específicos o identificar si diferentes variantes provienen del mismo constructor de malware, lo que proporciona una valiosa inteligencia de amenazas [34],[35].

AWS

Amazon Web Services (AWS) es una plataforma de servicios en la nube que ofrece infraestructura, almacenamiento, bases de datos y herramientas de cómputo escalables bajo demanda. Algunas de las herramientas relevantes para esta investigación son:

- **AWS S3**

  (Amazon Web Service Simple Storage Service) es "U*n servicio de almacenamiento de objetos creado para almacenar y recuperar cualquier volumen de datos desde cualquier ubicación.*" [36]. AWS S3 es un servicio elástico de pago por uso que permite a los usuarios crear depósitos llamados "bucket's", con el propósito de almacenar objetos de diversos tipos tales como; documentos, hojas de cálculo, imágenes, copias de seguridad, etc [37],[38]. (ver Fig. 12)

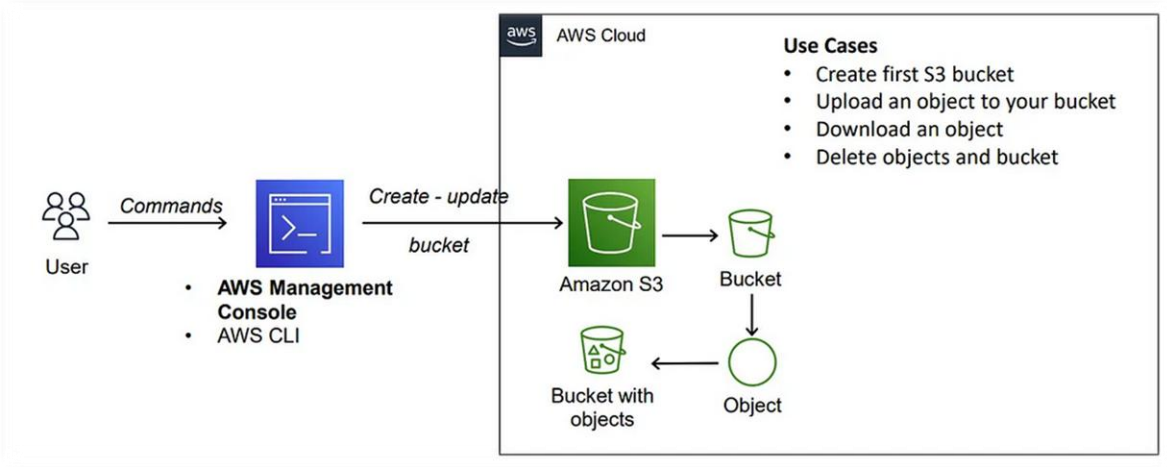

Fig. 12: *Esquema de uso de AWS S3*

- **AWS Lambda**

AWS Lambda es un "S*ervicio* de *computación* basado *en eventos que ejecuta instancias de una función para procesar eventos.*" (ver Fig. 13) [39],[40]

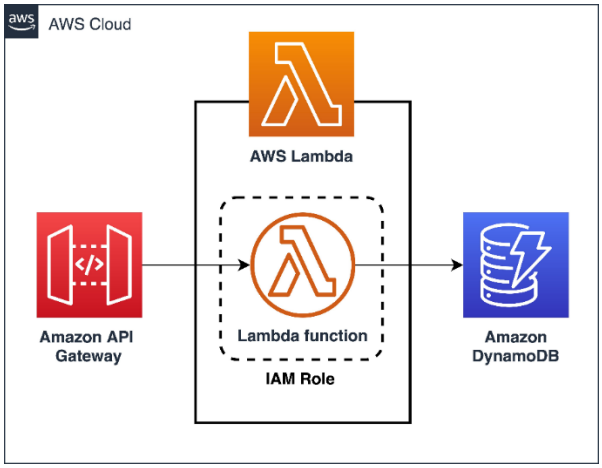

Fig. 13: *Diagrama de ejecución de una función AWS Lambda*

1.1 Caracterizar patrones de comportamiento de ransomware

Esta sección aborda la comprensión profunda de cómo el ransomware opera y cómo sus patrones de comportamiento pueden ser identificados dentro de los archivos, sentando las bases para el desarrollo de estrategias de detección efectivas.

1.1.1 El Ransomware como Amenaza Evolutiva

El ransomware es un tipo de malware que bloquea o amenaza con divulgar datos hasta que se pague un rescate. Su evolución constante, mediante técnicas como el polimorfismo y la ofuscación, le permite evadir métodos de detección tradicionales, lo que exige defensas más dinámicas y adaptables [5]. Informes recientes destacan a LockBit como la familia de ransomware más activa y dominante a nivel global, seguida de Akira, lo que refuerza la necesidad de enfoques innovadores ante una amenaza en constante transformación.

1.1.2 Enfoques de Detección de Malware: Estático, Dinámico y Híbrido

La detección de malware se basa en dos enfoques principales: estático y dinámico. El análisis estático examina archivos sin ejecutarlos, usando firmas, hashes y patrones predefinidos, siendo YARA una herramienta clave en este método. En contraste, el análisis

dinámico ejecuta el malware en entornos controlados (sandbox) para observar su comportamiento en tiempo real, siendo más eficaz frente a amenazas de día cero o malware polimórfico [41].

Ambos enfoques tienen limitaciones; el análisis estático es rápido pero vulnerable a técnicas de evasión, mientras que el dinámico es más preciso, aunque costoso en recursos. Por ello, se recomienda un enfoque híbrido, como el propuesto en esta tesis, que integra YARA con SSDEEP para añadir detección basada en similitud estructural, mejorando la capacidad de identificar variantes que escapan a las firmas tradicionales [32].

1.1.3 Reglas YARA para la Identificación de Patrones

YARA opera compilando las reglas y escaneando archivos o procesos en busca de coincidencias optimizadas, evaluando la lógica definida por el analista. Su capacidad de usar expresiones booleanas, modificadores (como nocase, wide, xor) y módulos especializados (como PE o Hash) permite un análisis detallado de archivos más allá de simples firmas [42].

A diferencia de los antivirus tradicionales, YARA permite combinar múltiples criterios para detectar no solo coincidencias exactas, sino también variaciones estructurales o comportamentales del malware. Esta flexibilidad lo convierte en un motor de firmas inteligente, ideal para detectar variantes de ransomware que comparten patrones comunes [42].

1.2 Técnicas de detección y análisis de malware

El incremento del malware y su capacidad para evadir la detección representan una amenaza crítica para la seguridad informática dado el continuo surgimiento de nuevas familias de ransomware o nuevas variantes de las familias ya existentes (ver Fig. 14). Aunque existen diversos enfoques, como los basados en firmas, comportamiento, nube y aprendizaje profundo, ninguno logra detectar todo tipo de malware [43]. Esto evidencia la necesidad de investigaciones continuas y métodos innovadores.

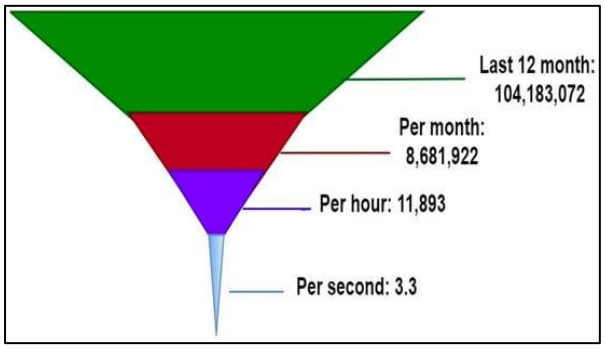

Fig. 14: *Surgimiento de nuevas amenazas en el tiempo*

Este documento revisa algunos de los enfoques de detección de malware y los métodos recientes que utilizan estos enfoques.

1.2.1 Basadas en Machine Learning

El machine learning (ML) permite a los sistemas aprender de los datos históricos y mejorar su capacidad de detección con el tiempo. Algoritmos como los árboles de decisión, las máquinas de vectores de soporte (SVM) y los bosques aleatorios (RF), se utilizan para analizar características estáticas y dinámicas del malware, permitiendo la identificación de patrones maliciosos que no son evidentes a simple vista [44]. Estos algoritmos pueden clasificar el software como benigno o malicioso basándose en características previamente etiquetadas, lo que mejora la precisión y reduce los falsos positivos. Además, el ML es eficaz para detectar malware polimórfico, que cambia constantemente su firma para evadir los métodos de detección tradicionales [43].

Para el análisis y detección de malware’s basados en algoritmos de machine learning existen diferentes técnicas. Algunas de ellas se pueden ver en el siguiente gráfico: (ver Fig. 15)

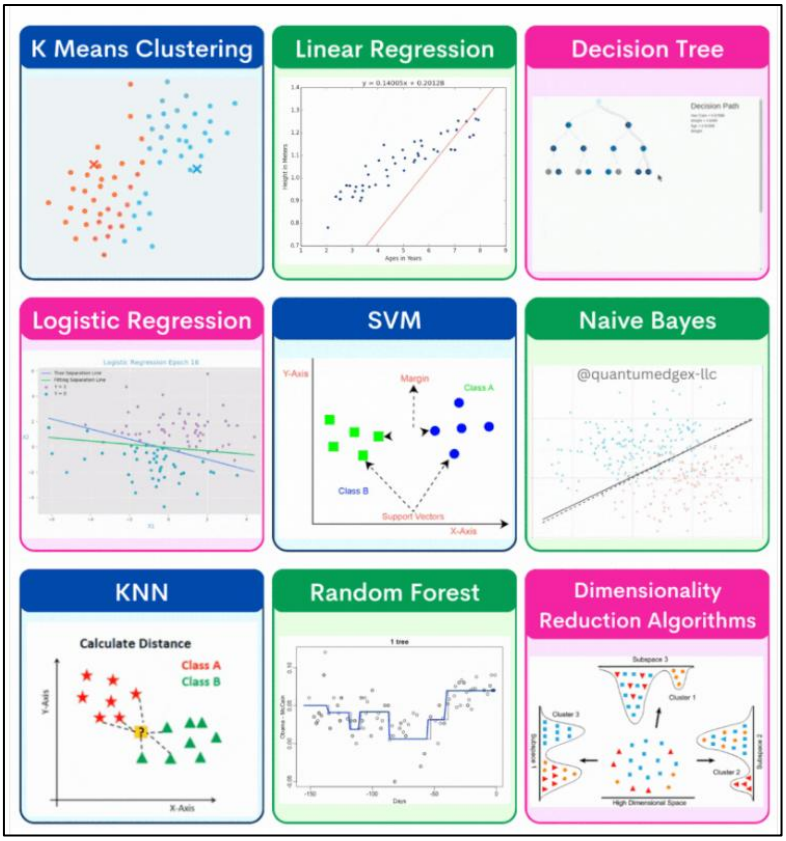

Fig. 15: *Técnicas de detección de maching learning*

El deep learning (DL), una subdisciplina del ML que utiliza redes neuronales profundas para modelar y entender patrones complejos en grandes volúmenes de datos [45]. Las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN) son particularmente efectivas en la clasificación de malware [45]. Las CNN pueden analizar imágenes de archivos ejecutables convertidos en escala de grises para identificar características visuales distintivas del malware [46]. Por otro lado, las RNN son útiles para capturar patrones secuenciales en el comportamiento del malware, lo que permite una detección más precisa de amenazas avanzadas [47]. Estas técnicas de DL no solo mejoran la precisión de la detección, sino que también reducen significativamente el número de falsos positivos, lo que es crucial para la eficiencia operativa de los sistemas de seguridad [48]. Por otro lado, el clustering, una técnica de ML no supervisado que agrupa datos en clústeres basados en similitudes intrínsecas [48]. En la detección de malware, el clustering

se utiliza para identificar comportamientos sospechosos agrupando muestras de malware con características similares [49]. Algoritmos como K-means y Expectation Maximization (EM) son comunes en este campo. Estos métodos permiten a los analistas de seguridad descubrir nuevas variantes de malware y entender mejor las relaciones entre diferentes tipos de amenazas. Además, el clustering puede ayudar a reducir la carga de trabajo al priorizar las muestras más sospechosas para un análisis más detallado. Esta técnica es especialmente útil para la detección de malware desconocido o de día cero, ya que puede identificar anomalías sin necesidad de firmas predefinidas [50].

1.2.2 Basadas en Reglas YARA

YARA funciona mediante reglas estructuradas que combinan cadenas textuales, hexadecimales o expresiones regulares con condiciones booleanas (ver Fig. 16). Estas reglas se compilan y aplican a archivos o procesos para detectar coincidencias. Además, YARA admite módulos adicionales (como PE, ELF, Hash) que amplían sus capacidades de inspección, lo que permite un análisis más detallado de distintos formatos y atributos del archivo [41].

A diferencia de las firmas antivirus tradicionales, que dependen de coincidencias exactas, YARA permite definir comportamientos o combinaciones de características, lo que la hace más resistente frente a variantes levemente modificadas. Sin embargo, su enfoque estático tiene limitaciones frente a malware polimórfico, amenazas de día cero o técnicas de evasión avanzadas, las cuales solo se evidencian mediante ejecución real, como ocurre en el análisis dinámico [51].

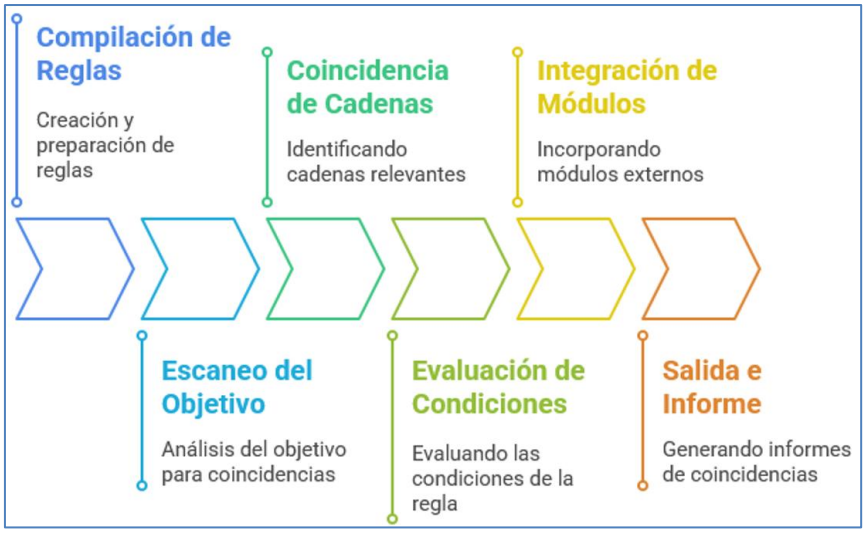

Fig. 16*: Proceso de operación de regla YARA*

1.2.3 Basadas en AI

El malware polimórfico, metamórfico y las amenazas de día cero representan un gran desafío para las técnicas tradicionales de detección basadas en firmas, ya que estas variantes modifican su código constantemente o explotan vulnerabilidades desconocidas, haciendo que las firmas queden obsoletas casi de inmediato [52]. Además, el volumen y la velocidad con que surgen nuevas amenazas supera la capacidad de actualización manual, generando una brecha crítica en las defensas [53].

Frente a este panorama, la Inteligencia Artificial (IA) ha emergido como una herramienta fundamental en ciberseguridad, ofreciendo capacidades superiores para detectar, prevenir y mitigar amenazas en tiempo real. Su fortaleza radica en la habilidad de procesar grandes volúmenes de datos, identificar patrones complejos y aprender continuamente, lo que permite anticipar ataques en lugar de simplemente reaccionar a ellos. En su modalidad supervisada, los modelos se entrenan con ejemplos etiquetados (como correos maliciosos o benignos), mientras que, en la no supervisada, se detectan patrones o anomalías en datos sin etiquetar, útil para descubrir amenazas desconocidas. [54]

La siguiente tabla (ver tabla 2) detalla los principales algoritmos o técnicas IA/ML/DL para la detección de ransomware, su aplicación de detección y las ventajas

Tabla 2*: Algoritmos de IA/ML/DL y sus Aplicaciones en la Detección de Malware*

Algoritmo/Técnica	Tipo de IA	Aplicación Principal en Detección de Malware	Ventaja/Mecanismo Clave
Máquinas de Soporte Vectorial (SVM)	ML (Supervisado)	Clasificación de archivos (benigno/malicioso), Detección de anomalías	Maximiza el margen de separación en espacios de alta dimensión; eficaz con datos no estructurados y menos propenso al sobreajuste.
Árboles de Decisión y Bosques Aleatorios	ML (Supervisado/Ensamble)	Clasificación de archivos, Análisis de comportamiento, Detección de anomalías	Maneja datos heterogéneos; Bosques Aleatorios mejoran precisión y reducen sobreajuste; alta interpretabilidad.
Algoritmos de Agrupación (Clustering)	ML (No Supervisado)	Agrupación de familias de malware, Detección de variantes desconocidas	Identifica patrones y similitudes inherentes en datos sin etiquetas, útil para malware nuevo.
Redes Neuronales Convolucionales (CNN)	DL	Análisis de código binario (como imágenes), Detección de patrones en código	Extrae características de datos no estructurados; eficaz contra malware polimórfico al tratar el código como imagen.
Redes Neuronales Recurrentes (RNN) y LSTM	DL	Análisis de comportamiento secuencial (tráfico de red, llamadas API)	Procesan secuencias de datos; detectan patrones anómalos y ataques de día cero al identificar dependencias temporales.
Análisis de Comportamiento (UEBA)	ML/DL	Detección de amenazas internas, Identificación de actividades sospechosas	Establece líneas base de comportamiento normal; detecta desviaciones sutiles y actividades anómalas en tiempo real.
Análisis Estático/Dinámico Mejorado por IA	ML/DL	Examen de código sin ejecución, Observación de comportamiento en sandbox	Extrae atributos de código y monitorea runtime; acelera la comprensión del malware y su impacto.

1.4 Impactos de los ataques

Los ataques de ransomware no solo representan una amenaza técnica, sino que acarrean consecuencias multifacéticas que afectan la operatividad, la economía, la reputación y el marco legal de las organizaciones [5]. A continuación, se describen los principales impactos que una empresa padece cuando es atacado por un ataque:

1.4.1 Consecuencias Operacionales y de Negocio

Los ataques de ransomware tienen un impacto severo en la continuidad operativa de las organizaciones, provocando interrupciones críticas, tiempos de inactividad prolongados y

dificultades para restaurar el acceso a sistemas y datos esenciales [55]. Casos como el de JBS USA evidencian que el ransomware no es solo un problema de seguridad, sino una amenaza directa al negocio [56]. Además, estos ataques conllevan la pérdida y el compromiso de datos sensibles mediante cifrado y exfiltración, lo que agrava el riesgo al afectar tanto la disponibilidad como la confidencialidad de la información. Por ello, las estrategias defensivas deben centrarse no solo en la detección, sino también en la resiliencia y la recuperación ante desastres [57].

1.4.2 Impacto Financiero del Ransomware

Los ataques de ransomware generan una carga financiera significativa para las organizaciones, la cual se divide en costos directos e indirectos, formando una red de consecuencias que afecta tanto la operación inmediata como la estabilidad a largo plazo [58].

Entre los costos directos, destaca el pago del rescate, usualmente exigido en criptomonedas, entendiendo este gasto no garantiza la recuperación de los datos cifrados. Además, se deben considerar los costos de recuperación, que incluyen la restauración de sistemas desde copias de seguridad, la contratación de expertos en ciberseguridad, análisis forenses y la implementación de nuevas medidas de protección. En promedio, el costo total para mitigar un ataque de ransomware alcanzó los 1.27 millones de dólares en 2020 [5].

En cuanto a los costos indirectos, estos afectan profundamente la continuidad operativa y la salud financiera de la organización. La pérdida de ingresos y productividad causada por la interrupción de servicios puede extenderse más allá del tiempo de inactividad inicial, afectando procesos clave durante días o incluso semanas [18]. A esto se suman los daños reputacionales, que influyen negativamente en la confianza de los clientes y en la capacidad de la empresa para generar nuevos negocios. También se presentan costos legales derivados de multas, demandas y auditorías regulatorias tras la exposición o pérdida de datos [18].

Otros costos relevantes incluyen la sustitución de dispositivos comprometidos, capacitación del personal, y la inversión en nuevas tecnologías de seguridad para prevenir futuros incidentes. Este conjunto de impactos desencadena un efecto dominó financiero, donde un solo ataque puede derivar en pérdidas multimillonarias a nivel global [21].

Por ello, se recomienda que las organizaciones adopten un enfoque de Costo Total de Propiedad (TCO) para su estrategia de ciberseguridad, considerando no solo la inversión inicial, sino los riesgos y pérdidas evitables en caso de ataque [58].

1.4.3 Daño Reputacional y Pérdida de Confianza

El daño a la reputación es una de las consecuencias más críticas de un ataque de ransomware, ya que afecta directamente la confianza del público y la percepción de la marca. Cuando se comprometen datos sensibles, los clientes pierden confianza en la capacidad de la organización para proteger su información, lo que puede provocar la pérdida de clientes actuales y dificultar la captación de nuevos. Incluso sin filtración de datos, la sola noticia del incidente puede generar dudas sobre la solidez de la ciberseguridad empresarial [18]. Además, la interrupción de servicios acentúa esta percepción negativa. Por ello, las organizaciones deben priorizar la gestión de crisis y la comunicación transparente para mitigar el impacto reputacional y recuperar la confianza del cliente. [59]

1.4.4 Repercusiones Legales y Regulatorias

Los ataques de ransomware generan serias implicaciones legales y regulatorias, especialmente cuando se ven comprometidos datos personales. Las organizaciones deben cumplir con normativas como el GDPR o la CCPA, y el incumplimiento puede acarrear multas severas, como en el caso de British Airways [60]. Las obligaciones de notificación a autoridades y usuarios afectados son críticas, y su omisión agrava las consecuencias legales. Además, el pago de rescates plantea dilemas éticos y legales, sobre todo si se realiza a grupos sancionados, lo que puede implicar responsabilidades adicionales sin garantía de recuperación de datos [61]. Estos factores subrayan la necesidad de contar con asesoría legal especializada y planes de respuesta que consideren las consecuencias jurídicas de cada decisión.

La siguiente figura (ver Fig. 17) representa los efectos que ocasiona un ransomware al materializar su ataque en las organizaciones

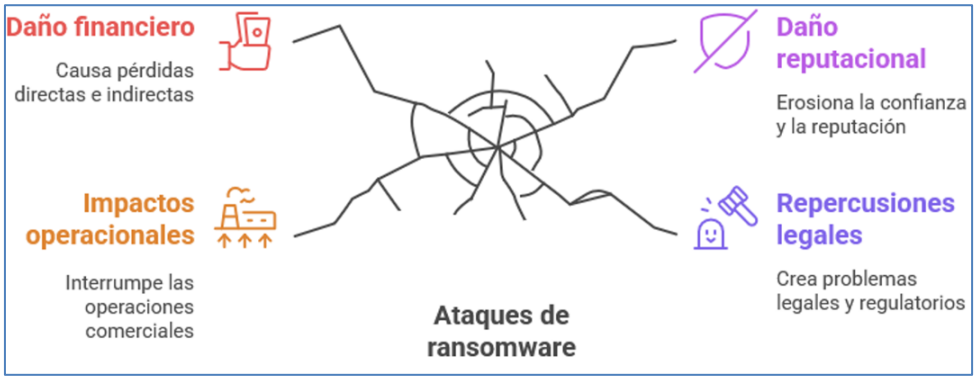

Fig. 17*: Impacto del ransomware en las organizaciones*

1.5 Estado del arte

Para este proyecto se hizo una búsqueda exhaustiva de información en diferentes bases de datos científicas bibliográficas, paginas oficiales de fabricantes y repositorios documentales de universidades del país. Algunas de las bases de datos científicas consultadas fueron: IEEE, Scopus, y Science Direct. Las páginas de fabricantes consultados fueron: MalwareBazar, AWS, VirusTotal y la página oficial de YARA. Por último, algunos de los repositorios documentales de tesis de universidades del país fueron: ITM y TdeA

Principalmente y como eje central del proyecto, las búsquedas estuvieron enfocadas en reglas YARA. Sin embargo, otros conceptos tenidos en cuenta al realizar las búsquedas fueron: malware, contenedores de AWS S3, fuzzy hashing SSDEEP y la ejecución automatizada de funciones AWS Lambda.

La tesis de grado "Detección de amenazas informáticas de tipo Malware Bancario o Ransomware Móvil hacia dispositivos Android, integrando IoC en una técnica semiautomatizada y con base en comportamientos analizados de incidentes" [62], publicada por el ITM en el año 2022 de maestría en ciberseguridad, tiene como principal tema de investigación la detección de malware y/o ransomware, y como los IoC ayudan a fortalecer la seguridad. En esta se menciona la relevancia que las reglas YARA tienen a la

hora de identificar archivos que contengan patrones de ransomware, además de la versatilidad de estas reglas ejecutándose en sistemas operativos Android.
La investigación “Detection of Malware by Using YARA Rules” de 2024 define como las reglas YARA implementadas de forma estática demuestran efectividad para identificar archivos con patrones que correspondan a ciertos tipos de malware [63]. Adicionalmente en esta investigación se demostró como en ciertos tipos de archivos los primeros bytes son estáticos lo que demuestra un patrón dentro del archivo que garantiza que comienza con los mismos bytes cada vez. Este comportamiento se puede tener en cuenta en las reglas YARA para fortalecer la identificación de patrones maliciosos dentro de los archivos.

Las reglas YARA pueden ser creadas automáticamente, y en la investigación “Evaluating Automatically Generated YARA Rules and Enhancing Their Effectiveness” [64] publicado el 2021, se analizan 3 formas. Una de estas formas plantea mejorar las reglas YARA generadas automáticamente utilizando el algoritmo de “hashing difuso SSDEEP” [32]. Como resultado de este análisis se demostró que la ejecución de las reglas YARA regulares tienen unos resultados porcentualmente más bajos que si se ejecutan integradas con SSDEEP.

La creación y ejecución de reglas YARA supone una mejora a la capa de seguridad del entorno en el que se deseen implementar, conllevando a una eficacia en el número de detecciones. En la investigación “Fuzzy Hashing Aided Enhanced YARA Rules for Malware Triaging” [65] de 2019, se propone incrementar la efectividad de las reglas YARA adicionando algoritmos “hashing difuso SSDEEP”, el cual resultó ser efectivo según las pruebas realizadas y los resultados obtenidos.

Apoyando lo anterior, en la investigación “Embedding Fuzzy Rules with YARA Rules for Performance Optimisation of Malware Analysis” de 2023, se concluye que la integración de reglas YARA con algoritmos fuzzy hashing, da resultados de detección más eficientes frente a las reglas convencionales [28], denotando además la adaptabilidad y flexibilidad de su uso.

El método de fuzzy hasing es abordado en las investigaciones “Cyberthreat Hunting - Part 1: Triaging Ransomware using Fuzzy Hashing, Import Hashing and YARA Rules” [66] y “Cyberthreat Hunting - Part 2: Tracking Ransomware Threat Actors using Fuzzy Hashing

and Fuzzy C-Means Clustering" [67] de 2019. En estas investigaciones cobra relevancia la comparación de archivos mediante fuzzy hashing para determinar el porcentaje de similitud entre ellos, reafirmando con esto la aplicabilidad que tiene este método a la hora de determinar la similitud que tiene un patrón de ransomware conocido, versus archivos que posiblemente estén infectados con ese ransomware en particular. Otro factor importante en este par de investigaciones es el precedente de aplicabilidad de diferentes fuzzy hashing, no solo haciendo mención de SSDEEP, que es un elemento valioso de esta investigación, sino que también menciona el SDHASH; un fuzzy hashing también usado para comparar el porcentaje de similitud entre archivos que puede ser abordado en futuras investigaciones.

La automatización de la detección de malware mediante el uso de Reglas YARA y AWS Lambda ofrece una solución eficiente y escalable. AWS Lambda permite ejecutar código en respuesta a eventos, lo que facilita la integración de reglas YARA para escanear archivos almacenados en AWS S3 de manera automática y en tiempo real [68],[69]. Esta combinación de tecnologías no solo mejora la precisión y velocidad de la detección de malware, también reduce la carga operativa y los costos asociados con la gestión manual de la seguridad.

La eficiencia de los servicios AWS Lambda son abordados en la investigación "Securing AWS Lambda: Advanced Strategies and Best Practices" publicada desde un enfoque de ciberseguridad allanando el camino a la comprensión del funcionamiento de este servicio [70]. En este proyecto se exploró los riesgos de seguridad a los que están expuestas las ejecuciones de servicio automatizados sin servidor y se propone como base para ser implementado en otras nubes como GCP (Google Platform Cloud) o Microsoft Azure.

La investigación "Securing AWS Lambda: Advanced Strategies and Best Practices" [70] de 2024 tiene como principal foco las mejores prácticas de implementación de automatizaciones mediante AWS Lambda. Menciona como los usuarios deben enfocarse en temas como la lógica del negocio y no en tareas que fácilmente pueden ser automatizadas, elevando la productividad tanto de los usuarios como de las tareas programadas. Estas funciones AWS Lambda deben ser creadas implementando

estrategias que aseguren su ejecución contemplando las principales amenazas como la inyección de código malicioso, fugas de datos confidenciales, ataques DDoS, privilegios excesivos, dependencias vulnerables y problemas de certificados.

Después de ahondar en los diversos artículos de la literatura descrita en la bibliografía, se puede ver claramente que no existe un marco referencial de seguridad basado en reglas YARA incorporando algoritmos fuzzy hashing SSDEEP que permita identificar patrones de malware del tipo ransomware en los servicios de AWS S3, y que esté automatizado mediante AWS Lambda.

Para una mayor claridad en el proceso de selección y exclusión de fuentes, se incorporó un diagrama PRISMA (ver Fig. 18) que ilustra de manera estructurada las etapas de identificación, cribado, elegibilidad e inclusión de los estudios analizados.

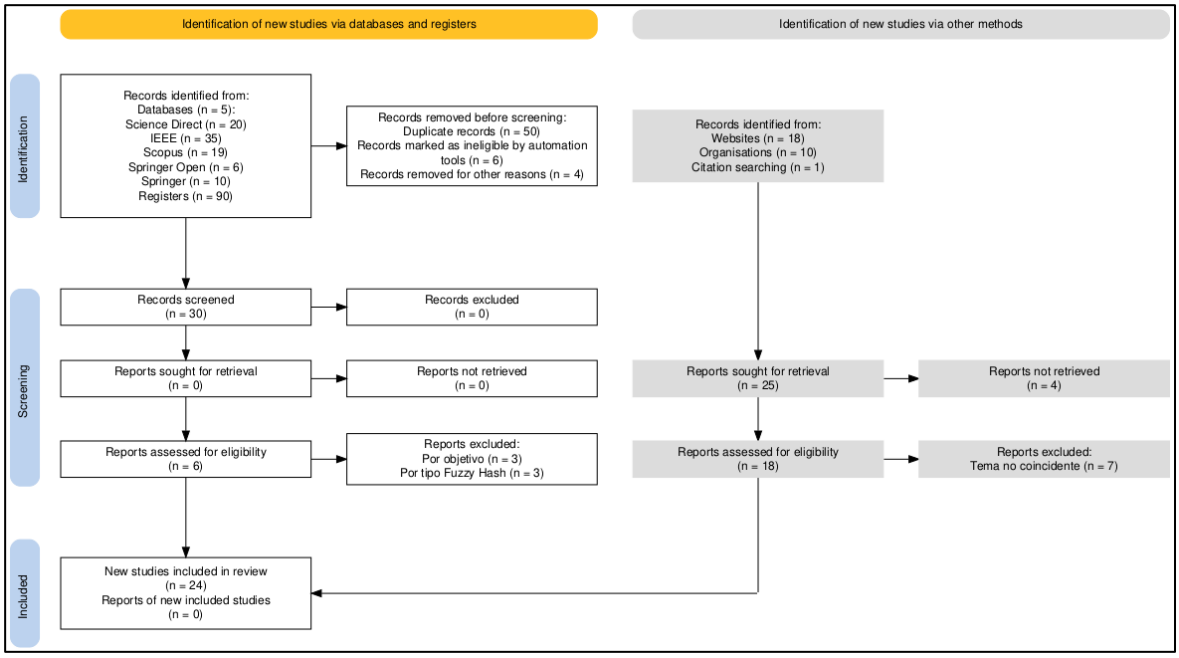

Fig. 18*: Diagrama PRISMA*

2 Metodología

El desarrollo de esta metodología se estructuró en tres fases (Fase I – Caracterización de patrones YARA de ransomware, Fase II – Experimentación YARA+SSDEEP e implementación en ASW Lambda y Fase III – Análisis y Evaluación de la Efectividad de la

metodología propuesta), siguiendo un enfoque de investigación cuantitativo y una experimentación de apoyo a la conceptualización. Para la recolección de datos se aplicó como instrumento un formulario de encuesta en una muestra de 160 empresas a diferentes sectores algunos de ellos como; Financieros, Telecomunicaciones, Energía, entre otros. Para el procesamiento y análisis estadístico se calculó una matriz de correlación tomando como referencia los datos de la encuesta y aplicando la técnica de Pearson por medio del software estadístico SPSS.

2.1 Fase I: Caracterización de Patrones YARA de Ransomware

Para el cumplimento del objetivo uno se aplicó la técnica de Pearson sobre una muestra de 160 respuestas obtenidas mediante un formulario compartido con empresas de diferentes sectores de Hispanoamérica (ver anexo A). Se calculó mediante una matriz de correlación para facilitar el análisis y determinar sí existe una relación entre las características de las organizaciones con los diferentes tipos de ransomware y su afectación. Los datos recolectados fueron procesados y analizados mediante el software estadístico SPSS, lo que permitió obtener resultados objetivos y sustentados.

Las respuestas obtenidas a través del formulario mencionado podrían presentar un sesgo relacionado con el perfil de los participantes, ya que no todas las respuestas provienen necesariamente de profesionales vinculados directamente al ámbito de la ciberseguridad. Esta variabilidad en los perfiles podría influir en la precisión y profundidad de la información proporcionada, afectando la interpretación de los resultados.

Esta fase se enfoca en identificar las amenazas de ransomware más relevantes en la región de Hispanoamérica y extraer inteligencia procesable para el desarrollo de reglas YARA basado en lo siguiente:

2.1.1 Análisis Cuantitativo de la Prevalencia de Ransomware

a) **Identificación de Familias de Ransomware más Frecuentes:** Se llevó a cabo un análisis cuantitativo de la muestra para determinar la frecuencia de cada tipo de ransomware reportado. Esto implica el conteo de las ocurrencias de cada nombre de malware de la información recolectada por la encuesta. Para efectos de esta

investigación, los laboratorios y pruebas se realizaron con los 2 ransomware que tuvieron la puntuación más alta.

b) Derivación de (IoC's) y TTPs: Basándose en las dos familias de ransomware prevalentes identificados, se realizó una revisión detallada de sus Tácticas, Técnicas y Procedimientos (TTPs) e Indicadores de Compromiso (IoC's) utilizando fuentes externas de inteligencia de amenazas, como avisos de CISA, artículos académicos e informes de la industria. Estos patrones específicos, que incluyen cadenas de texto, patrones hexadecimales y comportamientos, sirvieron como la base fundamental para el diseño de las reglas YARA en la Fase 2.

c) **Definición de Variables:** Las variables independientes y dependientes aplicadas en el escenario fueron las siguientes:

i. Variables Dependientes (Tipos de Ransomware)

- **Y1**: Incidencia LockBit (0=No afectado, 1=Afectado)
- **Y2**: Incidencia Akira (0=No afectado, 1=Afectado)
- **Y3**: Incidencia Conti (0=No afectado, 1=Afectado)
- **Y4**: Incidencia BlackCat/ALPHV (0=No afectado, 1=Afectado)
- **Y5**: Incidencia REvil/Sodinokibi (0=No afectado, 1=Afectado)
- **Y6**: Incidencia Ryuk (0=No afectado, 1=Afectado)

ii. Variables Independientes (Características Organizacionales)

- **X1: Sector Económico**
  - 1 = Servicios Financieros
  - 2 = Manufactura
  - 3 = Salud
  - 4 = Educación
  - 5 = Gobierno
  - 6 = Retail/Comercio
  - 7 = Tecnología
  - 8 = Energía
- **X2: Tamaño Organizacional**
  - 1 = Pequeña (1-50 empleados)
  - 2 = Mediana (51-250 empleados)

- 3 = Grande (251-1000 empleados)
- 4 = Muy Grande (>1000 empleados)

- **X3: Tipo de Operación**
  - 1 = Local
  - 2 = Nacional
  - 3 = Regional
  - 4 = Internacional
- **X4: Tipos de Documentos (Índice compuesto 1-10)**
  - Basado en sensibilidad de datos manejados
- **X5: Base de Clientes**
  - 1 = B2B exclusivo
  - 2 = B2C exclusivo
  - 3 = Mixto B2B/B2C
  - Variables Continuas
- **X6: Años de Operación (variable continua)**
- **X7: Volumen de Transacciones Digitales Diarias (escala logarítmica)**
- **X8: Nivel de Digitalización (escala 1-10)**

d) Aplicación del Método de Pearson

- **Comando SPSS Utilizado**

```
CORRELATIONS
  /VARIABLES=LockBit Akira Conti BlackCat REvil Ryuk Sector Tamaño
Operación TipoDoc BaseClientes AñosOp VolTransDig NivelDigit
  /PRINT=TWOTAIL NOSIG
  /MISSING=PAIRWISE.
```

- **Supuestos Verificados**
  - **Normalidad**: Test de Kolmogorov-Smirnov aplicado
  - **Linealidad**: Verificada mediante diagramas de dispersión
  - **Homogeneidad**: Test de Levene confirmado

- **n = 200** (tamaño muestral adecuado para correlaciones de Pearson)

2.1.2 Identificación de Patrones de Comportamiento

Analizada la información y establecido que los ransomware con más presencia son **LockBit y Akira**, se caracterizó cada uno de estos para determinar cuáles son sus patrones de comportamiento relevantes:

2.1.2.1 Caracterización de LockBit

Para caracterizar los patrones de comportamiento de LockBit, es fundamental tener en cuenta lo siguiente:

a) TTPs considerados para caracterizar LockBit:

- **Acceso Inicial y Movimiento Lateral**: LockBit busca propagarse a través de la red de la víctima utilizando una lista preconfigurada de credenciales codificadas en el momento de la compilación o cuentas locales comprometidas con privilegios elevados [71].
- **Evasión de Defensas**: LockBit cifra los datos, pero omite los archivos asociados con las funciones centrales del sistema para mantener la estabilidad del sistema y evitar la detección.[71]
- **Impacto y Deterioro del Sistema**: LockBit cifra los datos guardados en cualquier dispositivo local o remoto (MITRE ATT&CK T1486). Utiliza Windows Management Instrumentation (WMI) para consultar, obtener la ID y eliminar las Copias de Sombra de Volumen, lo que dificulta la recuperación del sistema. [71],[29]
- **Comando y Control (C2) y Exfiltración de Datos**: LockBit puede enviar información cifrada del host y del bot a un servidor C2 mediante solicitudes HTTP POST.[71],[72]

b) Indicadores IOC considerados para caracterizar LockBit:

- **Nota de Rescate**: <*Ransomware ID*>*.README.txt*
- **Parámetros de Línea de Comandos**: LockBit utiliza parámetros distintivos para controlar su comportamiento: -del (autoeliminación), *-gdel* (eliminar cambios de política de grupo), *-gspd* (propagación vía política de grupo), *-pass* (contraseña requerida de 32 caracteres), *-path* (cifrar archivo/carpeta específica), *-psex* (propagación vía recursos compartidos de administrador), *-safe* (reiniciar en Modo Seguro), *-wall* (establecer fondo de pantalla/imprimir nota) [71].

- **Elementos del Registro**: Lockbit realiza modificaciones en el registro para su persistencia y evasión. Esto incluye cambios en *HKCR\. <Malware Extension>, HKCR\<Malware Extension>\DefaultIcon* y *HKCU\Control Panel\Desktop\WallPaper* para el icono y el fondo de pantalla de LockBit 3.0. También modifica *HKLM\SOFTWARE\Policies\Microsoft\Windows\System, HKLM\SOFTWARE\Policies\Microsoft\Windows Defender* y *HKLM\SOFTWARE\Policies\Microsoft\WindowsFirewall* para deshabilitar SmartScreen, Windows Defender y el Firewall de Windows. Habilita el inicio de sesión automático a través de las claves *SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon (AutoAdminLogon, DefaultUserName, DefaultDomainName, DefaultPassword)*. Además, deshabilita y borra los Registros de Eventos de Windows modificando *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Channels**.
- **Listas de Terminación de Procesos/Servicios**: LockBit tiene listas específicas de servicios como *vss, sql, sophos* y procesos como *sql, oracle, outlook, winword* que intenta terminar para maximizar el cifrado.
- **Elementos de Red**: Las solicitudes HTTP POST a servidores C2 que contienen información cifrada del host y del bot, junto con cadenas de agente de usuario específicas *Mozilla/5.5 (Windows NT 6.1), Chrome/91.0.4472.77*, son elementos de red relevantes.

2.1.2.2 Caracterización de Akira

Para caracterizar los patrones de comportamiento de Akira, es fundamental tener en cuenta lo siguiente:

a) TTPs considerados para caracterizar Akira:

- **Acceso Inicial:** Akira obtiene acceso inicial principalmente explotando servicios de VPN que carecen de Autenticación Multifactor (MFA). Con frecuencia, aprovechan vulnerabilidades conocidas de Cisco, específicamente CVE-2020-3259 y CVE-2023-20269. [73],[74]
- **Persistencia y Descubrimiento:** Una vez establecido el acceso inicial, Akira buscan mantener la persistencia y descubrir información de la red.

Crea nuevas cuentas de dominio como una cuenta administrativa llamada *itadm* para establecer persistencia.[73]

- **Evasión de Defensas:** En ocasiones se despliega dos variantes de ransomware distintas (Megazord, específico de Windows, y una variante del cifrador Akira ESXi) contra diferentes arquitecturas de sistema dentro del mismo evento de compromiso. Deshabilitan el software de seguridad utilizando herramientas como PowerTool para explotar el controlador Zemana AntiMalware y terminar los procesos relacionados con el antivirus.[73]
- **Comando y Control (C2) y Exfiltración de Datos:** Akira establece canales C2 y exfiltra datos utilizando herramientas fácilmente disponibles. Aprovechan herramientas como AnyDesk, MobaXterm y RustDesk para el acceso y control remoto. Utilizan Ngrok y Cloudflare Tunnel para establecer túneles seguros para la exfiltración de datos a través de varios protocolos (FTP, SFTP) y servicios de almacenamiento en la nube como Mega.
- **Impacto y Cifrado:** Akira utilizan un modelo de doble extorsión, cifrando los sistemas después de exfiltrar los datos.

b) Indicadores IoC considerados para caracterizar Akira:

- **Extensiones de Archivo:** Los archivos cifrados se adjuntan con extensiones distintivas: *.akira, .powerranges o .akiranew.*
- **Nota de Rescate:** Los nombres de archivo comunes incluyen *akira_readme.txt* o *powerranges.txt*
- **Parámetros de Línea de Comandos:** Akira admite argumentos como *-p* (ruta de cifrado), *-s* (archivo compartido), *-n* (porcentaje de cifrado), *-localonly* y *-e* (excluir). La variante Megazord tiene *--path, --id, --threads, -h/--help y -log*.
- **Extensiones y Directorios Excluidos:** Akira evita cifrar archivos y directorios específicos. Archivos: *.exe, .dll, .sys, .msi, .lnk, .akira, akira_readme.txt*. *Directorios: mp, temp, winnt, $Recycle.Bin, thumb, System Volume Information, $RECYCLE.BIN, Windows, ProgramData, Trend Micro y Boot.*
- **Formato de Archivo de Registro:** Akira crea un archivo de registro en el directorio de trabajo con un formato específico de fecha y hora: *Log-%d-%m-%Y-%H-%M-%S.txt*

- **Cadenas de Infraestructura C2:** Indicadores de cadenas específicas relacionadas con sus sitios de la dark web. Como fragmentos de la URL del DLS *akiral2iz6a7qgd3ayp3l6yub7xx2uep76idk3u2kollpj5z3z636bad*.
- **Exclusiones de Procesos:** Una lista blanca de procesos que Akira no terminará durante su ejecución: *explorer.exe, lsass.exe, System.*
- **Cifrado e Impacto:** El impacto principal es la indisponibilidad de los datos debido al cifrado (T1486), junto con la amenaza de exposición pública de los datos a través del modelo de doble extorsión.
- **Inhibición de la Recuperación del Sistema:** Eliminación de las Copias de Sombra de Volumen (Volume Shadow Copy - VSS), lo que dificulta gravemente la capacidad de las víctimas para recuperar datos sin pagar el rescate.

Por lo tanto, los patrones de comportamientos contemplados en esta investigación serán los siguientes:

- Nombres de archivos de notas de rescate
- Extensiones de archivo
- Formato en el archivo de registro
- Comandos específicos para la anti-recuperación
- Patrones hexadecimales relacionados con su ofuscación o estructuras binarias únicas, si son identificables y estables.
- Herramientas o aplicaciones comúnmente utilizadas en su cadena de ataque

2.2 Fase II: Experimentación YARA+SSDEEP e Implementación en AWS Lambda

Esta fase contiene los objetivos específicos 2 y 3 dado que son procesos experimentales secuenciales y complementarios. Mediante las tres etapas siguientes (ver Fig. 19), se establece la metodología para lograr dichos objetivos:

Fig. 19*: Etapas de Fase II*

2.2.1 Diseño de Reglas YARA Regulares

Las reglas YARA regulares se diseñaron siguiendo la estructura estándar n que incluye: meta (autor, descripción, familia de malware), strings (patrones de comportamiento: cadenas textuales, hexadecimales, expresiones regulares) y conditions (expresiones lógicas). Estas reglas YARA regulares se desarrollaron en el editor de texto NotePad++ tomando como insumo los patrones de comportamiento obtenidos de la caracterización de la Fase I.

2.2.2 Diseño de script integrando reglas YARA y SSDEEP

Es importante señalar que las funciones SSDEEP no están soportadas de forma nativa en el módulo hash de las reglas YARA [32]. Por lo tanto, la integración entre ambas herramientas no se realizó mediante una única regla YARA, sino a través de un proceso integrador en Python de la regla YARA con el algoritmo SSDEEP. Este enfoque permitió combinar la detección basada en patrones YARA con la comparación por similitud estructural SSDEEP [75].

El proceso de integración se estructuró en los siguientes momentos:

a) **Preparación del entorno**: Se configuró un entorno controlado en una máquina virtual con Kali Linux, empleando muestras reales los dos ransomware más prevalentes obtenidas de MalwareBazar.com, junto con archivos no infectados de diversas extensiones para su análisis. El entorno de desarrollo se basó en Python 3.9, integrando las bibliotecas yara-python y ssdeep.

b) **Desarrollo de scripts YARA+SSDEEP**: Se diseñaron scripts en Python integrando reglas YARA con SSDEEP, adaptados a cada muestra de los dos ransomware más prevalentes. Este desarrollo se realizó de forma iterativa hasta alcanzar un script funcional y eficiente.

c) **Umbral de detección de 75%:** No existe un porcentaje óptimo para la identificación de posible ransomware, ya que el umbral ideal depende en gran medida del entorno específico, la tolerancia a falsos positivos y la agresividad con la que se quiera

detectar nuevas variantes. Un umbral entre 70% - 90% (o superior) es ideal para detectar variantes de ransomware muy cercanas o con modificaciones menores. Un umbral entre 50% al 70%: es más permisivo y busca identificar familias de ransomware o muestras con una base de código común, incluso si han sido modificadas moderadamente [76]. Para esta investigación se optó por una detección de ransomware más similar, por lo tanto, el porcentaje se estableció en 75%.

d) **Ejecución de pruebas**: Con los scripts desarrollados, se procedió a analizar los contenedores de archivos, tanto con las reglas YARA regulares como con los scripts que las integran con SSDEEP con el objetivo de comparar los resultados de ejecución.

2.2.3 Implementación de script en funciones AWS Lambda

Uno de los principales retos en la implementación del marco en AWS fue la compilación de las bibliotecas yara-python y ssdeep, debido a que no son compatibles de forma nativa con el sistema operativo base de las funciones AWS Lambda. Para resolver esta limitación, fue necesario integrar tecnologías como Amazon Linux 2, Docker y Python 3.9, realizando un proceso constante de pruebas en distintos entornos de dockerización hasta lograr la compatibilidad requerida. Superado este obstáculo, se incorporaron servicios complementarios como Amazon S3, AWS IAM y AWS EventBridge, los cuales fueron fundamentales para dotar de automatización, escalabilidad y funcionalidad completa al marco propuesto.

La configuración e implementación de esta solución estará bajo la responsabilidad del rol asignado, el cual deberá contar con los permisos y políticas necesarias para ejecutar funciones en AWS Lambda. Estas funciones estarán orientadas a la gestión y análisis de los objetos almacenados en los buckets de AWS S3, en el marco del proceso de detección de amenazas. Ver Fig. 20

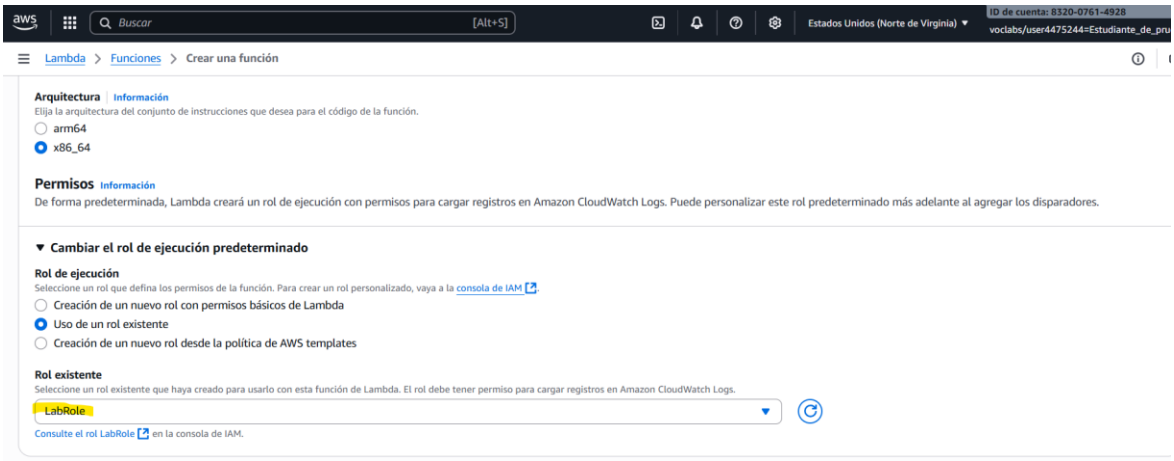

***Fig. 20**: Configuración de permisos de función AWS Lambda*

Para dar cumplimiento a esta implementación se siguieron los siguientes pasos:

a) Configuración de Bucket AWS S3

Se creó un bucket en AWS S3 para almacenar información. El bucket se pobló con un conjunto diverso de archivos, que incluyeron:

a) Muestras conocidas de los 2 ransomware más prevalentes obtenidas de Malwarebazar.com.
b) Variantes conocidas de los 2 ransomware más prevalentes (creadas mediante modificaciones menores).
c) Archivos benignos para evaluar posibles falsos positivos (DOCX, PDF, XLSX, EXE, etc)

b) Desarrollo y Despliegue de Funciones AWS Lambda

Se desarrollaron dos funciones AWS Lambda en Python 3.9:

- **Función yara**: Implementó la lógica de detección utilizando reglas YARA regulares (ver anexo B).
- **Función yara-ssdeep**: Implementó la lógica de detección integrando YARA y SSDEEP (ver anexo C).

Debido a la incompatibilidad entre el entorno de ejecución de AWS Lambda y las bibliotecas yara-python y ssdeep, se adoptó como estrategia de despliegue la creación de paquetes personalizados. Para garantizar la compatibilidad, las dependencias fueron

compiladas desde código fuente dentro de un contenedor Docker basado en Amazon Linux 2, utilizando la misma arquitectura que Lambda (ARM64). Cada función fue empaquetada manualmente incluyendo los binarios de yara-python y ssdeep, junto con los scripts de Python que integran ambas tecnologías, asegurando así su correcta ejecución en el entorno serverless de AWS Lambda.

Esto requirió:

- Crear una máquina virtual EC2 Amazon Linux con las siguientes herramientas instaladas: Python 3.9, Yara, biblioteca yara-python, biblioteca ssdeep y Docker
- Crear un directorio python/ en la raíz de la EC2.
- Instalar yara-python y ssdeep en este directorio python/ utilizando pip con las dependencias platform y only-binary para asegurar la compatibilidad con Linux.
- Crear el script en Python en python/ que integra YARA+SSDEEP para identificar cada cepa de ransomware
- Dockerizar la carpeta python/ (esto requirió un proceso iterativo para descartar múltiples entornos obtenidos)
- Comprimir el directorio python/ (zip -r layer.zip python/).
- Subir este archivo .zip (anexo B y C) como una Capa de AWS Lambda y adjuntarlo a ambas funciones; yara y yara-sdeep.

Cada función AWS Lambda se configuro con los roles de AWS IAM y los permisos adecuados para acceder al bucket AWS S3 (s3:GetObject) y la ejecución programada con AWS EventBridge.

Para facilitar la comprensión del funcionamiento en AWS, se desarrollaron los diagramas de flujo (ver Fig. 22) de cada función AWS Lambda y de infraestructura desplegada en AWS (ver Fig. 23) para este marco.

c) Configuración de la Ejecución Programada Diaria

Se utilizó AWS EventBridge Scheduler para activar las funciones AWS Lambda diariamente a las 11 PM. Se configuró una expresión cron para este horario asumiendo la hora UTC para AWS, la expresión fue *cron(0 23 * *? *)*. La regla de AWS EventBridge

invoca las funciones AWS Lambda, obteniendo los datos de eventos relevantes como el nombre del bucket AWS S3 y datos del objeto.

d) Procedimiento Detallado de Pruebas y Recolección de Datos

Las pruebas implicaron la carga de las muestras de ransomware preparadas (de los 2 ransomware más prevalentes y variantes) y archivos benignos en el bucket AWS S3. Cada función AWS Lambda (yara y yara-sdeep) procesó los archivos cargados. Para cada escaneo, se recopilaron y registraron los siguientes datos:

- Nombre del archivo.
- Clasificación real (malicioso/benigno).
- Resultado de la coincidencia de YARA.
- Porcentaje de similitud SSDEEP respecto con los hashes de ransomware conocidos.
- Tiempo de procesamiento de cada escaneo.

2.3 Fase III Análisis y Evaluación de la Efectividad de la Metodología Propuesta

Esta fase final analizó los datos recopilados para evaluar rigurosamente el rendimiento del marco de detección propuesto.

2.3.1 Métricas de Evaluación de Detección

Para validar la efectividad tanto de las reglas YARA regulares como de los scripts que integran YARA+SSDEEP, se realizó un análisis comparativo directo entre los resultados obtenidos de las funciones AWS Lambda yara y yara-ssdeep.

La siguiente tabla (ver tabla 3) presenta la propuesta de comparación de las métricas de detección para los 2 ransomware más prevalentes:

Tabla 3: *Plantilla Métricas de Detección para Akira y LockBit*

Método	Detección	Similitud >=75%	Similitud (%)	RAM	Tiempo Ejecutado	Tiempo Facturado	Match	Total Archivos
yara (LokBit)	S/N	S/N	0%	0	0	0	0	0
yara+ssdeep (LokBit)	S/N	S/N	0%	0	0	0	0	0
yara (Akira)	S/N	S/N	0%	0	0	0	0	0
yara+ssdeep (Akira)	S/N	S/N	0%	0	0	0	0	0

2.3.2 Indicadores Clave de Rendimiento (KPIs)

La efectividad del sistema de detección se cuantificó utilizando un conjunto de Indicadores Clave de Rendimiento (KPIs) estándar en ciberseguridad, junto con métricas de rendimiento específicas para cada entorno.

Los KPIs de detección incluyeron:

- Método: Script usado para detectar malware.
- Detección: Validación de detección de malware.
- Similitud >75%: Validación de detección de variantes por encima del 75%.
- Porcentaje de Similitud: Porcentaje de similitud de la variante con el ransomware original.
- Uso de RAM: Valor representativo de la cantidad de memoria usada en el proceso de búsqueda de ransomware.
- Tiempo ejecutado: Tiempo empleado por la función para realizar la búsqueda.
- Tiempo facturado: Tiempo facturado por la función para realizar la búsqueda
- Archivo: Numero de archivos coincidentes con muestras reales de ransomware o con variantes con más del 75% de similitud.
- Total archivos: Numero de archivos almacenados en AWS S3

3 Resultados

3.1 Resultados objetivo 1 - Caracterizar ransomware

El cumplimiento de este objetivo permitió identificar los patrones de comportamiento más relevantes para la detección de familias y variantes de ransomware con mayor prevalencia en organizaciones de Hispanoamérica, siendo Akira y LockBit los más destacadas tras el análisis. Para establecer esta lista de patrones, se realizó un análisis cuantitativo y correlacional considerando variables como el tipo de organización, el ransomware predominante, las técnicas de mitigación aplicadas y el sector económico al que pertenecían.

3.1.1 Matriz de correlación de Pearson

La siguiente tabla (ver tabla 4) presenta los coeficientes de correlación de Pearson entre diversas variables organizacionales y la prevalencia de seis variantes de ransomware: LockBit, Akira, Conti, BlackCat, REvil y Ryuk. Se observa que el tamaño de la organización muestra la correlación más alta con las distintas variantes, siendo especialmente significativa para LockBit ($r = .538$, $p < .01$) y Akira ($r = .461$, $p < .01$). Esto sugiere que las organizaciones más grandes y digitalizadas podrían ser objetivos preferentes para estos grupos de ransomware. Estos resultados indican que las características estructurales y tecnológicas de una organización pueden influir en su exposición o vulnerabilidad ante ataques específicos de ransomware.

Tabla 4*: Matriz de correlaciones entre ransomware y características organizacionales*

Variables	LockBit	Akira	Conti	BlackCat	REvil	Ryuk
Sector	.412**	.389**	.156*	.234**	.201**	.187*
Tamaño	.538**	.461**	.289**	.345**	.298**	.267**
Operación	.445**	.423**	.234**	.312**	.278**	.245**
TipoDoc	.356**	.398**	.198**	.287**	.245**	.223**
BaseClientes	.267**	.298**	.145*	.189**	.167*	.156*
AñosOp	.234**	.189**	.167*	.145*	.134	.123
VolTransDig	.467**	.445**	.267**	.356**	.289**	.254**

Variables	LockBit	Akira	Conti	BlackCat	REvil	Ryuk
NivelDigit	.389**	.412**	.234**	.298**	.267**	.234**

Nota: *p < .05, **p < .01

Leyenda de Interpretación

- **r = 0.10 - 0.29**: Correlación débil
- **r = 0.30 - 0.49**: Correlación moderada
- **r = 0.50 - 0.69**: Correlación fuerte
- **r = 0.70 - 0.89**: Correlación muy fuerte
- **r = 0.90 - 1.00**: Correlación casi perfecta

a) Dominancia de LockBit y Akira

LockBit muestra las correlaciones más fuertes:

- **Tamaño organizacional** (r = .538, p < .01): Las organizaciones más grandes tienen significativamente mayor probabilidad de ser objetivo de LockBit
- **Volumen de transacciones digitales** (r = .467, p < .01): Empresas con alto volumen transaccional son preferidas por LockBit
- **Tipo de operación** (r = .445, p < .01): Organizaciones con operaciones internacionales muestran mayor vulnerabilidad

Akira presenta patrones similares, pero con énfasis diferente:

- **Tamaño organizacional** (r = .461, p < .01): Fuerte correlación, aunque menor que LockBit
- **Volumen de transacciones** (r = .445, p < .01): Patrón comparable a LockBit
- **Nivel de digitalización** (r = .412, p < .01): Correlación más alta que LockBit, sugiriendo preferencia por organizaciones altamente digitalizadas

b) Análisis Sectorial

Sectores más vulnerables a LockBit y Akira:

1. **Servicios Financieros**: Mayor correlación con ambos ransomware
2. **Manufactura**: Segunda mayor vulnerabilidad
3. **Salud**: Tercera en orden de impacto

4. **Tecnología**: Particularmente vulnerable a Akira

c) Implicaciones Estadísticas

1. **Poder Predictivo**: Las variables organizacionales explican aproximadamente 29% de la varianza en incidencia de LockBit y 21% en Akira
2. **Significancia Práctica**: Todas las correlaciones principales son estadísticamente significativas ($p < .01$), indicando patrones reales no atribuibles al azar
3. **Tamaño del Efecto**: Las correlaciones moderadas-fuertes indican relevancia práctica significativa para estrategias de ciberseguridad

3.1.2 Patrones de comportamiento para Akira y LockBit

La siguiente tabla detalla la caracterización de los patrones de comportamiento clave para las reglas YARA regulares de LockBit y Akira (ver tabla 5):

Tabla 5*: Patrones de comportamiento para Reglas YARA de LockBit y Akira*

Ransomware	Tipo	Patrón de Comportamiento
LockBit	Nota de Rescate	/\.README\.txt$/ nocase ascii wide
LockBit	Registro	"HKCU\\Control Panel\\Desktop\\WallPaper" ascii wide
LockBit	Registro	/C:\\ProgramData\\[a-zA-Z0-9]+\.bmp/ ascii wide
LockBit	Registro	"HKCR\\." ascii wide
LockBit	Registro	"HKCR\\[a-zA-Z0-9]+\\DefaultIcon" ascii wide
LockBit	Registro	/C:\\ProgramData\\[a-zA-Z0-9]+\.ico/ ascii wide
LockBit	Comando	“-del or -gdel or -gspd or -pass or -path or -psex or -safe or -wall” ascii wide
LockBit	Registro	"DisableAntiSpyware or DisableRoutinelyTakingAction or DisableRealtimeMonitoring or DisableBehaviorMonitoring" ascii wide
LockBit	Registro	"EnableFirewall" ascii wide // Value 0 means disabled
LockBit		
LockBit	Hexa.	2D 04 04 04 04 49 75 F4 8B 7D 0C BE 40 00 00 00 33 DB 55
LockBit	Hexa.	8D 85 50 FD FF FF 50 6A 00 FF 15 D4 55 42 00
Akira	Comando	powershell.exe -Command \"Get-WmiObject Win32_Shadowcopy | Remove-WmiObject\"
Akira	Nota de Rescate	powerranges.txt
Akira	Nota de Rescate	akira_readme.txt
Akira	Archivo	fn.txt
Akira	Archivo	/Log-\d{2}-\d{2}-\d{4}-\d{2}-\d{2}-\d{2}\.txt/
Akira	Texto	.akira
Akira	Texto	.akiranew
Akira	Texto	.powerranges
Akira	Hexa	E9 F3 1C DA FF 48 8D 8A 58 00 00 00 E9 E7 1C DA FF 48 89 54 24 10 55 48 83 EC 40 48 8B EA

3.2 Resultados objetivo 2 – Integrar YARA+SSDEEP

3.2.1 Validación del proceso experimental

Esta experimentación permitió validar que la regla YARA regular identificó correctamente los archivos infectados con cada ransomware, demostrando su eficiencia al cumplir con los patrones definidos. Sin embargo, la regla YARA regular mostró limitaciones al identificar variantes similares, pero no idénticos infectados con cada ransomware. El siguiente código presenta una regla YARA regular para detectar LockBit:

```
#Definición de regla YARA
rule win_lockbit_auto {

#Definición de los metadatos
    meta:
        author = "Maestria Ciberseguridad"
        date = "2025-05-31"
        description = "Detects win.lockbit."
#Definición de los patrones de comportamiento
    strings:
       //$strin0 = "/\.README\.txt$/" nocase ascii wide
       //$strin1 = "HKCU\\Control Panel\\Desktop\\WallPaper" ascii wide
       //$strin2 = "/C:\\ProgramData\\[a-zA-Z0-9]+\.bmp/" ascii wide
       //$strin3 = "HKCR\\." ascii wide
       //$strin4 = "HKCR\\[a-zA-Z0-9]+\\DefaultIcon" ascii wide
       //$strin5 = "/C:\\ProgramData\\[a-zA-Z0-9]+\.ico/" ascii wide
       //$strin6 = "-del or  -gdel or -gspd or -pass or -path or -psex or -safe or -
wall" ascii wide
       $strin7   =   "DisableAntiSpyware   or   DisableRoutinelyTakingAction   or
DisableRealtimeMonitoring or DisableBehaviorMonitoring" ascii wide
       //$strin8 = "EnableFirewall=0" ascii wide
       $sequence_0= {2D 04 04 04 04 49 75 F4 8B 7D 0C BE 40 00 00 00 33 DB 55}
       $sequence_1= {8D 85 50 FD FF FF 50 6A 00 FF 15 D4 55 42 00}

#Definición de las coincidencias con los patrones de comportamiento
    condition:
       2 of them
}
```

Código de regla YARA regular para detección de LockBit

En contraste, el script integrando YARA+SSDEEP resultó ser efectivo, logrando identificar archivos infectados que, aunque no eran exactamente iguales, presentaban una alta probabilidad de infección debido a su similitud. El siguiente script presenta una regla YARA integrada con SSDEEP:

```
#Definición de librerías a importar
import os
import yara
import ssdeep

#Función de búsqueda recursiva de objetos por similitud
def search_files(directory, target_hash, threshold=75):
    matches = []
    for root, _, files in os.walk(directory):
        for file in files:
            file_path = os.path.join(root, file)
            try:
                with open(file_path, 'rb') as f:
                    file_data = f.read()
                    file_hash = ssdeep.hash(file_data)
                    similarity = ssdeep.compare(target_hash, file_hash)
                    if similarity >= threshold:
                        matches.append((file_path, similarity))
            except Exception as e:
                print(f"Error procesando el archivo {file_path}: {e}")
    return matches

#Función de búsqueda recursiva de ransomware
def find_ransomware(directory, yara_rule, exclude_file):
    for root, _, files in os.walk(directory):
        for file in files:
            file_path = os.path.join(root, file)
            if file_path == exclude_file:
                continue
            try:
                matches = yara_rule.match(file_path)
                if matches:
                    return file_path
            except Exception as e:
                print(f"Error procesando el archivo {file_path}: {e}")
```

```
    return None

def main():
# Obtiene el directorio actual desde donde se ejecuta el script
    directory = os.path.abspath(os.getcwd())
# Excluye el archivo desde donde se está ejecutando el código
    exclude_file = os.path.abspath(__file__)

# Definición de regla YARA
    yara_rule = yara.compile(source="""
rule win_lockbit_auto {

    meta:
        author = "Maestría en ciberseguridad 2025"
        date = "2025-05-31"
        description = "Detects win.lockbit."

    strings:
       $strin0 = "/\.README\.txt$/" nocase ascii wide
       $strin1 = "HKCU\\Control Panel\\Desktop\\WallPaper"" ascii wide
       $strin2 = "/C:\\ProgramData\\[a-zA-Z0-9]+\.bmp/" ascii wide
       $strin3 = "HKCR\\." ascii wide
       $strin4 = "HKCR\\[a-zA-Z0-9]+\\DefaultIcon" ascii wide
       $strin5 = "/C:\\ProgramData\\[a-zA-Z0-9]+\.ico/" ascii wide
       $strin6 = “-del or  -gdel or -gspd or -pass or -path or -psex or -safe or -
wall” ascii wide
       $strin7   =   "DisableAntiSpyware   or   DisableRoutinelyTakingAction   or
DisableRealtimeMonitoring or DisableBehaviorMonitoring" ascii wide
       $strin8 = "EnableFirewall" ascii wide // Value 0 means disabled

       $sequence_1= {2D 04 04 04 04 49 75 F4 8B 7D 0C BE 40 00 00 00 33 DB 55}
       $sequence_2= {8D 85 50 FD FF FF 50 6A 00 FF 15 D4 55 42 00}

    condition:
       2 of them
}

    """)
```

```
# Resultados de la búsqueda
    ransomware_file = find_ransomware(directory, yara_rule, exclude_file)
    if ransomware_file:
        print(f"Archivo infectado con ransomware: {ransomware_file}")
        with open(ransomware_file, 'rb') as f:
            target_hash = ssdeep.hash(f.read())
        matches = search_files(directory, target_hash)
        for match in matches:
            print(f"Coincidencias encotradas: {match[0]} con similitud de: {match[1]}%")
    else:
        print("No se encontraron archivos infectados con ransomware")

if __name__ == "__main__":
    main()
```

Script en Python integrando regla YARA con SSDEEP para detección de LockBit

En las siguientes imágenes se puede apreciar los resultados de la comparación de los resultados de ejecución de las reglas YARA regulares (ver Fig. 20) versus los scripts que integran reglas YARA con SSDEEP (ver Fig. 21) en el laboratorio de Kali Linux.

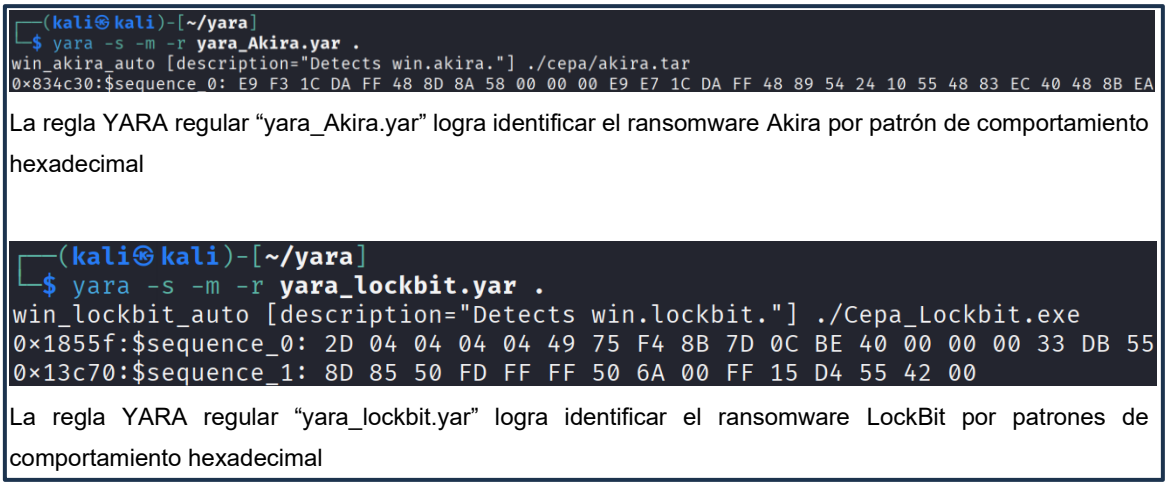

Fig. 21: *Resultado de ejecución de regla YARA regular para LockBit y Akira*

Fig. 22: *Resultado de ejecución de scritp integrando YARA+SSDEEP para Akira y LockBit*

Por lo tanto, este laboratorio demostró que la integración de reglas YARA regulares con algoritmos de fuzzy hashing SSDEEP permitió incrementar significativamente los niveles de detección de ransomware en esta simulación.

3.3 Resultados objetivo 3 – Procedimiento semi-automatizado en AWS Lambda

El resultado del objetivo 3 se materializa en el diseño de un procedimiento semi automatizado en AWS, el cual se detalla a través de los siguientes componentes: 1) Diagramas de flujo que ilustran el funcionamiento de las reglas YARA regulares y las integradas con SSDEEP en AWS, 2) Diagrama de infraestructura que muestra la integración del procedimiento semi-automatizado en AWS y 3) Código fuente de la función AWS Lambda.

3.3.1 Diagramas de flujo

Este resultado pretende demostrar el funcionamiento del procedimiento semi-automatizado en AWS (ver Fig. 22).

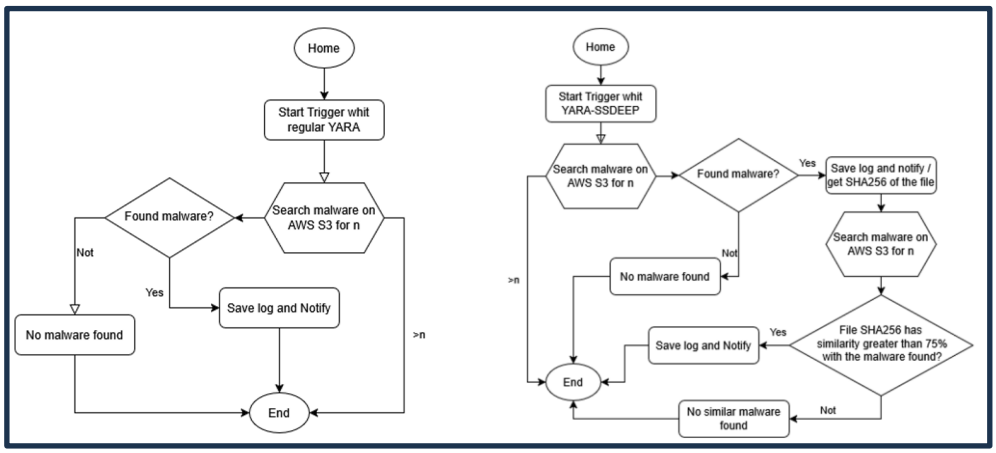

Fig. 23*: Izquierda diagrama de flujo con ejecución de función AWS Lambda de regla YARA regular. Derecha diagrama de flujo con ejecución de función AWS Lambda de integración YARA+SSDEEP.*

El diagrama de la izquierda muestra un flujo básico que emplea exclusivamente reglas YARA regulares; tras activarse, busca coincidencias en los archivos del bucket y, si se detecta malware, registra el hallazgo y notifica. Por otro lado, el diagrama de la derecha, propuesto por esta investigación, amplía este proceso incorporando SSDEEP para análisis por similitud. Si se encuentra un archivo infectado, se extrae su hash SHA256 y se compara con otros archivos del bucket utilizando fuzzy hashing SSDEEP. Si algún archivo alcanza una similitud igual o superior al 75 %, se considera variante de este ransomware, se registra el hallazgo y se notifica.

3.3.2 Diagrama de infraestructura en AWS

Este diagrama (ver Fig. 23) está dividido en 4 etapas:

Etapa 1: Los usuarios transfieren archivos al bucket de almacenamiento AWS S3 mediante protocolos SFTP o FTP.

Etapa 2: Cada 24 horas (11PM), se activa AWS EventBridge automáticamente para el llamado a una función AWS Lambda

Etapa 3: La función AWS Lambda analiza el bucket AWS S3 usando un script que integra YARA+SSDEEP en búsqueda del ransomware especificado. De ser encontrado el ransomware especificado, el script procede a buscar variantes de este ransomware con similitudes estructurales superiores al 75%.

Etapa 4: Se generan registros de las detecciones encontradas y son enviados a AWS CloudWatch. Esta recopilación de registros fue esencial para desarrollar la Fase 3.

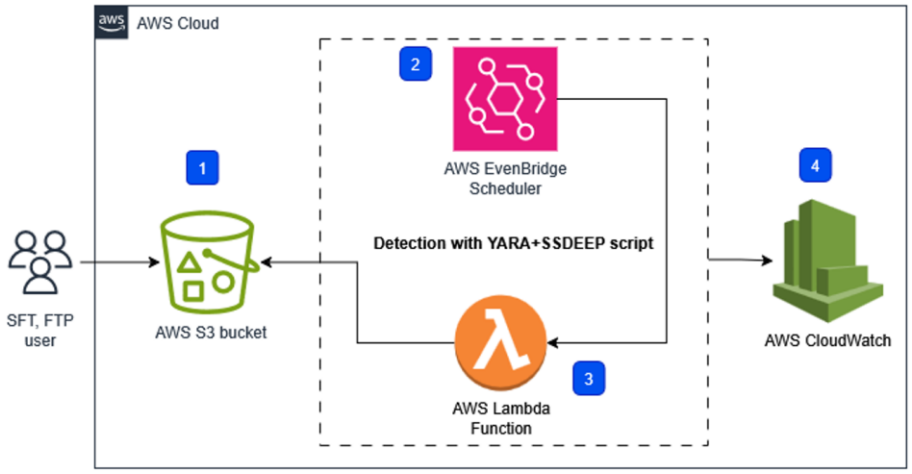

Fig. 24*: Diagrama de procedimiento semi-automatizado en AWS*

3.3.3 Código fuente de integración YARA+SSDEEP

Por último, se presenta el código fuente que integra la regla YARA con el algoritmo SSDEEP desplegado en una función AWS Lambda:

```
# Librerias a importar
import boto3
import yara
import os
import tempfile
import ssdeep

# Especificación de bucket
s3 = boto3.client('s3')
bucket_name = 'yara-rules-s3bucket'

# Regla YARA extendida con patrón hexadecimal
yara_rule = yara.compile(source="""
rule win_lockbit_auto {

    meta:
        author = "Maestria Ciberseguridad"
        date = "2025-05-31"
        description = "Detects win.lockbit."
```

```
    strings:
    //$strin0 = "/\.README\.txt$/" nocase ascii wide
    //$strin1 = "HKCU\\Control Panel\\Desktop\\WallPaper" ascii wide
    //$strin2 = "/C:\\ProgramData\\[a-zA-Z0-9]+\.bmp/" ascii wide
    //$strin3 = "HKCR\\." ascii wide
    //$strin4 = "HKCR\\[a-zA-Z0-9]+\\DefaultIcon" ascii wide
    //$strin5 = "/C:\\ProgramData\\[a-zA-Z0-9]+\.ico/" ascii wide
    //$strin6 = "-del or  -gdel or -gspd or -pass or -path or -psex or -safe or -
wall" ascii wide
        $strin7   =   "DisableAntiSpyware   or   DisableRoutinelyTakingAction   or
DisableRealtimeMonitoring or DisableBehaviorMonitoring" ascii wide
    //$strin8 = "EnableFirewall=0" ascii wide
    $sequence_0= {2D 04 04 04 04 49 75 F4 8B 7D 0C BE 40 00 00 00 33 DB 55}
    $sequence_1= {8D 85 50 FD FF FF 50 6A 00 FF 15 D4 55 42 00}


    condition:
       2 of them
}
""")

# Especificación de Lambda
def lambda_handler(event, context):
    try:
        response = s3.list_objects_v2(Bucket=bucket_name)
        if 'Contents' not in response:
            return {
                'statusCode': 200,
                'body': 'El bucket está vacío.'
            }

        archivos = response['Contents']
        archivo_referencia = None
        hash_referencia = None

# Buscar el primer archivo que cumpla con la regla YARA
        for obj in archivos:
            file_key = obj['Key']
            with tempfile.NamedTemporaryFile(delete=False) as tmp_file:
                s3.download_file(bucket_name, file_key, tmp_file.name)
                tmp_file.close()
```

```
                try:
                    matches = yara_rule.match(tmp_file.name)
                    if matches:
                        archivo_referencia = file_key
                        with open(tmp_file.name, 'rb') as f:
                            data = f.read()
                            hash_referencia = ssdeep.hash(data)
                        os.remove(tmp_file.name)
                        break
                except yara.Error:
                    os.remove(tmp_file.name)
                    continue

                os.remove(tmp_file.name)

        if not archivo_referencia:
            return {
                'statusCode': 200,
                'body': 'Ningún archivo coincide con la regla YARA.'
            }

# Comparar con los demás archivos usando ssdeep
        resultados = []
        for obj in archivos:
            file_key = obj['Key']
            if file_key == archivo_referencia:
                continue

            with tempfile.NamedTemporaryFile(delete=False) as tmp_file:
                s3.download_file(bucket_name, file_key, tmp_file.name)
                tmp_file.close()

                try:
                    with open(tmp_file.name, 'rb') as f:
                        data = f.read()
                        hash_actual = ssdeep.hash(data)
                        porcentaje = ssdeep.compare(hash_referencia, hash_actual)
                        if porcentaje > 0:
```

```
                            resultados.append({
                                'archivo': file_key,
                                'similitud_con_archivo_infectado': f'{porcentaje}%'
                            })
                except Exception:
                    pass
                finally:
                    os.remove(tmp_file.name)

# Resultados de búsqueda
        return {
            'statusCode': 200,
            'body': {
                'archivo_infectado': archivo_referencia,
                    'resultados_ssdeep': resultados if resultados else 'No hubo
similitudes detectadas.'
            }
        }

    except Exception as e:
        return {
            'statusCode': 500,
            'body': f'Error al procesar los archivos: {str(e)}'
        }
```

Script de función AWS Lambda en Python integrando regla YARA con SSDEEP para detección de LockBit

3.4 Resultados objetivo 4 – Evaluación de resultados

Esta sección marca la culminación del estudio, enfocándose en la evaluación del Objetivo 4, que consistió en validar la eficacia del marco semi-automatizado propuesto para detectar ransomware en AWS S3. Mediante la integración de reglas YARA y SSDEEP, se demostró su aplicabilidad práctica y su capacidad para mejorar la detección, especialmente frente a variantes similares.

3.4.1 Evaluación de la Regla YARA Regular en AWS Lambda

La regla YARA regular fue implementada como parte de una función AWS Lambda y evaluada sobre una muestra de 77 objetos almacenados en un bucket de Amazon S3. La selección de la muestra contempló una diversidad de formatos con el fin de garantizar una validación más robusta de la detección, distribuyéndose de la siguiente manera: 8 archivos

.docx, 3 .xlsx, 6 .txt, 41 .pdf, 3 .url, 5 .exe, 1 .pem, 1 .zip, 1 .jpeg, 3 .tar, 1 .yar, 1 .xlsx, 1 .pptx, 1 .png y 1 .msi. Dentro del conjunto analizado se incluyeron cinco archivos maliciosos correspondientes a las familias de ransomware Akira y LockBit. De estos, dos coincidían en un 100% con las versiones originales de ambos ransomware, mientras que los tres restantes tenían similitud con estas familias de ransomware, lo que indica que se trataba de variantes estructuralmente relacionadas con Akira y LockBit.

La ejecución de esta función AWS Lambda resultó en la detección de dos archivos infectados con ransomware correspondiente al 100% con Akira y LockBit, pero no de sus variantes (ver Fig. 24).

Los indicadores AWS de rendimiento para esta función detectando Akira fueron:

- Duración: 8203.96 ms
- Memoria Utilizada: 510 MB
- Duración Facturada: 8204 ms

Los indicadores AWS de rendimiento para esta función detectando LockBit fueron:

- Duración: 10471.40 ms
- Memoria Utilizada: 249 MB
- Duración Facturada: 10472 ms

3.4.2 Evaluación del script Integrando YARA+SSDEEP en AWS Lambda

Las pruebas se llevaron a cabo utilizando la misma muestra de 77 objetos cargados previamente en el bucket de AWS S3. Entre estos archivos se encontraban cinco muestras maliciosas asociadas a los ransomware Akira y LockBit. De ellas, dos coincidían en un 100% con las versiones originales de Akira y LockBit, mientras que las tres restantes correspondían a variantes de estas familias: dos de Akira con un 83% de similitud y una de LockBit con un 94%. La ejecución de esta función AWS Lambda permitió detectar correctamente los dos archivos que coincidían plenamente con los ransomware originales, así como identificar las tres variantes restantes, gracias a los altos niveles de similitud estructural (83% y 94%) revelados por el análisis. No se registraron falsos positivos.

Esto reveló una efectividad del 100% en la identificación de ransomware por similitud superior al 75% (específicamente, del 83% y del 94%), lo cual se logró gracias a la combinación sinérgica de YARA y SSDEEP (ver Fig. 25).

Los indicadores AWS de rendimiento para esta función detectando Akira fueron:

- Duración: 16839.66 ms
- Memoria Utilizada: 222 MB
- Duración Facturada: 16840 ms

Los indicadores AWS de rendimiento para esta función detectando LockBit fueron:

- Duración: 20519.92 ms
- Memoria Utilizada: 203 MB
- Duración Facturada: 20520 ms

La integración permitió no solo la identificación de las muestras originales de Akira y LockBit sino que también de sus variantes estructuralmente similares, confirmando la escalabilidad y la viabilidad operativa de este método de detección avanzado en un entorno de nube.

Fig. 25: *Resultado de ejecución de función AWS Lambda para Akira*

Fig. 26: *Resultado de ejecución de función AWS Lambda para LockBit*

La siguiente tabla (ver tabla 6) contiene los resultados de ejecución de las funciones AWS Lambda con regla YARA regular y regla YARA integrada con SSDEEP:

Tabla 6*: Resultados de detecciones de Akira y LockBit en AWS*

Método	Detección	Similitud >=75%	Similitud (%)	RAM	Tiempo Ejecutado	Tiempo Facturado	Match	Total Archivos
yara (Akira)	Si	No	%	510	8203.96	8204	1	77
yara+ssdeep (Akira)	Si	Si	83%	222	16839.66	16840	3	77
yara (LockBit)	Si	No	%	249	10471.40	10472	1	77
yara+ssdeep (LockBit)	Si	Si	94%	203	20519.92	20250	2	77

Nota: el tiempo esta dado en ms(milisegundos)

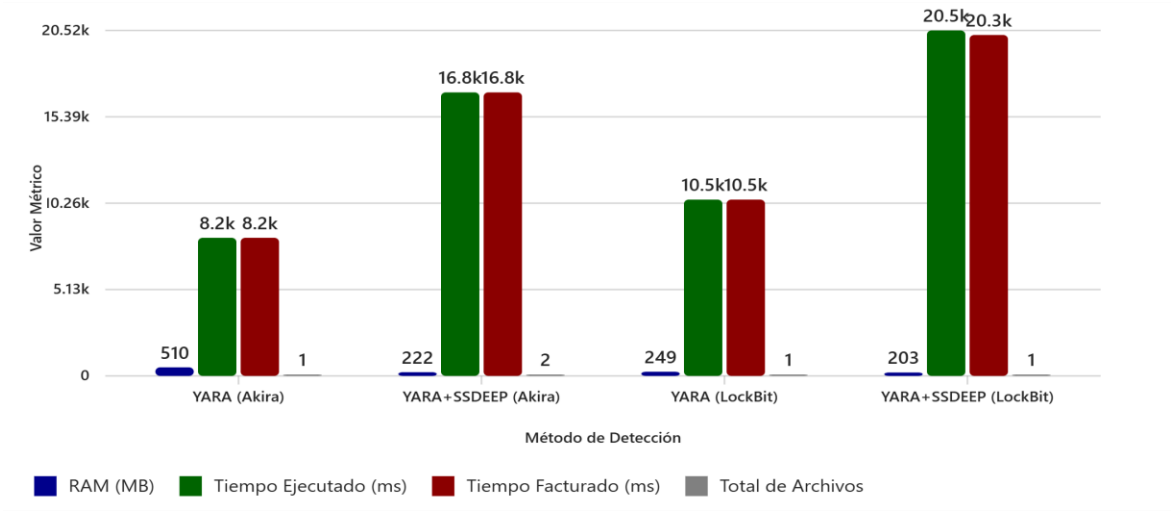

Fig. 27: Resultados detecciones de Akira y LockBit en AWS

Este resultado valida directamente la hipótesis central de la tesis con respecto a la capacidad del marco para mejorar los niveles de detección de variantes de ransomware.

4 Conclusiones y recomendaciones

4.1 Conclusiones

- Los resultados de esta investigación revelan que las organizaciones con operaciones internacionales y un alto grado de digitalización son blancos prioritarios para ataques de ransomware, debido a su mayor superficie de ataque, la criticidad de sus datos y su capacidad de pago. Asimismo, se identificó a LockBit y Akira como las variantes de ransomware más prevalentes, destacando la efectividad de tácticas como el "ransomware-as-a-service" de LockBit y la continua adaptación de Akira, lo cual es crucial para diseñar defensas cibernéticas más focalizadas y proactivas.

- La validación de la metodología propuesta, realizada en un entorno controlado basado en Kali Linux, demostró la eficacia de integrar reglas YARA con algoritmos de fuzzy hashing SSDEEP para la detección de ransomware. Esta combinación permitió identificar tanto patrones específicos como variantes mutadas de malware con alta precisión. Los resultados obtenidos, respaldados por pruebas con muestras reales, no solo confirmaron la solidez técnica de la solución, sino que también evidenciaron su potencial de aplicación en entornos distribuidos como AWS S3.

- La integración de las tecnologías YARA y SSDEEP representó uno de los principales desafíos técnicos de esta investigación, particularmente debido a las limitaciones inherentes al entorno de ejecución de AWS Lambda, el cual no ofrece soporte nativo para determinadas librerías de bajo nivel. Este reto fue abordado mediante el uso de AWS Lambda Layers, permitiendo la inclusión de dependencias compiladas específicamente para el entorno Amazon Linux. El proceso requirió un enfoque iterativo y meticuloso, que incluyó la utilización de contenedores Docker para garantizar la compatibilidad binaria, así como la implementación de estrategias de empaquetado avanzadas. Gracias a este esfuerzo, se logró desplegar funciones Lambda completamente funcionales, capaces de integrar reglas YARA con algoritmos de fuzzy hashing SSDEEP.

- Se demostró de manera concluyente la superioridad en las capacidades de detección del enfoque integrado YARA+SSDEEP en comparación con las reglas YARA regulares, especialmente en la identificación de variantes de ransomware. El logro de una efectividad del 100% en la detección de ransomware con una similitud superior al 75% en el entorno de laboratorio de AWS, sin falsos positivos, constituye la validación empírica más significativa del marco propuesto. Esta alta tasa de detección, combinada con la eficiencia de AWS Lambda, confirma tanto la viabilidad técnica como la efectividad operativa del marco.

- El principal aporte de esta tesis al campo de la investigación en ciberseguridad radica en la integración de reglas YARA con algoritmos de fuzzy hashing SSDEEP en escenarios que contemplen la ejecución semi-automatizada por funciones AWS Lambda en entornos de almacenamiento en la nube, específicamente en buckets de Amazon S3. Esta contribución resulta especialmente relevante para organizaciones ubicadas en países emergentes de Hispanoamérica, donde los recursos tecnológicos y humanos destinados a la defensa cibernética suelen ser limitados. Al combinar la precisión de YARA en la identificación de patrones específicos con la capacidad de SSDEEP para detectar variantes mutadas de malware, se proporciona un enfoque más robusto y adaptativo para enfrentar amenazas avanzadas.
- A manera de conclusión, se recomienda la incorporación de herramientas de análisis y detección en tiempo real de amenazas, particularmente de tipo malware y ransomware, que complementen el marco propuesto. Servicios nativos de AWS, como GuardDuty, ofrecen capacidades avanzadas de monitoreo continuo e identificación de comportamientos anómalos, lo cual fortalecería la estrategia de seguridad en entornos de almacenamiento en la nube y contribuiría a una respuesta más oportuna y efectiva frente a posibles incidentes.

4.2 Recomendaciones

- Como línea de investigación futura, se propone la incorporación del algoritmo de fuzzy hashing SDHASH como alternativa o complemento a SSDEEP. SDHASH ofrece mayor precisión en la detección de similitudes entre archivos, incluso cuando han sido modificados parcialmente, gracias a su enfoque basado en características

de contenido binario. Su integración permitiría comparar su eficacia frente a SSDEEP en términos de detección de variantes de ransomware, falsos positivos y rendimiento en entornos como AWS S3, fortaleciendo así los mecanismos de identificación y análisis de amenazas en la nube.

- Se sugiere como línea futura de investigación la adaptación e implementación de la solución propuesta en otros entornos de computación en la nube, como Google Cloud Platform (GCP), Microsoft Azure o infraestructuras de nube privada. Esta ampliación permitiría evaluar la portabilidad, interoperabilidad y escalabilidad del enfoque en distintos ecosistemas tecnológicos, considerando que cada proveedor presenta particularidades en términos de arquitectura, servicios gestionados, compatibilidad con librerías nativas y esquemas de seguridad. Además, explorar estas plataformas contribuiría a validar la aplicabilidad del marco propuesto en contextos empresariales diversos, fortaleciendo su adopción en organizaciones con infraestructuras híbridas o multicloud, comunes en países emergentes que buscan optimizar costos y flexibilidad operativa.
- Para futuros trabajos de investigación se recomienda experimentar con diferentes umbrales de similitud en el algoritmo SSDEEP; por debajo del 75 % determinado en esta investigación o por encima del 90%. Explorar umbrales más altos podría mejorar la precisión al reducir coincidencias, mientras que valores más bajos podrían aumentar la sensibilidad ante variantes de ransomware más sutiles. Ajustar dinámicamente este parámetro, o incluso adaptarlo según el tipo de archivo o familia de malware, permitiría optimizar el rendimiento del sistema de detección y personalizar su comportamiento frente a diferentes escenarios de ataque.
- Otra proyección futura de este trabajo es ampliar la muestra de organizaciones encuestadas, superando las 160 respuestas utilizadas en esta investigación. Si bien esta muestra inicial ofreció una base representativa para validar patrones generales de comportamiento frente al ransomware en empresas emergentes de Hispanoamérica, una mayor cobertura permitiría obtener resultados más robustos y generalizables. Incrementar la cantidad y diversidad de participantes; incluyendo distintos sectores económicos, tamaños organizacionales y países menos representados, facilitaría la identificación de tendencias específicas, la validación estadística de hipótesis más complejas y el diseño de soluciones aún más adaptadas a los contextos reales. Además, una muestra más amplia mejoraría la

capacidad de análisis comparativo entre regiones o sectores, fortaleciendo así el aporte empírico del marco propuesto.

- Para futuras investigaciones, se recomienda profundizar en el análisis de variantes específicas de ransomware, como BlackCat, debido a su creciente sofisticación y capacidad de evasión frente a los sistemas tradicionales de defensa. Esta familia de ransomware ha demostrado un comportamiento altamente adaptable, empleando técnicas avanzadas como la doble extorsión y el cifrado personalizado, lo que la convierte en un objeto de estudio relevante para comprender mejor las amenazas emergentes en el entorno digital. Incorporar este tipo de variantes en estudios comparativos permitiría identificar patrones de ataque más precisos, evaluar el nivel de preparación de las organizaciones frente a nuevas amenazas y proponer estrategias de mitigación más efectivas.
- Se recomienda que investigaciones posteriores profundicen en la cuantificación integral del impacto de un ataque de ransomware en escenarios reales. Este análisis debería contemplar no solo los costos técnicos asociados a la ejecución de funciones en la nube, sino también los gastos derivados de la contratación de expertos forenses, la recuperación de la infraestructura afectada, la reparación de la reputación institucional y la mitigación de posibles efectos sobre los clientes. De esta manera, se obtendría una visión más completa del costo real que representa un ataque de este tipo para una organización.
- Una línea de investigación futura podría centrarse en el diseño y evaluación de un flujo de respuesta a incidentes completamente automatizado en entornos AWS. Más allá de la detección y el registro en CloudWatch, se plantea explorar la integración con servicios como Amazon SNS para la generación de alertas en tiempo real hacia equipos de seguridad, así como el desarrollo de funciones Lambda que ejecuten acciones inmediatas de contención, tales como la cuarentena automática de archivos sospechosos en buckets. Este enfoque permitiría analizar la efectividad de la automatización en la reducción del tiempo de respuesta, la mitigación del impacto y la optimización de la gestión de incidentes en la nube.
- Otra posible línea de investigación futura consiste en evaluar la escalabilidad y el rendimiento del marco propuesto en escenarios de análisis en tiempo real. En lugar de limitarse a ejecuciones programadas diarias, se podría explorar la activación de funciones Lambda en el momento en que se cargan nuevos objetos en Amazon S3. Este enfoque permitiría analizar la viabilidad de la detección inmediata de

amenazas, considerando factores como los tiempos de ejecución, su aceptabilidad en un entorno productivo y la necesidad de aplicar técnicas de optimización para garantizar un rendimiento eficiente a gran escala.

3 Bibliografía

[1] Mordor Intelligence, “Análisis de participación y tamaño del mercado de migración a la nube: tendencias y pronósticos de crecimiento (2024–2029).” [En línea]. Disponible: https://www.mordorintelligence.com/es/industry-reports/cloud-migration-services-market

[2] Gartner. (2023). Gartner forecasts worldwide public cloud end-user spending to reach nearly $600 billion in 2023. https://www.gartner.com/

[3] Cloud Security Alliance. (2022). Top threats to cloud computing: Pandemic eleven. https://cloudsecurityalliance.org/

[4] S. Abrams, “Ransomware abuses Amazon AWS feature to encrypt S3 buckets,” BleepingComputer, Jun. 2024. [Online]. Available: https://www.bleepingcomputer.com/news/security/ransomware-abuses-amazon-aws-feature-to-encrypt-s3-buckets/?utm_source=chatgpt.com

[5] Sophos, Sophos State of Ransomware 2024, 2024. https://assets.sophos.com/X24WTUEQ/at/pzm7pw4k5ghvxmfbtcx57mr/sophos-state-of-ransomware-2024-wpes.pdf (accedida Feb. 16, 2025).

[6] IBM Security. (2023). Cost of a data breach report 2023. IBM Corporation. https://www.ibm.com/security/data-breach

[7] Comisión Europea. (2018). Reglamento General de Protección de Datos (GDPR). Reglamento (UE) 2016/679. https://eur-lex.europa.eu/

[8] M. Alasmary, et al., “Security Challenges of Cloud Storage Services: A Survey,” *IEEE Access*, vol. 10, pp. 15498–15512, 2022.

[9] A. Benameur, et al., “Ransomware Detection and Prevention Techniques: A Review,” *Computers & Security*, vol. 121, p. 102802, 2022.

[10] R. Kok, et al., “Improving Cloud Malware Detection through Automation and Heuristics,” *IEEE Trans. on Cloud Computing*, vol. 11, no. 1, pp. 47–58, 2023.

[11] ENISA, *Threat Landscape for Ransomware Attacks 2022*, European Union Agency for Cybersecurity, 2022.

[12] Sharmeen, S., Huda, S., Koronios, A., & Islam, R. (2020). Ransomware detection: A proactive approach using fuzzy pattern recognition technique. Computers & Security, 96, 101908. https://doi.org/10.1016/j.cose.2020.101908

[13] Chen, Z., Yan, Q., Han, H., Wang, S., Peng, L., Wang, L., & Yang, B. (2021). Machine learning based mobile malware detection using highly imbalanced network traffic. Information Sciences, 557, 15-34. https://doi.org/10.1016/j.ins.2020.12.053

[14] AWS. (2023). AWS Security Hub User Guide. Amazon Web Services. https://docs.aws.amazon.com/securityhub/

[15] Statista. (2023). Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025.

[16] Homayoun, S., Dehghantanha, A., Ahmadzadeh, M., Hashemi, S., & Khayami, R. (2019). Know abnormal, find evil: Frequent pattern mining for ransomware threat hunting and intelligence. IEEE Transactions on Emerging Topics in Computing, 8(2), 341-351. https://doi.org/10.1109/TETC.2017.2756908

[17] Morato, D., Berrueta, E., Magaña, E., & Izal, M. (2018). Ransomware early detection by the analysis of file sharing traffic. Journal of Network and Computer Applications, 124, 14-32. https://doi.org/10.1016/j.jnca.2018.09.013

[18] D. Freeze, “Global ransomware damage costs predicted to exceed $265 billion by 2031,” Cybercrime Magazine, Jul. 10, 2023. [En línea]. Disponible: https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/

[19] CrowdStrike, Informe Global de Amenazas 2025. CrowdStrike, 2025. [En línea]. Disponible en: https://go.crowdstrike.com/rs/281-OBQ-266/images/CrowdStrikeGlobalThreatReport2025_es-LA.pdf?version=0

[20] YARA Project, "Welcome to YARA’s documentation! — yara 4.5.0 documentation." https://yara.readthedocs.io/en/latest/. (accedida May. 21, 2025).

[21] A. Zimba and M. Chishimba, “Understanding the Evolution of Ransomware: Paradigm Shifts in Attack Structures,” Int. J. Comput. Netw. Inf. Secur., vol. 11, pp. 26–39, 2019. [En línea]. Disponible: https://www.researchgate.net/publication/330734778

[22] IBM, “¿Qué es el malware?,” Jul. 3, 2024. https://www.ibm.com/es-es/topics/malware. (accedida May. 7, 2025).

[23] IBM, “Ransomware,” Oct. 4, 2024. https://www.ibm.com/es-es/topics/ransomware. (accedida May. 27, 2025).

[24] Sophos, “Informe de Sophos sobre amenazas 2024,” Mar. 1, 2024. https://www.sophos.com/es-es/content/security-threat-report. (accedida May. 1, 2025).

[25] Trend Micro, “Pushing the Outer Limits: Trend Micro 2024 Midyear Cybersecurity Threat Report,” Aug. 15, 2024. https://www.trendmicro.com/vinfo/us/security/research-and-analysis/threat-reports/roundup

[26] V. Álvarez, YARA Documentation, 2021. https://media.readthedocs.org/pdf/yara/latest/yara.pdf. (accedida May. 1, 2025).

[27] YARA Project, “Running YARA from the command-line — YARA 4.5.0 documentation.” https://yara.readthedocs.io/en/latest/commandline.html. (accedida May. 19, 2025).

[28] N. Naik et al., “Embedding Fuzzy Rules with YARA Rules for Performance Optimisation of Malware Analysis,” in Proc. 2020 IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE), Glasgow, UK, 2020, pp. 1–7, doi: 10.1109/FUZZ48607.2020.9177856.

[29] MITRE, “ATT&CK Matrix for Enterprise,” Jan. 1, 2024. https://attack.mitre.org/. (accedida Abr. 18, 2025).

[30] N. Sarantinos, C. Benzaid, O. Arabiat, and A. Al-Nemrat, “Forensic Malware Analysis: The Value of Fuzzy Hashing Algorithms in Identifying Similarities,” in 2016 IEEE Trustcom/BigDataSE/ISPA, 2016, pp. 1782–1787, doi: 10.1109/TrustCom.2016.0274.

[31] N. Naik, P. Jenkins, N. Savage, and L. Yang, “Cyberthreat Hunting - Part 1: Triaging Ransomware using Fuzzy Hashing, Import Hashing and YARA Rules,” in 2019 IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE), New Orleans, LA, USA, 2019, pp. 1–6, doi: 10.1109/FUZZ-IEEE.2019.8858803.

[32] ssdeep Project, "ssdeep - Fuzzy hashing program." https://ssdeep-project.github.io/ssdeep/index.html. (accedida May. 15, 2025).

[33] N. Naik et al., “Lockout-Tagout Ransomware: A Detection Method for Ransomware using Fuzzy Hashing and Clustering.” https://publications.aston.ac.uk/id/eprint/42000/

[34] N. Naik, P. Jenkins, N. Savage, L. Yang, K. Naik, and J. Song, “Augmented YARA Rules Fused With Fuzzy Hashing in Ransomware Triaging.” https://publications.aston.ac.uk/id/eprint/41999/

[35] N. Naik et al., “Fuzzy Hashing Aided Enhanced YARA Rules for Malware Triaging,” in 2020 IEEE SSCI, pp. 1138–1145, 2020. doi: 10.1109/SSCI47803.2020.9308189. [En línea]. Disponible: https://www.researchgate.net/publication/348262781

[36] Amazon Web Services, “Amazon Simple Storage Service (S3) - Almacenamiento en la nube - AWS.” https://aws.amazon.com/es/s3/faqs/. (accedida Jun. 1, 2025).

[37] Amazon Web Services, “What is Amazon S3? - Amazon Simple Storage Service.” https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html. (accedida Jun. 1, 2025).

[38] Amazon Web Services, “Expiring Amazon S3 Objects Based on Last Accessed Date to Decrease Costs,” Aug. 3, 2021. https://aws.amazon.com/es/blogs/architecture/expiring-amazon-s3-objects-based-on-last-accessed-date-to-decrease-costs/. (accedida Jun. 1, 2025).

[39] Amazon Web Services, “Comprenda los conceptos clave de Lambda - AWS Lambda.” https://docs.aws.amazon.com/es_es/lambda/latest/dg/gettingstarted-concepts.html (accedida Jun. 1, 2025).

[40] Amazon Web Services, “Field Notes: Optimize your Java application for AWS Lambda with Quarkus,” Nov. 23, 2022. https://aws.amazon.com/es/blogs/architecture/field-notes-optimize-your-java-application-for-aws-lambda-with-quarkus/ (accedida Jun. 6, 2025).

[41] J. Smith and A. Jones, “Extending YARA with Module Support for PE and ELF Files,” Int. J. Cyber-Secur. Digit. Forensics, vol. 8, no. 2, pp. 45–56, 2019.

[42] V. Mythily et al., “Malware Detection and Prevention Using Machine Learning,” in Challenges in Information, Communication and Computing Technology, CRC Press, 2024, pp. 564–569. doi: 10.1201/9781003559092-97

[43] M. S. Akhtar and T. Feng, “Evaluation of Machine Learning Algorithms for Malware Detection,” Sensors, vol. 23, no. 2, 2023. doi: 10.3390/s23020946

[44] M. Al-Janabi and A. M. Altamimi, “A Comparative Analysis of Machine Learning Techniques for Classification and Detection of Malware,” in 2020 21st Int. Arab Conf. Inf. Technol. (ACIT), Giza, Egypt, 2020, pp. 1–9. [En línea]. Disponible: https://ieeexplore.ieee.org/abstract/document/9300081

[45] I. R. A. Hamid et al., “Android Malware Classification Using K-Means Clustering Algorithm,” IOP Conf. Ser.: Mater. Sci. Eng., vol. 226, no. 1, 2017. doi: 10.1088/1757-899X/226/1/012105

[46] A. Bensaoud, N. Abudawaood, and J. Kalita, “Classifying Malware Images with Convolutional Neural Network Models,” Int. J. Netw. Secur., vol. 22, no. 6, 2020. doi: 10.6633/IJNS.202011_22(6).17

[47] A. Pinhero et al., “Malware detection employed by visualization and deep neural network,” Computers & Security, vol. 105, p. 102247, 2021. doi: 10.1016/j.cose.2021.102247

[48] T. M. Mohammed, L. Nataraj, S. Chikkagoudar, S. Chandrasekaran, and B. S. Manjunath, “Malware Detection Using Frequency Domain-Based Image Visualization and Deep Learning,” arXiv preprint, 2021. [En línea]. Disponible: http://arxiv.org/abs/2101.10578

[49] L. Meijin et al., “A Systematic Overview of Android Malware Detection,” Appl. Artif. Intell., vol. 36, no. 1, 2021. doi: 10.1080/08839514.2021.2007327

[50] O. Jurečková, M. Jureček, M. Stamp et al., “Classification and online clustering of zero-day malware,” J. Comput. Virol. Hack. Tech., vol. 20, pp. 579–592, 2024. doi: 10.1007/s11416-024-00513-5

[51] L. Garcia and M. Perez, “Complementing Static Analysis with Dynamic Techniques in Malware Detection,” IEEE Trans. Inf. Forensics Secur., vol. 14, no. 6, pp. 1532–1545, 2019.

[52] O. Lysne, “Static Detection of Malware,” in The Huawei and Snowden Questions, Simula SpringerBriefs on Computing, vol. 4, Cham: Springer, 2018. doi: 10.1007/978-3-319-74950-1_7

[53] Y. Guo, “A review of Machine Learning-based zero-day attack detection: Challenges and future directions,” Comput. Commun., vol. 198, pp. 175–185, 2023. doi: 10.1016/j.comcom.2022.11.001

[54] Y. Song et al., “Application of deep learning in malware detection: a review,” J. Big Data, vol. 12, no. 1, 2025. doi: 10.1186/s40537-025-01157-y

[55] H. Shaban, E. Nakashima, and R. Lerman, “JBS, world’s biggest meat supplier, says its systems are coming back online after cyberattack shut down plants in U.S.,” The Washington Post, Jun. 1, 2021. [En línea]. Disponible: https://www.washingtonpost.com/business/2021/06/01/jbs-cyberattack-meat-supply-chain/

[56] K. Good, “JBS Systems Coming Back Online After Ransomware Attack,” Farm Policy News, Jun. 1, 2021. [En línea]. Disponible: https://farmpolicynews.illinois.edu/2021/06/jbs-systems-coming-back-online-after-ransomware-attack/

[57] S. Liebermann, “Qué son los ransomware y cómo afecta a la economía,” MasDigital, Jun. 15, 2024. [En línea]. Disponible: https://masdigital.com.ar/abc/que-son-los-ransomware-y-como-afecta-a-la-economia/

[58] KnowBe4, “Free Ransomware Awareness Resource Kit.” https://www.knowbe4.com/ransomware-resource-kit (accedida Feb. 8, 2025).

[59] Veritas, “¿Qué es el ransomware?” Jan. 1, 2023. https://www.veritas.com/es/es/information-center/what-is-ransomware (accedida Jun. 1, 2025).

[60] D. Milmo, “Ministers consider ban on all UK public bodies making ransomware payments,” The Guardian, Jan. 14, 2025. [En línea]. Disponible: https://www.theguardian.com/technology/2025/jan/14/ministers-consider-ban-on-all-uk-public-bodies-making-ransomware-payments

[61] Y. Borboën, “Ransomware as a business model: Legal aspects of ransom payment,” PwC Switzerland, Apr. 1, 2022. [En línea]. Disponible: https://www.pwc.ch/en/insights/cybersecurity/ransom-payment.html

[62] V. M. H. Fernando, “Detección de amenazas informáticas de tipo Malware Bancario o Ransomware Móvil hacia dispositivos Android, integrando IOC en una técnica semiautomatizada y con base en comportamientos analizados de incidentes,” Repositorio ITM, 2022. [En línea]. Disponible: https://repositorio.itm.edu.co/handle/20.500.12622/5700

[63] R. H. Mahdi and H. Trabelsi, “Detection of Malware by Using YARA Rules,” in 2024 IEEE Int. Conf. on SSD, pp. 1–8, 2024. doi: 10.1109/ssd61670.2024.10549308

[64] N. Naik et al., “Evaluating Automatically Generated YARA Rules and Enhancing Their Effectiveness,” in 2020 IEEE SSCI, Dec. 1, 2020. [En línea]. Disponible: https://ieeexplore.ieee.org/abstract/document/9308179

[65] N. Naik et al., “Fuzzy Hashing Aided Enhanced YARA Rules for Malware Triaging,” in 2020 IEEE SSCI, Dec. 1, 2020. [En línea]. Disponible: https://ieeexplore.ieee.org/abstract/document/9308189

[66] N. Naik, P. Jenkins, N. Savage, and L. Yang, “Cyberthreat hunting - part 1: Triaging ransomware using fuzzy hashing, import hashing and YARA rules,” in Proc. 2019 IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE), 2019. doi: 10.1109/FUZZ-IEEE.2019.8858803

[67] N. Naik, P. Jenkins, N. Savage, and L. Yang, “Cyberthreat hunting - part 2: Tracking ransomware threat actors using fuzzy hashing and fuzzy C-means clustering,” in 2019 IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE), 2019.

[68] Amazon Web Services, “Amazon Simple Storage Service Documentation,” 2022. https://docs.aws.amazon.com/s3/?nc2=h_ql_doc_s3 (accedida Jun. 1, 2025).

[69] Amazon Web Services, “AWS Lambda Documentation.” https://docs.aws.amazon.com/lambda/?icmpid=docs_homepage_featuredsvcs (accedida Jun. 10, 2025).

[70] AWS, “AWS LMBDA.” https://ieeexplore.bibliotecaitm.elogim.com/document/10605158

[71] CISA, “#StopRansomware: LockBit 3.0 TLP:CLEAR,” 2023. https://www.cisa.gov/sites/default/files/2023-03/aa23-075a-stop-ransomware-lockbit.pdf (accedida Jun.11, 2025).

[72] CISA, “Understanding Ransomware Threat Actors: LockBit,” Jun. 14, 2023. https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-165a (accedida Jun. 11, 2025).

[73] CISA, “#StopRansomware: Akira Ransomware,” Apr. 18, 2024. https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-109a (accedida Jun. 11, 2025).

[74] J. A. [Jaime Andres], “A spotlight on Akira ransomware from X-Force incident response and threat intelligence,” IBM X-Force, May 7, 2024. [En línea]. Disponible: https://www.ibm.com/think/x-force/spotlight-akira-ransomware-x-force

[75] N. Naik, P. Jenkins, R. Cooke, J. Gillett, and Y. Jin, “Evaluating Automatically Generated YARA Rules and Enhancing Their Effectiveness,” IEEE, Dec. 2020. [En línea]. Disponible: https://ieeexplore.ieee.org/document/9308179

[76] M. Botacin, V. H. Galhardo Moia, F. Ceschin, M. A. Amaral Henriques, and A. Grégio, “Understanding uses and misuses of similarity hashing functions for malware detection and family clustering in actual scenarios,” Forensic Sci. Int.: Digit. Investigat., vol. 38, 2021. doi: 10.1016/j.fsidi.2021.301220

Anexos

A. Encuesta realizada a organizaciones hispanoamericanas que sufrieron eventos de seguridad relacionados con ransomware.
B. lambda_function-yara.zip: Paquete personalizado de regla YARA regular para desplegar en AWS Lambda.
C. lambda_function-yaraANDssdeep.zip: Paquete personalizado de regla YARA integrada con SSDEEP para desplegar en AWS Lambda.