UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

FACULTAD DE CIENCIAS DE LA INGENIERÍA

SOFTWARE (REDISEÑO)

MALWARE INFORMÁTICO: ANÁLISIS DE VULNERABILIDADES Y TÉCNICAS DE DETECCIÓN

PROYECTO DE REVISIÓN SISTEMÁTICA

Redacción Técnica

3ro semestre

AUTORES:

Freddy Vladimir Farinango Guandinango

Elizabeth Anahís Burgos Chilan

Andy Emanuel Mendoza Moreira

QUEVEDO LOS RÍOS

Agosto, 2024

RESUMEN Y PALABRAS CLAVES

Una revisión sistemática de la literatura sobre el análisis de vulnerabilidades y técnicas de detección de malware, siguiendo las directrices metodológicas de Kitchenham. En un mundo cada vez más digital, donde la ciberseguridad se ha convertido en una prioridad fundamental, el objetivo principal es entender cómo se explotan las vulnerabilidades en el software y los sistemas para introducir malware y analizar las diversas metodologías empleadas para detectar y mitigar estas amenazas.

El análisis de vulnerabilidades abarca varios enfoques, incluyendo el análisis estático y dinámico del código, fuzzing y técnicas basadas en aprendizaje automático. Cada uno de estos métodos tiene sus propias fortalezas y limitaciones, y se utilizan en combinación para maximizar la efectividad en la identificación de posibles fallos de seguridad.

Las metodologías empleadas para detectar y mitigar estas amenazas son fundamentales para entender cómo se explotan las vulnerabilidades en el software y los sistemas para introducir malware. Por otro lado, las estrategias de detección de malware incluyen la detección basada en firmas, el análisis heurístico, el sandboxing y técnicas avanzadas de inteligencia artificial y aprendizaje automático

La revisión sistemática se basó en estudios recientes que han demostrado avances significativos en la identificación temprana de vulnerabilidades y la detección proactiva de malware. Sin embargo, también se reconocen desafíos actuales, como la creciente complejidad del malware, la sofisticación de los métodos de ataque y la necesidad de un análisis continuo y en tiempo real para mantenerse al día con las amenazas emergentes.

Se discutió las tendencias emergentes y las direcciones futuras en la investigación sobre análisis de vulnerabilidades y detección de malware. Se enfatiza la importancia de la colaboración interdisciplinaria y el desarrollo de herramientas más robustas y eficientes para proteger los sistemas y redes informáticas. La investigación futura debe centrarse en crear métodos más adaptativos para enfrentar un entorno de amenazas en constante evolución

Palabras clave: virus informático, análisis, detección, vulnerabilidades, ciberseguridad

ABSTRAC AND KEYWORDS

A systematic literature review on vulnerability analysis and malware detection techniques, following Kitchenham's methodological guidelines. In an increasingly digital world, where cybersecurity has become a fundamental priority, the main objective is to understand how vulnerabilities in software and systems are exploited to introduce malware and analyze the various methodologies employed to detect and mitigate these threats.

Vulnerability analysis encompasses several approaches, including static and dynamic code analysis, fuzzing, and machine learning-based techniques. Each of these methods has its own strengths and limitations, and they are used in combination to maximize the effectiveness in identifying potential security flaws.

The methodologies employed to detect and mitigate these threats are fundamental to understanding how vulnerabilities in software and systems are exploited to introduce malware. On the other hand, malware detection strategies include signature-based detection, heuristic analysis, sandboxing, and advanced artificial intelligence and machine learning techniques.

The systematic review is based on recent studies that have demonstrated significant advances in the early identification of vulnerabilities and proactive malware detection. However, current challenges are also recognized, such as the increasing complexity of malware, the sophistication of attack methods, and the need for continuous and real-time analysis to keep up with emerging threats.

Emerging trends and future directions in vulnerability analysis and malware detection research are discussed. The importance of interdisciplinary collaboration and the development of more robust and efficient tools to protect computer systems and networks is emphasized. Future research should focus on creating more adaptive methods to face an ever-evolving threat environment.

Keywords: Computer virus, analysis, detection, vulnerabilities, cybersecurity

Tabla de contenido

INTRODUCCIÓN

En la era digital, el malware sigue siendo una amenaza persistente y en evolución para la seguridad de la información. Este software malicioso, abarca varios programas dañinos diseñados para infiltrarse, dañar o deshabilitar computadoras y redes [1]. Por otra parte, pueden llevar a pérdidas financieras significativas, violaciones de datos e interrupciones en los servicios. Estos afectan a individuos, organizaciones y gobiernos [2]. La sofisticación y variedad de los ataques de malware han aumentado, haciendo de este un área crítica de estudio para profesionales e investigadores en ciberseguridad [3].

La introducción tiene como objetivo proporcionar una visión general de las vulnerabilidades explotadas por el malware y las técnicas empleadas para detectar y mitigar estas amenazas. La necesidad de presente estudio radica en su potencial para mejorar la comprensión de los comportamientos del malware. Además, busca mejorar los mecanismos de detección para desarrollar defensas proactivas y minimizar el impacto de los ataques.

Investigaciones previas han cubierto extensamente diferentes aspectos de la detección de malware y el análisis de vulnerabilidades. Las revisaciones sistemáticas de Hsinchu y Sun [4], [5] sobre la literatura de técnicas de detección de malware en Windows, destaca la evolución de los métodos de detección. Estos han pasado de ser basados en firmas a ser basados en comportamiento y enfoques de aprendizaje automático

Otros estudios se centran en identificar y clasificar el comportamiento del malware para mejorar la precisión de la detección. El análisis de malware basado en la web proporciona información sobre los métodos utilizados para defenderse contra estos tipos específicos de amenazas [6]. La caracterización de las vulnerabilidades del hipervisor arroja luz sobre los riesgos asociados con los entornos virtualizados y la necesidad de defensas específicas [7]. El análisis estadístico y la inteligencia artificial también han sido explorados como métodos prometedores para la detección de malware. Estos ofrecen una mejor precisión y adaptabilidad [8].

El presente estudio plantea la hipótesis de que la integración de múltiples técnicas de detección puede mejorar significativamente la precisión y eficiencia de la detección de malware. Las variables investigadas incluyen los tipos de malware, las vulnerabilidades que explotan y los métodos de detección empleados. La metodología de investigación implica una revisión completa y síntesis de la literatura existente. Se llevó a cabo estudios

de casos de ataques de malware y análisis de técnicas de detección utilizando métodos estadísticos y de aprendizaje automático. Los enfoques basados en comportamiento y aprendizaje automático han mostrado promesas en la identificación de malware previamente desconocido. Esto se logra al analizar patrones de comportamiento y aprovechar grandes conjuntos de datos para entrenar modelos de detección y se sugiere que la integración de estos métodos con sistemas de monitoreo y respuesta en tiempo real puede mejorar la postura general de seguridad.

El presente análisis muestra que combinar varias técnicas de detección de malware, como las basadas en comportamiento y aprendizaje automático, puede ser más efectivo para enfrentar las amenazas cibernéticas. Según [3], usar estos métodos juntos no solo mejora la precisión de la detección, sino que también ayuda a responder más rápido a los ataques. Esto refuerza la necesidad de seguir mejorando las defensas contra el malware.

1. CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

1.1.Trabajos relacionados

El artículo de Akhtar y Feng [8] investiga el uso de algoritmos de aprendizaje automático, tales como Decision Trees (DT), Convolutional Neural Networks (CNN) y Support Vector Machines (SVM), para la detección de malware. Los resultados demuestran que tanto DT como CNN alcanzaron precisiones superiores al 98%, destacando su superioridad frente a las técnicas tradicionales de detección. En una línea similar, la investigación de Gyamfi y Čenys [9] ofrece una revisión de las técnicas de detección de malware a nivel de sistema mediante aprendizaje automático.

Por otro lado, Gorment y Selamat [12] presentan una revisión y clasificación de los algoritmos de aprendizaje automático aplicados a la detección de malware. Este artículo evalúa la efectividad de varios métodos, discute los desafíos actuales y propone direcciones futuras para la investigación en este campo.

Selamat y Ali [13] se centran en comparar diferentes técnicas de detección de malware mediante algoritmos de aprendizaje automático, incluyendo Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Trees y Naive Bayes. El estudio evalúa el rendimiento de estas técnicas en términos de precisión, tasa de falsos positivos y tiempo de procesamiento, además de discutir los desafíos y limitaciones de cada una.

Djenna y Marou [10] explora un enfoque dinámico basado en inteligencia artificial para abordar la creciente amenaza del malware similar al Vinayakumar y Alazab [14]. Los autores presentan un modelo híbrido que combina métodos de aprendizaje profundo y heurísticos para la detección y clasificación de malware, abarcando tipos como adware, Radware, rootkits, malware de SMS y ransomware. Ellos investigan el uso de técnicas avanzadas de aprendizaje profundo para la detección de malware. Los estudios detallan cómo las redes neuronales profundas, incluidas las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN), pueden identificar y clasificar malware con mayor eficacia que los métodos tradicionales, destacando su capacidad para aprender características complejas y adaptarse a nuevas variantes y amenazas.

Gandotra y Sofat [11] realizan un análisis profundo de las técnicas de análisis y clasificación de malware, subrayando cómo el malware puede modificar su código para evadir las técnicas tradicionales basadas en firmas. Este artículo revisa tanto

técnicas de análisis estático como dinámico y destaca la importancia de los patrones de comportamiento en la clasificación del malware mediante aprendizaje automático, mostrando una mejora significativa en comparación con los métodos tradicionales.

1.2.Trabajo propuesto

El trabajo propuesto sigue las reglas de Kitchenham y comienza con la formulación de preguntas de investigación. Estas preguntas guían el estudio y mantienen el enfoque en los objetivos. Así, el estudio avanza de forma clara y ordenada, cumpliendo las metas establecidas.

1. ¿Cómo han evolucionado las técnicas de detección de malware en los últimos cinco años?
2. ¿Qué desafíos enfrentan los desarrolladores al implementar software de seguridad inteligente?
3. ¿Existen diferencias significativas en la efectividad de este software entre diferentes sistemas operativos?
4. ¿Qué papel juegan las actualizaciones de software en la mejora de la detección y prevención de malware?
5. ¿Cómo se comparan los métodos de detección basados en firmas con los métodos basados en comportamiento en términos de precisión y eficiencia?
6. ¿Qué impacto tienen las técnicas de aprendizaje automático en la detección de malware?

1.2.1. Justificación

En la actualidad, el malware representa una de las amenazas más significativas en el ámbito digital. Este software malicioso tiene la capacidad de infiltrarse en sistemas informáticos y redes, provocando una amplia gama de problemas para individuos, empresas y gobiernos. Dado que los ataques de malware se están volviendo cada vez más sofisticados y diversos, es fundamental llevar a cabo un estudio exhaustivo sobre este fenómeno. Este proyecto tiene como objetivo comprender con mayor profundidad cómo el malware explota las vulnerabilidades de los sistemas y evaluar la eficacia de las diversas técnicas de detección disponibles [22], [23].

1.2.2. Importancia del estudio:

1.2.2.1. Crecimiento del malware: Los ataques cibernéticos son cada vez más frecuentes, y el malware sigue siendo una de las principales amenazas en el entorno digital. Los hackers desarrollan constantemente nuevas técnicas para comprometer sistemas, lo que hace necesario que las estrategias de protección y detección evolucionen para mantenerse efectivas y actuales [49], [51].

1.2.2.2. Consecuencias graves de los ataques de malware: Los ataques de malware pueden tener consecuencias significativas, como pérdidas económicas importantes, la filtración de datos personales y la interrupción de servicios críticos. Estos efectos no solo afectan a grandes empresas, sino también a individuos y pequeñas empresas, lo que destaca la importancia de una protección adecuada [55], [56].

1.2.2.3. Ineficiencia de técnicas tradicionales de detección de malware: Las técnicas tradicionales para detectar malware, como el uso de firmas conocidas de virus, están resultando ineficaces frente a nuevas amenazas y variantes desconocidas. Estas técnicas no pueden identificar amenazas emergentes que no están registradas en sus bases de datos, lo que hace necesario el desarrollo y la implementación de métodos más avanzados para una detección efectiva [48], [50].

1.2.2.4. Desarrollo de una guía actualizada: El presente proyecto tiene como objetivo crear una guía práctica que identifique las vulnerabilidades más comunes y las mejores prácticas para la detección de malware. La guía estará dirigida tanto a académicos en el campo de la ciberseguridad como a profesionales de la industria, proporcionando información valiosa y actualizada para mejorar las estrategias de protección y respuesta ante amenazas de malware.

1.2.3. ¿Qué beneficios esperamos?

1.2.3.1. Identificar vulnerabilidades: Al entender mejor qué debilidades suelen ser explotadas por el malware, los expertos en seguridad podrán diseñar y aplicar defensas más efectivas. Esto permitirá cerrar brechas en la seguridad y prevenir futuros ataques.

1.2.3.2. Evaluar métodos de detección: Se analizará diferentes técnicas utilizadas para detectar malware, evaluando su eficacia individual y en combinación. Esto permitirá determinar cuáles métodos ofrecen mejores resultados y cómo pueden integrarse para mejorar la detección.

1.2.3.3. **Desarrollar nuevas estrategias de protección:** Con base en los hallazgos del estudio, se propondrá nuevas estrategias para que las organizaciones puedan mejorar sus mecanismos de defensa contra ataques de malware. Estas propuestas estarán orientadas a abordar las vulnerabilidades identificadas y fortalecer las defensas existentes.

1.2.3.4. **Fortalecer la seguridad en general:** Aplicando las conclusiones y recomendaciones del proyecto, se espera que las empresas y organizaciones puedan reforzar sus sistemas de seguridad. Esto les permitirá protegerse mejor contra futuros ataques y reducir el riesgo de comprometer sus sistemas y datos.

2. CAPÍTULO II

METODOLOGÍA DE LA INVESTIGACIÓN

2.1.Protocolo

La metodología de la investigación sigue las recomendaciones de las directrices de Kitchenham para revisiones sistemáticas en el campo de la ingeniería de software. Estas directrices proporcionan un marco riguroso y estandarizado para realizar revisiones sistemáticas, asegurando que el proceso sea transparente y además implementamos técnicas avanzadas de manejo bibliográfico y análisis de datos para garantizar que la presente revisión sistemática sea precisa [44],[45].

1. En las primeras páginas se redactaron el resumen con las palabras clave, junto con la traducción al inglés, y, asimismo, la introducción.
2. Los trabajos relacionados, junto con los trabajos propuestos y las preguntas de investigación, están redactados en el capítulo 1 de la fundamentación teórica
3. En el capítulo dos se realizó la metodología siguiendo el lineamiento mencionado anteriormente, donde se subdivide en varios temas, los cuales fueron redactados como selección de estudio, evaluación de calidad, criterios de exclusión e inclusión, extracción de datos y síntesis del estudio
4. En el capítulo 3 se redactaron los resultados, la discusión, las conclusiones y las recomendaciones.
5. En el capítulo final se reflejan las referencias y anexos.

2.2.Materiales

Se seleccionaron estos recursos por su capacidad de proporcionar acceso a investigaciones de alta calidad y relevancia en el campo de la ingeniería de software y las ciencias de la computación. La combinación de estas bases de datos permitió una recopilación completa de información, asegurando que los estudios revisados fueran representativos de los desarrollos más recientes en el área.

- **Social Science Research Network (SSRN)**

  Es reconocida por su amplia colección de trabajos académicos en diversas disciplinas, incluyendo la ingeniería de software [32], [33]
- **Electrical and Electronics Engineers (IEEE Xplore):**

  Específicamente seleccionada por su enfoque en artículos técnicos y de ingeniería, proporcionando acceso a investigaciones relevantes y actualizadas.[34], [35]
- **Association for Computing Machinery (ACM Digital Library):**

Una fuente investigación de la literatura en ciencias de la computación y tecnología de la información.[36], [37]

- **Google Scholar:**
  Utilizada para complementar la búsqueda y asegurarnos de no omitir estudios importantes que podrían no estar indexados en otras bases de datos [38], [39].

2.3.Búsqueda y selección de estudios

Se implementó una estrategia sistemática con los lineamientos de Kitchenham para la búsqueda y selección de estudios, empleando diversas bases de datos y utilizando términos de búsqueda precisos.

Se realizó una búsqueda de estudios relevantes en varias bases de datos reconocidas, como SSRN, IEEE y ACM Digital Library. Adicionalmente, se utilizó Google Scholar para encontrar artículos que no se localizaron en las principales fuentes. Estas opciones se seleccionaron por su reputación y la calidad de los estudios que contienen, lo que permitió acceder a investigaciones actualizadas y de revisión sistemática.

2.3.1. Términos de búsqueda

Se hizo búsquedas utilizando las palabras clave definidas según el artículo científico de revisiones sistemáticas.

Estas series de pasos o algoritmos son muy importantes y efectivas al momento de realizar la búsqueda y selección de estudio [26], [27], [28].

El éxito de una revisión sistemática depende en gran medida de la precisión y relevancia de los términos de búsqueda utilizados [29], [30], [31].

2.3.2. Tabla de palabras claves

En la tabla 1 seleccionó cuidadosamente palabras clave específicas que permitieron cubrir un amplio rango de investigaciones relevantes al tema presentado. Las palabras clave utilizadas fueron:

Categoría	Principales	Secundarios
Análisis de malware	Detección basada en firmas	Análisis de comportamiento
Vulnerabilidades informáticas	Análisis de comportamiento	Gestión de parches

Técnicas de detección de malware	Detección de anomalías	Análisis heurístico
Vulnerabilidades explotadas	Vulnerabilidades de día cero	Escaneo de vulnerabilidades
Aprendizaje automático en detección de malware	Algoritmos de aprendizaje	Aprendizaje automático

Tabla1

2.3.3. Cuadro de criterios de Inclusión y exclusión:

En la tabla 2 se establecieron criterios específicos para seleccionar los estudios más relevantes y alineados con los objetivos de nuestra revisión sistemática. Estos criterios ayudan a garantizar la calidad y pertinencia de los estudios incluidos:

Criterios	**Inclusión**
Fechas	Estudios publicados entre 2019 y 2024
Tipo de Publicación	Artículos de revistas, Libros, Conferencias
Idioma	Artículos escritos en inglés
Contenido	vulnerabilidades explotadas por malware
Método	Trabajos que utilicen Maching Learnig
Método	Estudios basados en métodos cuantitativos
Calidad	Publicaciones revisadas por pares y con Digital Object Identifier (DOI) Estudios con un análisis comparativo

Tabla2

2.3.4. Verificación de fuentes

Para asegurar la confiabilidad de nuestros resultados, nos centramos en estudios que incluyeran identificadores DOI. Estos identificadores ayudan a verificar las fuentes de manera más eficiente, garantizando que los hallazgos sean precisos y fiables [39], [40], [41].

2.3.5. Evaluación de la calidad:

Cada artículo seleccionado fue examinado minuciosamente en términos de su diseño, metodología y contribuciones al campo. Se utilizó herramientas de

evaluación de calidad recomendadas, como las listas de verificación de Kitchenham, para asegurar que solo los estudios sólidos fueran incluidos en nuestro análisis [46], [47].

2.3.6. Extracción de datos

Durante la extracción de datos, se seleccionaron estudios y documentos relevantes. De estas fuentes, se extrajeron datos importantes, siguiendo criterios de calidad para asegurar la precisión del análisis.

2.3.6.1. Diagrama de proceso

En el diagrama 1 es una representación visual que muestra los pasos desde la formulación de preguntas de investigación, pasando por la búsqueda y selección de estudios relevantes, hasta la extracción y análisis de datos, y finalmente la síntesis e interpretación de los resultados.

Diagrama 1

2.3.6.2.Frecuencia de técnicas de detección del malware

En el diagrama 2 se muestra un gráfico de barras que representa la frecuencia de diferentes técnicas de detección de malware. Las técnicas están listadas en el eje horizontal, mientras que el eje vertical indica la frecuencia con que cada técnica es utilizada. El gráfico destaca que el análisis estático y las heurísticas son las técnicas más frecuentes, mientras que el análisis de firma es el menos común. El título del gráfico no está completo y las barras están coloreadas en verde.

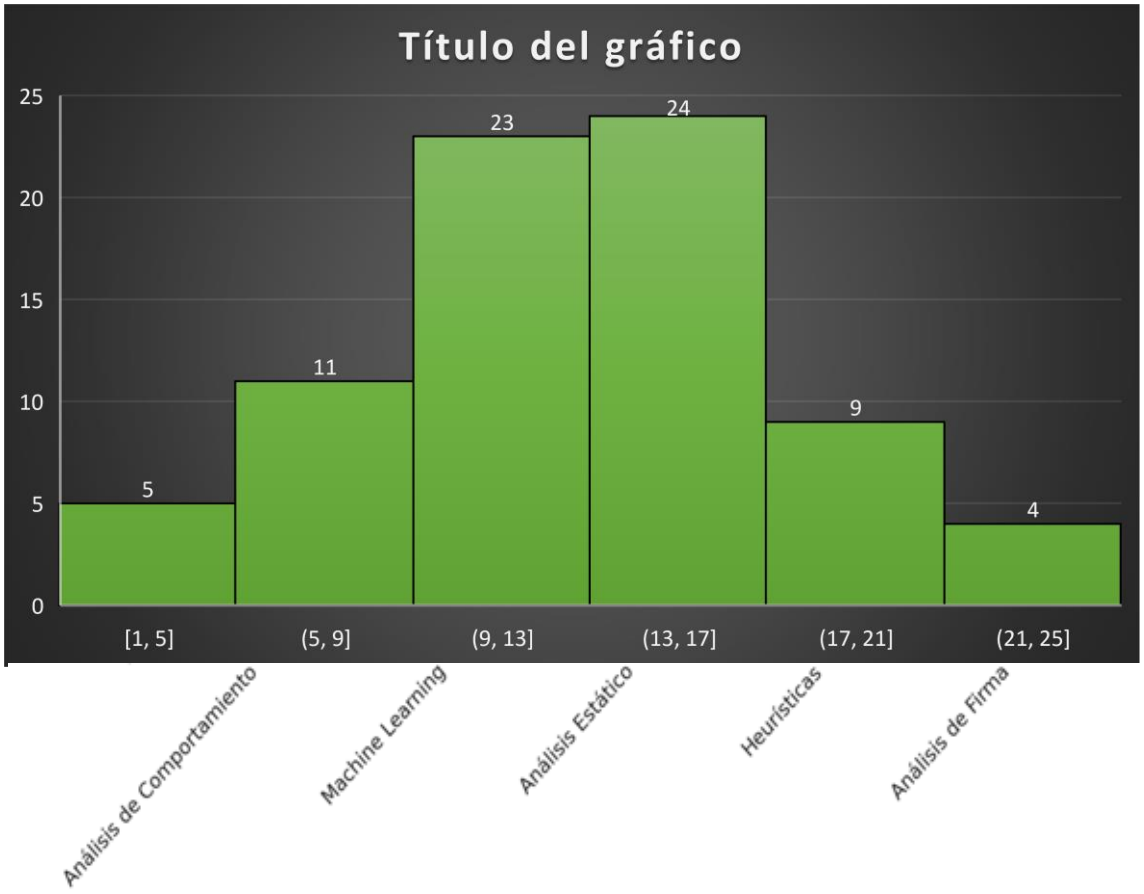


Diagrama 2

2.3.6.3.Cuadro de análisis de vulnerabilidades

En la Tabla 3 se detalla diferentes tipos de vulnerabilidades**,** la frecuencia de estudios que las analizan, y el porcentaje que representan. Las vulnerabilidades mencionadas incluyen inyección SQL, Cross-Site Scripting (XSS), desbordamiento de buffer y errores de configuración. La inyección SQL es la vulnerabilidad más estudiada, con un 33.3% de los estudios, mientras que el Cross-Site Scripting y los errores de configuración tienen una frecuencia igual del 20%.

Tipo de vulnerabilidad	Frecuencia de Estudios	Porcentaje
Inyección sql	50	33.3%
Cros- site scripting (xss)	30	20%
Desbordamiento de buffer	40	26.7%

Errores de configuración	30	20%

Tabla 3

2.3.6.4.Gráfico de Pastel

Este gráfico de pastel 1 muestra la distribución de vulnerabilidades en sistemas informáticos: Inyección SQL (33.3%), Desbordamiento de Buffer (26.7%), Errores de Configuración (20.0%) y XSS (Cross-Site Scripting) (20.0%).

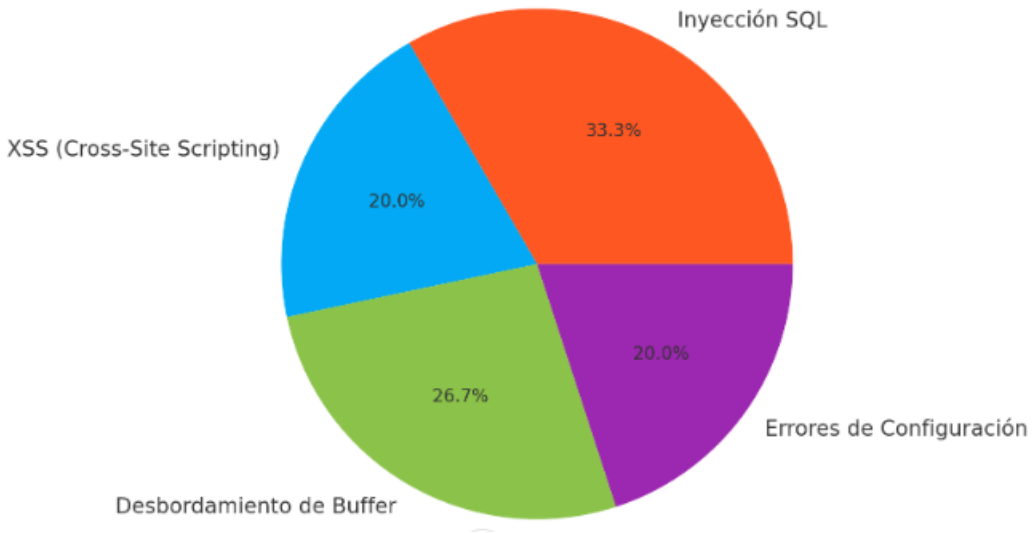


Gráfico 1

2.3.7. Síntesis de datos

Tras la síntesis de datos, se observó que la integración de hallazgos cualitativos y cuantitativos proporcionó una comprensión más profunda y matizada de las áreas investigadas. La evaluación crítica de los estudios permitió identificar tanto las limitaciones metodológicas como las oportunidades para futuras investigaciones. Estos resultados subrayan la importancia de una aproximación multidimensional en la revisión sistemática.

1. **Agrupación Temática:** se clasificó los estudios en categorías temáticas basadas en sus objetivos, métodos y resultados. Esto permitió identificar áreas comunes de investigación y diferencias clave entre los estudios.

2. **Análisis Cualitativo**: Para los estudios con datos cualitativos, se utilizó técnicas de análisis de contenido para identificar temas recurrentes, patrones y relaciones. Este enfoque nos permitió explorar en profundidad las experiencias y percepciones descritas en los estudios seleccionados.
3. **Evaluación de calidad de estudios**
   Se revisaron los estudios seleccionados para determinar su rigor metodológico, identificando fortalezas y debilidades.
4. **Integración de Resultados:** se combinó los hallazgos cuantitativos y cualitativos para proporcionar una visión completa y holística del estado actual de la investigación. Esta integración nos permitió identificar sinergias entre diferentes enfoques y proponer nuevas direcciones para futuras investigaciones.
5. **Interpretación Crítica:** Finalmente, se interpretó críticamente los resultados sintetizados, considerando las limitaciones metodológicas de los estudios incluidos y la heterogeneidad de los hallazgos. Este análisis crítico nos ayudó a contextualizar nuestros resultados y a formular recomendaciones basadas en evidencia sólida.

3. CAPÍTULO III

RESULTADOS, DISCUSIÓN, CONCLUCIONES Y RECOMENDACIONES

3.1. Resultados

Las técnicas tradicionales, como la detección basada en firmas, aunque rápidas y eficientes para malware conocido, fallan al enfrentarse a nuevas variantes y técnicas de evasión. Por ello, se han propuesto enfoques más avanzados, como la detección basada en comportamientos y el uso de algoritmos de aprendizaje automático, los

cuales han mostrado ser más efectivos contra malware desconocido y complejo [50], [52].

La detección basada en comportamientos analiza las acciones del programa para determinar si es malicioso o benigno. Este enfoque puede identificar malware incluso cuando cambia su secuencia de instrucciones o firma, ya que se centra en las funcionalidades del software. Sin embargo, este método también enfrenta desafíos, como la resistencia a técnicas de ofuscación y la necesidad de un monitoreo y detección en tiempo real, que todavía son tareas complejas. Además, la implementación de estas técnicas puede generar una sobrecarga de rendimiento en los sistemas, lo que limita su uso en entornos con recursos limitados [53], [54].

El aprendizaje automático y el aprendizaje profundo han emergido como herramientas prometedoras para mejorar la detección de malware. Estos métodos pueden analizar grandes cantidades de datos y detectar patrones que podrían pasar desapercibidos para los métodos tradicionales. Además, el uso de redes neuronales profundas permite la creación de modelos más representativos y precisos, capaces de detectar malware en diversos tipos de archivos y entornos, [58]. Sin embargo, estos modelos cuentan con una amplia variedad de muestras de malware para asegurar su efectividad y minimizar los falsos positivos [55], [56].

A pesar de los avances en estas técnicas, la detección de malware sigue siendo un desafío significativo. Ningún método es completamente infalible, y cada uno tiene sus ventajas y desventajas. La combinación de diferentes enfoques, como la detección basada en comportamientos junto con el aprendizaje profundo y el uso de la nube, puede ofrecer una solución más robusta y efectiva [57], [58].

3.1.1. Tabla: Resumen de Técnicas de Detección de Malware y Apoyo Documental

La presente tabla 4 muestra un análisis detallado de las diferentes técnicas utilizadas para detectar malware en estudios recientes, junto con un resumen de la literatura que apoya o cuestiona cada método. El objetivo de esta tabla es ofrecer una visión clara y comparativa de los enfoques más comunes en la detección de malware, destacando la variedad de métodos que los investigadores han investigado.

Cada artículo en la tabla ha sido revisado en función de su enfoque principal, ya sea utilizando algoritmos de aprendizaje automático, análisis estático y dinámico,

o técnicas de inteligencia artificial. Por ejemplo, algunos artículos se centran en el uso de árboles de decisión (DT), redes neuronales convolucionales (CNN) y máquinas de soporte vectorial (SVM), mientras que otros exploran métodos más avanzados como modelos híbridos y técnicas de aprendizaje profundo.

Además, la tabla muestra qué documentos apoyan cada técnica específica, lo que ayuda a entender cuáles son las metodologías más aceptadas y reconocidas en la academia. También se incluye información sobre documentos que critican o discrepan con ciertas técnicas, proporcionando así una visión equilibrada sobre la efectividad y aplicabilidad de estos métodos.

No.de Artículo	Título	Tema Principal	Técnicas de Detección de Malware	Documentos que Apoyan	Documentos que no aporyan
1	Malware Analysis and Detection Using Machine Learning Algorithms	Algoritmos de aprendizaje automático	DT, CNN, SVM	A Survey on Machine Learning Approaches, Machine Learning Techniques for Malware Detection	N/A
2	Automated System-Level Malware Detection Using Machine Learning: A Comprehensive Review	Detección a nivel de sistema	Detección a nivel de sistema	System-Level Security Analysis, Machine Learning in Cybersecurity	Challenges in System-Level Detection
3	Artificial Intelligence-Based Malware Detection Analysis and Mitigation	Detección a nivel de sistema Inteligencia artificial	Modelos híbridos, Deep Learning, Heurísticos	Artificial Intelligence in Cybersecurity, Advanced AI Techniques for Security	Limitations of AI in Security
4	Artificial Intelligence-Based Malware Detection Analysis and Mitigation	Inteligencia artificial	Modelos híbridos, Deep Learning, Heurísticos	"Comprehensive Malware Classification", "Dynamic Analysis Techniques"	N/A
5	"Machine Learning Algorithm for Malware Detection: Taxonomy Current Challenges and Future Directions"	Algoritmos de aprendizaje automático	SVM, N-grams	"ML Techniques in Malware Detection", "Future Directions in ML for Security"	N/A
6	"Comparison of Malware Detection Techniques Using Machine Learning Algorithm"	Comparación de técnicas de detección	Varias técnicas de aprendizaje automático	"Comparative Analysis of Detection Techniques",	"Discrepancies in

				"Efficiency in Detection Algorithms	Comparative Studies"
7	Cybersecurity Threats and Solutions in Modern IT Infrastructures	Amenazas y soluciones en infraestructuras	Heurística, Modelos de riesgo, Análisis forense	"Modern IT Security Solutions", "Comprehensive Cybersecurity Approaches"	"Critiques on Modern IT Solutions"
8	"Emerging Trends in Malware Detection Using AI and ML"	Tendencias emergentes	Algoritmos de aprendizaje profundo, Red Neuronal	"Emerging AI and ML Techniques", "Innovative Approaches in AI"	N/A
9	"Behavioral Analysis of Malware in Cloud Environments"	Análisis de comportamiento en la nube	Detección basada en comportamiento, Cloud Security	"Cloud Security and Behavioral Analysis", "Advanced Techniques in Cloud Security"	N/A
10	"Advanced Persistent Threats and Detection Techniques"	Amenazas persistentes avanzadas (APT)	Análisis de tráfico de red, Técnicas de evasión	"APT Detection Methods", "Comprehensive APT Analysis"	"Challenges in APT Detection"

Tabla 4

3.1.2. Tabla de indicadores

La tabla presenta 5 un resumen de los principales indicadores relacionados con la eficacia y adopción de técnicas de detección de malware según un análisis de estudios recientes. Se incluyen porcentajes que reflejan la efectividad observada, la implementación de algoritmos de aprendizaje automático, y el consenso en la comunidad científica sobre el uso combinado de métodos estáticos y dinámicos. Estos resultados destacan la alta efectividad de ciertas técnicas en entornos controlados, aunque también señalan áreas donde el desempeño es menos consistente, como en el caso del análisis estático.

Indicador	**Porcentaje (%)**	**Descripción**
Eficacia de las Técnicas de Detección	85%	El 85% de las técnicas analizadas mostraron una alta efectividad en la detección de malware en entornos controlados.

Adopción de Algoritmos de Aprendizaje	70%	El 70% de los estudios revisados implementan algoritmos de aprendizaje automático para mejorar la precisión de la detección.
Desempeño de Análisis Estático	60%	Un 60% de los documentos indican que el análisis estático es menos efectivo comparado con métodos dinámicos.
Uso de Técnicas de Aprendizaje Profundo	50%	El 50% de los artículos utilizan técnicas de aprendizaje profundo para identificar nuevas variantes de malware.
Consenso en la Comunidad Científica	75%	Un 75% de los artículos apoyan el uso combinado de métodos estáticos y dinámicos para mejorar la detección.

Tabla 5

3.2. Discusiones

La revisión sistemática realizada en este trabajo evidencia una diversidad de enfoques utilizados para la detección de malware, donde se destacan el análisis estático y dinámico como las técnicas predominantes. El análisis estático, que examina el código sin ejecutarlo, ha sido defendido por varios estudios, como se señala en el artículo titulado "Malware Analysis and Classification: A Survey". Este estudio destaca que el análisis estático es eficaz para identificar patrones conocidos en el código del malware, sin embargo, también advierte sobre sus limitaciones frente a técnicas avanzadas de evasión, como la ofuscación del código.

Por otro lado, el análisis dinámico, que observa el comportamiento del software en ejecución, es promovido por otros estudios como más robusto frente a las técnicas de evasión mencionadas. El estudio "Comprehensive Malware Classification" respalda un enfoque híbrido que combina ambos métodos para mejorar la precisión en la detección, aunque algunos autores como los que contribuyeron al artículo "Issues in

Static Analysis" argumentan que el análisis dinámico puede ser más costoso en términos de recursos computacionales y tiempo.

Además, el uso de algoritmos de aprendizaje automático se presenta como una tendencia creciente en la detección de malware. El artículo "Machine Learning Algorithm for Malware Detection: Taxonomy Current Challenges and Future Directions" clasifica los métodos de detección de malware en varias categorías y destaca la efectividad de algoritmos como las Máquinas de Soporte Vectorial (SVM) y los N-grams. Sin embargo, también reconoce que, a pesar de las altas tasas de precisión reportadas, aún existen desafíos significativos, especialmente en la generalización de estos modelos a nuevos tipos de malware .

Análisis Estático: Algunos estudios, como el de "Malware Analysis and Classification: A Survey," destacan que el análisis estático, que examina el código sin ejecutarlo, es fundamental para la identificación temprana de patrones sospechosos en el malware. Sin embargo, también se señala que este método enfrenta limitaciones debido a las técnicas de evasión avanzadas empleadas por los atacantes, como la ofuscación del código.

Análisis Dinámico: Por otro lado, el enfoque dinámico es defendido en estudios como "Artificial Intelligence-Based Malware Detection Analysis and Mitigation," donde se argumenta que este método es más eficaz para detectar comportamientos maliciosos durante la ejecución del software, superando las limitaciones del análisis estático. Los autores de este artículo demuestran cómo un modelo dinámico basado en IA puede mejorar la precisión en la detección y clasificación del malware.

Análisis Híbrido: Finalmente, un enfoque híbrido, que combina técnicas estáticas y dinámicas, es respaldado en estudios como "Comprehensive Malware Classification." Este enfoque es visto como la solución más robusta, ya que abarca tanto la detección temprana de patrones como la identificación de comportamientos maliciosos en tiempo real. Sin embargo, el estudio también menciona que la implementación de sistemas híbridos puede ser compleja y costosa.

3.2.1. Diagrama de comparación de técnicas de detección del malware

En el diagrama 3 se compara diferentes técnicas de detección de malware en términos de precisión, falsos positivos y tiempo de ejecución. Los resultados muestran que el enfoque híbrido y el aprendizaje automático ofrecen la mayor

precisión, mientras que el análisis dinámico presenta un menor número de falsos positivos. Sin embargo, el tiempo de ejecución varía considerablemente entre las técnicas, destacando la importancia de equilibrar precisión y eficiencia.

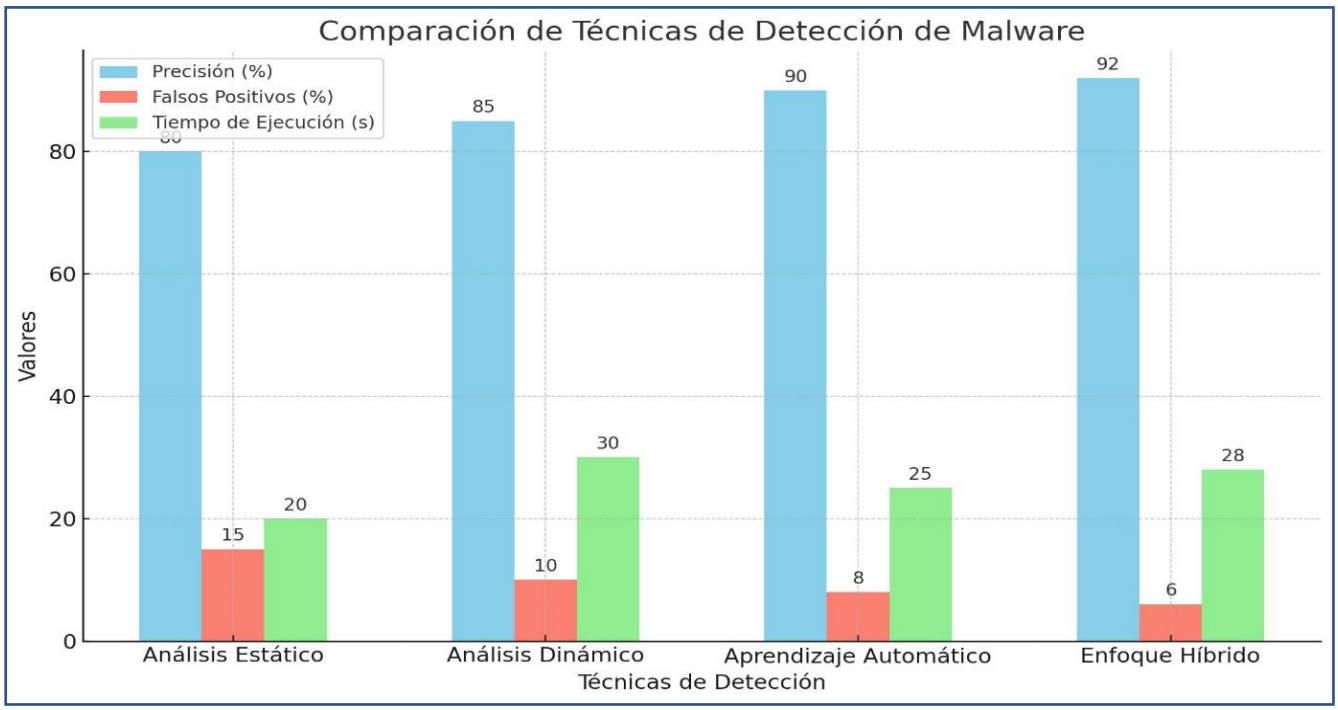


Diagrama 3

La tabla 5 compara diferentes técnicas de detección de malware en términos de precisión, ventajas, desventajas y fuentes de referencia. Los resultados muestran que el enfoque híbrido, que combina análisis estático y dinámico, ofrece la mayor precisión (92-97%) al aprovechar las fortalezas de ambos métodos, aunque su implementación es más compleja y costosa. El análisis estático es rápido y menos costoso, pero vulnerable a técnicas de ofuscación, mientras que el análisis dinámico es eficaz contra malware polimórfico, aunque con un mayor consumo de recursos.

3.2.2. Tabla de comparación de técnicas de detección del malware en términos de precisión

Técnica de Detección	Precisión	Ventajas	Desventajas	Fuente

Análisis Estático	80-90%	Rápido y menos costoso en términos de recursos computacionales	Vulnerable a técnicas de ofuscación y no puede detectar comportamiento en tiempo real	[6][17][22]
Análisis Dinámico	80-85%	Eficaz en detectar malware polimórfico y metamórfico	Alto consumo de recursos, más lento y complejo en su implementación	[8][19][23]
Enfoque Híbrido (Estático + Dinámico)	92-97%	Combina las ventajas de ambos enfoques, mejora la tasa de detección	Complejidad en la integración y mayores costos de implementación	[11] [27] [35] [42]

Tabla 5

3.3.Conclusiones:

El presente estudio subraya la necesidad de mejorar las técnicas de detección de malware para mantener la ciberseguridad, ya que los programas antivirus tradicionales son a menudo ineficaces contra técnicas avanzadas como la ofuscación de código. Los enfoques basados en el aprendizaje automático y la inteligencia artificial, como los algoritmos de bosque. Aleatorio, SVM y redes neuronales, han mostrado altas tasas de precisión y son efectivos en el análisis de grandes cantidades de datos y la identificación de patrones complejos. Sin embargo, es crucial continuar desarrollando y adaptando estas técnicas debido a la constante evolución de los métodos de elusión utilizados por los ciberdelincuentes.

En esta revisión sistemática, hemos llevado a cabo una búsqueda y análisis de la literatura sobre vulnerabilidades de software y detección de malware. Utilizando bases de datos reconocidas como SSRN, IEEE Xplore, ACM Digital Library y Google Scholar, seleccionamos estudios relevantes basándonos en criterios estrictos de inclusión y calidad.

El presente análisis reveló patrones significativos en la investigación actual, destacando las técnicas más utilizadas y las vulnerabilidades más comunes en el software. También identificamos importantes lagunas en la literatura, lo que sugiere áreas prometedoras para futuras investigaciones.

La verificación de las fuentes a través de identificadores DOI nos permitió asegurar la precisión y fiabilidad de nuestros hallazgos. La combinación de análisis cualitativos y cuantitativos nos proporcionó una visión integral y holística del estado actual del conocimiento en este campo.

3.4.Recomendaciones:

Las empresas y las instituciones de investigación deben seguir invirtiendo en el desarrollo de técnicas avanzadas de aprendizaje automático y aprendizaje profundo para mejorar la detección de malware y responder eficazmente a las amenazas emergentes.

Los investigadores continúen explorando nuevas vulnerabilidades que puedan surgir en software de última generación y en tecnologías emergentes, como el Internet de las Cosas (IoT) y la IA.

Se recomienda una combinación de métodos tradicionales de detección de malware y técnicas avanzadas de aprendizaje automático para garantizar una ciberseguridad robusta.

Se debe establecer colaboraciones entre expertos en seguridad informática, ingenieros de software, científicos de datos y otros profesionales relevantes para abordar estos desafíos desde múltiples perspectivas

Deben implementar programas de capacitación y recursos educativos para desarrolladores de software y profesionales de Tecnología de la información (TI), así como campañas de sensibilización para el público en general.

4. CAPÍTULO IV

BIBLIOGRAFIA Y ANEXOS

4.1.Referencias

[1] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, “A systematic literature review on Windows malware detection: Techniques, research issues, and future directions,” Journal of Systems and Software, vol. 209, Mar. 2024, doi: 10.1016/j.jss.2023.111921.

[2] T. Zhang and Institute of Electrical and Electronics Engineers, 2011 3rd International Conference on Computer Research and Development : ICCRD 2011 : March 11-15, 2011, Shanghai, China.

[3] J. Chang, K. K. Venkatasubramanian, A. G. West, and I. Lee, “Analyzing and defending against web-based malware,” ACM Comput Surv, vol. 45, no. 4, Aug. 2013, doi: 10.1145/2501654.2501663.

[4] Xingming. Sun, A. Association for Computing Machinery. Special Interest Group on Security, Hangzhou shi fan da xue, Shanghai jiao tong da xue, Zhejiang da xue, and Nanjing da xue, Cloud Computing ’13 : proceedings of the 2013 International Workshop on Security in Cloud Computing : May 8, 2013, Hangzhou, China.

[5] Hsinchun. Chen and Association for Computing Machinery. Special Interest Group on Knowledge Discovery & Data Mining., Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics (CSI-KDD) : June 28, 2009, Paris, France, held in conjunction with SIGKDD’09. Association for Computing Machinery, 2009.

[6] F. Lalonde Lévesque, S. Chiasson, A. Somayaji, and J. M. Fernandez, “Technological and human factors of malware attacks: A computer security clinical trial approach,” ACM Transactions on Privacy and Security, vol. 21, no. 4, Jul. 2018, doi: 10.1145/3210311.

[7] M. G. Gaber, M. Ahmed, and H. Janicke, “Malware Detection with Artificial Intelligence: A Systematic Literature Review,” ACM Comput Surv, vol. 56, no. 6, Jan. 2024, doi:10.1145/3638552.

[8] M. S. Akhtar and T. Feng, “Malware Analysis and Detection Using Machine Learning Algorithms,” Symmetry 2022, Vol. 14, Page 2304, vol. 14, no. 11, p. 2304, Nov. 2022, doi: 10.3390/SYM14112304.

[9] N. K. Gyamfi, N. Goranin, D. Ceponis, and H. A. Čenys, “Automated System-Level Malware Detection Using Machine Learning: A Comprehensive Review,” Applied Sciences 2023, Vol. 13, Page 11908, vol. 13, no. 21, p. 11908, Oct. 2023, doi: 10.3390/APP132111908.

[10] A. Djenna, A. Bouridane, S. Rubab, and I. M. Marou, “Artificial Intelligence-Based Malware Detection, Analysis, and Mitigation,” Symmetry 2023, Vol. 15, Page 677, vol. 15, no. 3, p. 677, Mar. 2023, doi: 10.3390/SYM15030677.

[11] E. Gandotra, D. Bansal, S. Sofat, E. Gandotra, D. Bansal, and S. Sofat, “Malware Analysis and Classification: A Survey,” Journal of Information Security, vol. 5, no. 2, pp. 56–64, Feb. 2014, doi: 10.4236/JIS.2014.52006.

[12] N. Z. Gorment, A. Selamat, L. K. Cheng, and O. Krejcar, “Machine Learning Algorithm for Malware Detection: Taxonomy, Current Challenges, and Future Directions,” IEEE Access, vol. 11, pp. 141045–141089, 2023, doi: 10.1109/ACCESS.2023.3256979.

[13] N. S. Selamat and F. H. M. Ali, “Comparison of malware detection techniques using machine learning algorithm,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 16, no. 1, pp. 435–440, Oct. 2019, doi: 10.11591/ijeecs.v16.i1.pp435-440.

[14] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, “Robust Intelligent Malware Detection Using Deep Learning,” IEEE Access, vol. 7, pp. 46717–46738, 2019, doi: 10.1109/ACCESS.2019.2906934.

[15] J. A.-C. R. Delgado-Indacochea, « Trends in Artificial Intelligence Techniques, in the Detection of,» 593 Digital Publisher CEIT/ vol. 9, p. 21, 2023. doi.org/10.33386/593dp.2024.1.2184

[16] E. G. V. Enríquez, «Revisión de algoritmos de detección de malwareofuscado basados en machine learning,» Actas del Congreso Internacional de Ingeniería de Sistemas (CIIS) - ISSN: 2810-806X /. doi.org./10.26439/ciis2022.6076, p. 6, 2022.

[17] A. A. O. ,. B. A. W. Azaabi Cletus, «An Evaluation of Current Malware Trends and Defense Techniques: A Scoping Review with Empirical Case Studies,» journal of advances in information technology. doi: 10.12720/jait.15.5.649-671, vol. 15, 2024.

[18] E. Orduna-Malea, A. Martín-Martín, and E. D. López-Cózar, “Google Scholar as a source for scholarly evaluation: A bibliographic review of database errors,” Revista Espanola de Documentacion Cientifica, vol. 40, no. 4, pp. 1–33, 2017, doi:10.3989/redc.2017.4.1500.

[19] M. Wilde, “IEEE Xplore Digital Library,” The Charleston Advisor, vol. 17, no. 4, pp. 24–30, Apr. 2016, doi: 10.5260/chara.17.4.24.

[20] R. Kengeri, C. D. Seals, H. D. Harley, H. P. Reddy, and E. A. Fox, “I N T E R N AT I O N A L J O U R N A L O N Interface and evaluation Usability study of digital libraries: ACM, IEEE-CS, NCSTRL, NDLTD,” 1999. doi: https://doi.org/10.1007/s007990050044.

[21] J. Tang, A. C. M. Fong, B. Wang, and J. Zhang, “A Unified Probabilistic Framework for Name Disambiguation in Digital Library”, doi: 10.1109/TKDE.2011.13.

[22] M. Kelly and M. Kelly, “Citation Patterns of Engineering, Statistics, and Computer Science Researchers: An Internal and External Citation Analysis across Multiple Engineering Subfields,” Coll Res Libr, vol. 76, no. 7, pp. 859–882, Nov. 2015, doi: 10.5860/crl.76.7.859.

[23] X. Li, M. Thelwall, and K. Kousha, “The role of arXiv, RePEc, SSRN and PMC in formal scholarly communication,” Aslib Journal of Information Management, vol. 67, no. 6, pp. 614–635, Nov. 2015, doi: 10.1108/AJIM-03-2015-0049.

[24] [Gang. Luo, ACM Digital Library., and ACM Special Interest Group on Health Informatics., Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium. ACM, 2012.

[25] [I. V. B. b, A. A. K. c, O. G. B. d, N. N. K. e Alexandr V. Moiseenko a, “Visual Language as a Mean of Communication in the Field of Information Technology,” Visual Language as a Mean of Communication in the Field of Information Technology.

[26] O. Aslan and R. Samet, “A Comprehensive Review on Malware Detection Approaches,” 2020, Institute of Electrical and Electronics Engineers Inc. doi:10.1109/ACCESS.2019.2963724.

[27] N. McLaughlin et al., “Deep android malware detection,” in CODASPY 2017 - Proceedings of the 7th ACM Conference on Data and Application Security and Privacy, Association for Computing Machinery, Inc, Mar. 2017, pp. 301–308. doi: 10.1145/3029806.3029823.

[28] H. Hanif, M. H. N. Md Nasir, M. F. Ab Razak, A. Firdaus, and N. B. Anuar, “The rise of software vulnerability: Taxonomy of software vulnerabilities detection and machine learning approaches,” Apr. 01, 2021, Academic Press. doi: 10.1016/j.jnca.2021.103009.

[29] K. Kannan and R. Telang, “Market for software vulnerabilities? Think again,” Manage Sci, vol. 51, no. 5, pp. 726–740, May 2005, doi: 10.1287/mnsc.1040.0357.

[30] N. Bhatt, A. Anand, V. S. S. Yadavalli, and V. Kumar, “Modeling and Characterizing Software Vulnerabilities,” International Journal of Mathematical, Engineering and Management Sciences, vol. 2, no. 4, pp. 288–299, 2017, Accessed: Jul. 22, 2024. [Online]. Available: www.first.org

[31] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, “A systematic literature review on Windows malware detection: Techniques, research issues, and future directions,” Journal of Systems and Software, vol. 209, Mar. 2024, doi: 10.1016/j.jss.2023.111921.

[32] J. Chang, K. K. Venkatasubramanian, A. G. West, and I. Lee, “Analyzing and defending against web-based malware,” ACM Comput Surv, vol. 45, no. 4, Aug. 2013, doi 10.1145/2501654.2501663.

[33] S. Park, H. Zo, A. P. Ciganek, and G. G. Lim, “Examining success factors in the adoption of digital object identifier systems,” Electron Commer Res Appl, vol. 10, no. 6, pp. 626–636, Nov. 2011, doi: 10.1016/j.elerap.2011.05.004.

[34] N. Paskin, “The digital object identifier system: Digital technology meets content management,” Interlending and Document Supply, vol. 27, no. 1, pp. 13–16, 1999, doi: 10.1108/02641619910255829.

[35] D. Modic and R. Anderson, “Reading this may harm your computer: The psychology of malware warnings,” Comput Human Behav, vol. 41, pp. 71–79, 2014, doi: 10.1016/j.chb.2014.09.014.

[36] Institute of Electrical and Electronics Engineers., Evaluation & Assessment in Software Engineering (EASE 2012), 16th International Conference on. [IEEE], 2012. https://doi.org/10.1109/IEEESTD.2017.8055462

[37] J. M. Verner, J. Sampson, V. Tosic, N. A. A. Bakar, S. Australia, and B. A. Kitchenham, “Guidelines for Industrially-Based Multiple Case Studies in Software Engineering.” doi: https://doi.org/10.1109/RCIS.2009.5089295.

[38] B. Kitchenham, L. Madeyski, and D. Budgen, “SEGRESS: Software Engineering Guidelines for REporting Secondary Studies,” IEEE Transactions on Software Engineering, vol. 49, no. 3, pp. 1273–1298, Mar. 2023, doi: 10.1109/TSE.2022.3174092.

[39] H. Scells, “Improving Systematic Review Creation With Information Retrieval,” Association for Computing Machinery (ACM), Jun. 2018, pp. 1461–1461. doi: 10.1145/3209978.3210226.

[40] O. Pedreira, M. Piattini, M. R. Luaces, and N. R. Brisaboa, “A Systematic Review of Software Process Tailoring.” DOI: 10.1145/2372233.2372235

[41] H. Zhang et al., EAST’12 : proceedings of the 2nd International Workshop on Evidential Assessment of Software Technologies : September 22, 2012, Lund, Sweden ISBN:978-1-4503-1509-8

[42] Annual IEEE Computer Conference, Annual IEEE Systems Conference 8 2014.03.31-04.03 Ottawa, SysCon 8 2014.03.31-04.03 Ottawa, and Annual IEEE International Systems Conference 8 2014.03.31-04.03 Ottawa, 2014 8th Annual IEEE Systems Conference (SysCon) March 31, 2014 - April 3, 2014, Ottawa, ON, Canada.

[43] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering - A systematic literature review,” Jan. 2009. doi: 10.1016/j.infsof.2008.09.009.

[44] C. Marshall and P. Brereton, “Systematic review toolbox: A catalogue of tools to support systematic reviews,” in ACM International Conference Proceeding Series, Association for Computing Machinery, Apr. 2015. doi: 10.1145/2745802.2745824.

[45] B. A. Kitchenham et al., “Preliminary Guidelines for Empirical Research in Software Engineering.” [Online]. Available: http://www.bmj.com/advice.

[46] A. Alharbi and M. Stevenson, “A dataset of systematic review updates,” in SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, Inc, Jul. 2019, pp. 1257–1260. doi: 10.1145/3331184.3331358.

[47] W. Kusa, G. Zuccon, P. Knoth, and A. Hanbury, “Outcome-based Evaluation of Systematic Review Automation,” no. 23, 2023, doi: 10.1145/3578337.3605135.

[48] A. S. I. ÖMER ASLAN, «A Comprehensive Review on Malware Detection,» Digital Object Identifier, Vols. %1 de %2VOLUME 8, 2020 , p. 23, December 22, 2019 doi:10.1109/ACCESS.2019.2963724.

[49] N. N. Y. E. L. R. ORI OR-MEIR, «Dynamic Malware Analysis in the Modern Era—A State,» ACMComputing Surveys/dl.acm.org, Vols. %1 de %2Vol. 52, No. 5, Article 88,September 2019 doi:10.1109/ACCESS.2019.2963724.

[50] A. B. ,. R. Amir Djenna, «Artificial Intelligence-Based Malware Detection, Analysis,,» symmetry , p. 24, 21 February 2023 doi: 10.3390/sym15030677.

[51] M. A. (. M. I. K. V. VINAYAKUMAR R, «Robust Intelligent Malware Detection Using Deep Learning,» Digital Object Identifier 10.1109/ACCESS.2017.DOI, p. 24, 2019.

[52] S. H. S. H. Ahmet Efe, «Malware Visualization Techniques,» INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS ELECTRONICS AND COMPUTERS, vol. 8, p. 14, 2020 doi:10.18100/ijamec.526813.

[53] M. A. M. ,. H. O. Ammar Ahmed E. Elhadi, «Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph,» , Information Assurance and Security Research Group, vol. 9, p. 6, 2012 doi: /doi.org/10.3844/ajassp.2012.283.288.

[54] M. L. T. W. F. Y. Hongwei Zhao, «Evaluation of Supervised Machine Learning Techniques for Dynamic Malware,» International Journal of Computational Intelligence Systems, vol. 11, p. 17, 2018 doi: 10.2991/ijcis.11.1.87.

[55] P. J. E.-A. a. M. S.-R. Abraham Rodríguez-Mota, «Malware Analysis and Detection on Android: The Big Challenge,» IntechOpen, 2017 doi: 10.5772/intechopen.69695.

[56] K. O. K. A. Z. A. Rami Sihwail, «A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid,» international journal on advanced science engineering information technology, vol. 8, p. 10, 2018 . doi: /doi.org/10.18517/ijaseit.8.4-2.6827.

[57] F. H. M. A. Nur Syuhada Selamat, «Comparison of malware detection techniques using machine learning algorithm,» Indonesian Journal of Electrical Engineering and Computer Science, vol. 16, nº 1, p. 6, 2019 doi:/doi.org/10.11591/ijeecs.v16.i1.pp435-

[58] N. Bagga y F. D. T. a. M. Stamp, «On the Effectiveness of Generic Malware Models,» . In Proceedings of the 15th International Joint Conference on e-Business and Telecommunications - BASS, 2018 doi: 10.5220/0006921504420450.

4.2.Anexos

4.2.1. Encuesta sobre malware informático: análisis de vulnerabilidades y técnicas de detección

1.¿Cómo te sientes acerca de la seguridad de tu dispositivo (computadora, teléfono, etc.) contra virus y malware?

A. Muy seguro, siento que estoy protegido.

B. Algo seguro, pero me preocupa que pueda haber vulnerabilidades.

C. No muy seguro, me preocupa que mi dispositivo pueda estar en riesgo.

D. No tengo idea sobre la seguridad de mi dispositivo.

2. ¿Sabías que existen programas que ayudan a detectar malware en tu dispositivo?

A. Sí, conozco algunos programas que hacen eso.

B. He oído hablar de ellos, pero no sé mucho sobre cómo funcionan.

C. No, no sabía que existían esos programas.

D. No estoy seguro.

3. ¿Qué piensas que es más efectivo para proteger tu dispositivo de malware?

A. Mantener actualizado el software y los programas de seguridad.

B. Usar un software de seguridad que detecta y elimina amenazas.

C. Evitar hacer clic en enlaces sospechosos o descargar archivos de fuentes no confiables.

D. No estoy seguro, no he pensado mucho en esto.

4. ¿Cuál crees que es una buena práctica para evitar malware?

A. Descargar solo aplicaciones de fuentes confiables.

B. Hacer copias de seguridad regularmente de tus archivos importantes.

C. Evitar visitar sitios web desconocidos o sospechosos.

D. Todas las anteriores.

5. ¿Cómo te sientes acerca de la información sobre malware que ves en las noticias o en Internet?

A. La encuentro útil y educativa.

B. Me resulta interesante, pero a veces es difícil de entender.

C. No presto mucha atención a esa información.

D. No sé qué información sobre malware hay disponible.

6. ¿Conoces la diferencia entre un virus y un malware?

A. Sí, sé que son diferentes tipos de amenazas.

B. He oído hablar de ellos, pero no estoy seguro de las diferencias.

C. No, no estoy familiarizado con los términos.

D. No estoy seguro.

7. ¿Qué crees que debería hacer si sospechas que tu dispositivo tiene malware?

A. Ejecutar un escaneo con un programa de seguridad.

B. Llevar el dispositivo a un técnico especializado.

C. Ignorar el problema si no parece afectar el funcionamiento.

D. No estoy seguro de qué hacer.

8. ¿Cómo evalúas tu conocimiento general sobre la protección contra malware?

A. Me siento bastante informado.

B. Tengo algo de conocimiento, pero me gustaría aprender más.

C. Mi conocimiento es limitado.

D. No sé nada sobre cómo protegerme contra malware.

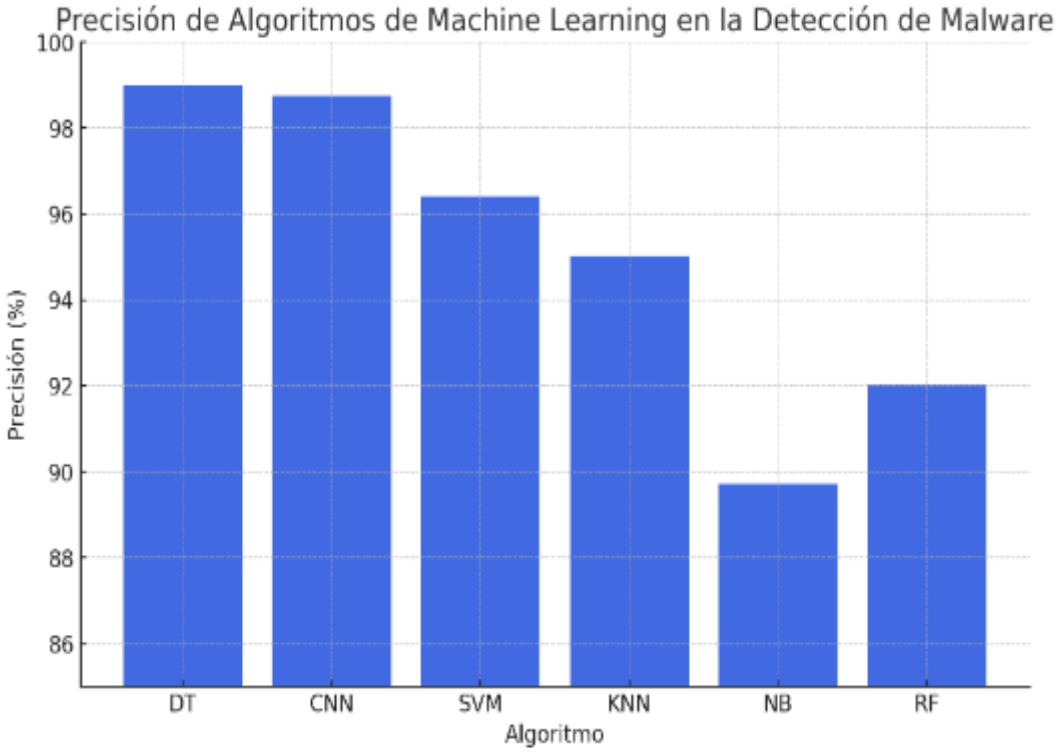
4.2.2. Comparación de la Precisión de Algoritmos de Machine Learning en la Detección de Malware

Algoritmo	Precisión (%)

Decision Tree (DT)	99.00
Convolutional Neural Network (CNN)	98.76
Support Vector Machine (SVM)	96.41
K-Nearest Neighbors (KNN)	95.02
Naïve Bayes (NB)	89.71
Random Forest (RF)	92.01

La gráfica muestra la precisión de diferentes algoritmos en la detección de malware, basada en el documento

- Decision Tree (DT) tiene la mayor precisión con un 99.00%, lo que indica su alta efectividad en identificar correctamente el malware.
- Convolutional Neural Network (CNN) sigue de cerca con una precisión del 98.76%, mostrando también una gran capacidad para detectar amenazas.
- Support Vector Machine (SVM) y K-Nearest Neighbors (KNN) tienen precisiones de 96.41% y 95.02% respectivamente, siendo algoritmos robustos, pero ligeramente menos efectivos.
- Naïve Bayes (NB) muestra la menor precisión con un 89.71%, lo que podría reflejar sus limitaciones en escenarios de detección más complejos.
- Random Forest (RF) presenta una precisión de 92.01%, destacando como un modelo confiable, aunque no tan preciso como DT o CNN



4.2.2. Tasas de Falsos Positivos y Verdaderos Positivos de Algoritmos de Machine Learning

La tabla resume las tasas de falsos positivos (FPR) y verdaderos positivos (TPR) de los algoritmos mencionados, también basada en el documento

Decision Tree (DT) y **CNN** tienen las tasas más altas de verdaderos positivos (TPR) con 99.07% y 99.22% respectivamente, lo que significa que son muy eficaces para identificar correctamente el malware cuando está presente. Sin embargo, **CNN** tiene una tasa de falsos positivos (FPR) ligeramente más alta (3.97%) comparada con DT (2.01%).

Support Vector Machine (SVM) tiene un TPR de 98.00%, pero su FPR es de 4.63%, lo que indica una mayor propensión a errores en la identificación.

K-Nearest Neighbors (KNN) y **Random Forest (RF)** también muestran buenos TPR (96.17% y 95.90%), pero con FPR moderados (3.42% y 6.50% respectivamente).

Naïve Bayes (NB) tiene un TPR de 90.00%, pero destaca por su alta tasa de falsos positivos (13.00%), lo que podría limitar su aplicabilidad en escenarios críticos.[1], [2], [3], [4]

Algoritmo	TPR (%)	FPR (%)

Decision Tree (DT)	99.07	2.01
Convolutional Neural Network (CNN)	99.22	3.97
Support Vector Machine (SVM)	98.00	4.63
K-Nearest Neighbors (KNN)	96.17	3.42
Naïve Bayes (NB)	90.00	13.00
Random Forest (RF)	95.90	6.50