Ingeniería Investiga

Vol. 6, e1078, año 2024
ISSN – Online: 2708-3039
DOI: https://doi.org/10.47796/ing.v6i00.1078

Artículo original

Implementación de un sistema antimalware inteligente para detección de enlaces maliciosos en códigos QR

Implementation of an intelligent antimalware system for the detection of malicious links in QR codes

Francisco Gerardo Huamanchumo Trujillo[1]
https://orcid.org/0009-0007-1937-4036

Alejandro Roman Campos Gamarra[2]
https://orcid.org/0009-0009-8492-1133

Rodrigo Alonso Guevara Saldaña[3]
https://orcid.org/0009-0002-6323-3517

Alberto Carlos Mendoza De Los Santos[4]
https://orcid.org/0000-0002-0469-915X

Recibido: 02/11/2024
Aceptado: 07/12/2024
Publicado: 17/12/2024

[1,2,3,4]Escuela de Ingeniería de Sistemas, Universidad Nacional de Trujillo, La Libertad, Perú

E-mail: [1]t1023300821@unitru.edu.pe, [2]t1513300121@unitru.edu.pe, [3]t1053300621@unitru.edu.pe, [4]amendozad@unitru.edu.pe

Facultad de Ingeniería
Publicación Oficial

Resumen

El aumento del uso de códigos QR en diversos sectores ha facilitado la transferencia de información, pero también ha expuesto a los usuarios a nuevas amenazas cibernéticas, como el *quishing*, una variante del *phishing* que utiliza estos códigos para redirigir a sitios maliciosos. Ante este problema, el estudio tuvo como objetivo implementar un sistema antimalware que emplea aprendizaje automático junto con la API de VirusTotal para analizar y clasificar enlaces embebidos en códigos QR en tiempo real. La metodología se estructuró en cuatro etapas: captura y decodificación de códigos QR mediante OpenCV, análisis de URLs extraídas utilizando la API de VirusTotal, emisión de alertas preventivas según la clasificación del enlace y evaluación del desempeño con un conjunto de datos de 100 códigos QR (50 seguros y 50 maliciosos). Los resultados mostraron una precisión del 100 %, una sensibilidad del 95 % y un tiempo de respuesta promedio de 48,95 ms. No se detectaron falsos positivos y se observó un bajo número de falsos negativos, aunque algunos códigos quedaron clasificados como inciertos debido a la falta de información en VirusTotal. Se concluye que el sistema es una herramienta adecuada y adaptable para prevenir ataques de *quishing*, con potencial para su implementación en aplicaciones móviles y sistemas de pago, y posibles expansiones a otras tecnologías de codificación visual.

Palabras clave: amenazas informáticas; aprendizaje automático; ciberseguridad.

Abstract

The increasing use of QR codes across various sectors has facilitated the transfer of information but has also exposed users to new cybersecurity threats, such as quishing, a variant of phishing that leverages these codes to redirect users to malicious websites. To address this issue, the study aimed to implement an antimalware system that employs machine learning alongside the VirusTotal API to analyze and classify links embedded in QR codes in real time. The methodology was structured into four stages: capturing and decoding QR codes using OpenCV, analyzing extracted URLs with the VirusTotal API, issuing preventive alerts based on the link classification, and evaluating system performance with a dataset of 100 QR codes (50 safe and 50 malicious). The results showed 100 % accuracy, 95 % sensitivity, and an average response time of 48.95 ms. No false positives were detected, and only a small number of false negatives were observed, although some codes were classified as uncertain due to insufficient information from VirusTotal. It is concluded that the system is a suitable and adaptable tool for preventing quishing attacks, with potential for implementation in mobile applications and payment systems, as well as possible expansions to other visual encoding technologies.

Keywords: cyber threats; machine learning; cybersecurity.

1. Introducción

La digitalización ha transformado la forma en que las personas interactúan con el mundo, elevando la conectividad y la accesibilidad. Desde la incorporación de Internet en la vida cotidiana, las tecnologías digitales redefinieron casi todos los aspectos de la sociedad, incluyendo el comercio, la educación, la salud y las relaciones sociales (Amoah y Hayfron-Acquah, 2022). Sin embargo, junto con estos avances tecnológicos surgieron nuevas vulnerabilidades y amenazas en los sistemas de información. La transición hacia un entorno digital interconectado aumentó la exposición de datos sensibles y personales a actores malintencionados, lo que ha generado una creciente preocupación por la ciberseguridad (Saeed et al., 2023). Entre las diversas amenazas cibernéticas, los ataques de phishing han sido uno de los más prevalentes, con métodos en constante evolución para eludir las medidas de seguridad.

En este contexto, los códigos QR (Quick Response code, "código de respuesta rápida") han ganado relevancia como una herramienta digital. Estos códigos son matrices bidimensionales, capaces de almacenar información que puede ser escaneada y transmitida de forma inmediata, fueron inicialmente desarrolladas en Japón para rastrear piezas automotrices (Japan Patent Office, 2022). Con el paso del tiempo, su uso ha crecido considerablemente, convirtiéndose en un medio popular para acceder a información de manera eficiente. Su presencia se ha expandido a diversos sectores como la educación, la salud, el comercio y la publicidad, donde facilitan la interacción directa de los usuarios con entornos virtuales mediante un simple escaneo. Los códigos QR permiten la transferencia rápida de datos, tales como enlaces web, detalles de productos, pagos y accesos a plataformas en línea, lo que los convierte en una herramienta esencial en la vida cotidiana (Hossain et al., 2018; Tiwari, 2016). Gracias a su versatilidad y facilidad de implementación, se han integrado de forma generalizada tanto en espacios físicos como virtuales, actuando como un puente entre el mundo físico y el online (Krombholz et al., 2014; Petrova et al., 2016).

Sin embargo, esta creciente dependencia de los códigos QR ha dado lugar a una nueva amenaza cibernética: el *quishing*, una variante del *phishing* que utiliza estos códigos como medio para llevar a cabo ataques. En un ataque de *quishing*, los cibercriminales manipulan códigos legítimos o crean versiones maliciosas para redirigir a los usuarios a sitios web fraudulentos. Estos sitios pueden solicitar información sensible, como contraseñas, credenciales bancarias o datos personales, que luego son utilizados para cometer fraudes o robos de identidad. Los ataques de quishing se aprovechan de la confianza que los usuarios depositan en los códigos QR, al considerarlos una forma rápida y segura de acceder a información. Debido a esto, rara vez se verifican antes de ser escaneados, lo que hace que los ataques sean difíciles de detectar, ya que los usuarios no pueden confirmar la autenticidad de los enlaces de forma inmediata (Sharevski et al., 2022; Slashnext, 2023). A pesar de los esfuerzos educativos y de concienciación para alertar a los usuarios sobre los riesgos asociados con códigos maliciosos, los incidentes de *quishing* siguen en aumento. Los atacantes continúan perfeccionando sus métodos para engañar a los usuarios, lo que resalta la necesidad de implementar soluciones más eficaces que protejan a los usuarios antes de que interactúen con estos códigos (Yong et al., 2019). Las herramientas actuales, como la verificación manual y las aplicaciones antivirus, resultan insuficientes para contrarrestar la velocidad con que los atacantes adaptan sus técnicas. Esto resalta la urgencia de desarrollar sistemas automatizados

que puedan analizar y clasificar los enlaces en los códigos QR en tiempo real, alertando a los usuarios de manera preventiva sobre posibles amenazas.

Una solución prometedora para abordar el *quishing* es el uso de técnicas de aprendizaje automático (*machine learning*), que permiten a los sistemas reconocer patrones en grandes volúmenes de datos y adaptarse a nuevas amenazas. Estos algoritmos mejoran la eficiencia y precisión en la detección de enlaces maliciosos en códigos QR, incluso aquellos con características desconocidas (Pawar et al., 2022; Sarkhi y Mishra, 2024). Además, al integrar plataformas de validación externa como VirusTotal, se refuerza la fiabilidad del sistema al validar los enlaces en tiempo real mediante múltiples motores antivirus, proporcionando una capa adicional de seguridad y aumentando la efectividad en la prevención de ataques (Spinelli et al., 2021; Spinelli y Dreizzen, 2021; Wahsheh y Luccio, 2020).

Por lo tanto, el objetivo del presente estudio fue implementar un sistema antimalware automatizado que emplea técnicas de aprendizaje automático junto con la API (Interfaz de Programación de Aplicaciones, por sus siglas en inglés *Application Programming Interface*) de VirusTotal, para analizar y clasificar los enlaces integrados en códigos QR en tiempo real, generando alertas preventivas para evitar que los usuarios accedan a sitios maliciosos.

2. Metodología

La implementación del sistema antimalware inteligente se organizó en varias etapas: captura y decodificación de los códigos QR, análisis y clasificación de los enlaces extraídos, emisión de alertas preventivas y evaluación del desempeño. Se recopilaron enlaces URL, se transformaron en códigos QR y se etiquetaron. Luego, se analizaron los códigos QR, midiendo el tiempo de respuesta y clasificando los resultados. Finalmente, se calcularon métricas clave y se estudiaron anomalías en las métricas de algunos códigos QR. Las fases se describen a continuación:

2.1. Captura y decodificación de códigos QR

La primera etapa consistió en la captura y decodificación de códigos QR mediante la biblioteca OpenCV, una herramienta ampliamente utilizada en el procesamiento de imágenes. El sistema inició cargando una imagen que contenía un código QR. Posteriormente, empleó algoritmos de detección para localizar y decodificar el contenido del código QR. Usando un script en Python. En los casos en que se detectó un código válido, el sistema extrajo el enlace o la información contenida en él. Si no se encontró un código QR, se notificó al usuario sobre la ausencia de información válida. Este proceso fue implementado mediante el código de la Figura 1. Esta fase fue fundamental para asegurar que el sistema pudiera trabajar con diferentes tipos de imágenes y proporcionar resultados rápidos y confiables.

2.2. Análisis y clasificación de enlaces

Una vez extraído el enlace del código QR, el siguiente paso fue el análisis y clasificación de este enlace, que se realizó mediante la integración de la API de VirusTotal. Esta herramienta permite analizar URLs utilizando más de 70 motores de antivirus y bases de datos actualizadas. El sistema enviaba una solicitud HTTP GET a la API de VirusTotal, utilizando la URL extraída del código QR y una clave API personal. VirusTotal devolvía un informe en formato JSON con la información sobre la seguridad de la URL. En función de los resultados obtenidos, el sistema

clasificaba el enlace en una de tres categorías: “seguro”, si no se detectaron riesgos; “malicioso”, si se identificaron amenazas; o “información insuficiente”, si la URL no había sido previamente analizada por VirusTotal. Este proceso se implementó mediante el código que se ilustra en la Figura 2.

Figura 1

Codificación para leer un código QR desde una imagen

```python
def leer_qr(imagen_path):
    # Cargar la imagen usando OpenCV
    imagen = cv2.imread(imagen_path)

    # Iniciar el detector QR
    detector_qr = cv2.QRCodeDetector()

    # Detectar y decodificar el código QR
    data, vertices_array, _ = detector_qr.detectAndDecode(imagen)

    if vertices_array is not None:
        print(f"Código QR detectado: {data}")
        return data
    else:
        print("No se detectó ningún código QR")
        return None
```

Figura 2

Codificación para verificar si una URL es segura utilizando la API de virusTotal

```python
def verificar_url_segura(url, api_key):
    # Endpoint de la API de VirusTotal
    endpoint = "https://www.virustotal.com/vtapi/v2/url/report"

    # Parámetros de la solicitud
    params = {'apikey': api_key, 'resource': url}

    # Realizar la solicitud a la API de VirusTotal
    response = requests.get(endpoint, params=params)
    result = response.json()

    # Verificar el estado de la URL en VirusTotal
    if result['response_code'] == 1:
        positives = result.get('positives', 0)
        total = result.get('total', 0)
        print(f"La URL ha sido reportada en {positives} de {total} análisis.")
        if positives > 0:
            return False
        else:
            return True
    else:
        print("La URL no ha sido analizada previamente.")
        return None
```

2.3. Emisión de alertas preventivas

En esta sección, el sistema generó alertas preventivas en tiempo real para informar al usuario sobre el estado del enlace. Las alertas se clasificaron en tres categorías principales: Si el enlace se clasificaba como malicioso, se emitía una alerta inmediata advirtiendo sobre los riesgos potenciales. Si el enlace se consideraba seguro, el sistema confirmaba que el usuario podía interactuar sin problemas. En los casos en los que no había suficiente información para clasificar la URL, el sistema informaba al usuario sobre la falta de datos y le recomendaba proceder con precaución. La funcionalidad de esta etapa fue implementada mediante el Código que se detalla en la Figura 3. Este enfoque aseguró que los usuarios pudieran tomar decisiones informadas antes de interactuar con cualquier enlace, lo que resulta esencial para prevenir ataques de pishing.

Figura 3

Codificación para escanear QR y verificar si la URL es segura

```
def escanear_qr_y_verificar(imagen_path, api_key):
    # Leer el código QR desde la imagen
    url = leer_qr(imagen_path)

    if url:
        # Verificar si la URL es segura
        es_segura = verificar_url_segura(url, api_key)

        if es_segura is None:
            print("No hay suficiente información sobre la URL.")
        elif es_segura:
            print("La URL es segura.")
        else:
            print("¡Alerta! La URL es potencialmente maliciosa.")
    else:
        print("No se pudo leer una URL del código QR.")
```

2.4. Evaluación del desempeño

Para evaluar el desempeño del sistema, se utilizó un conjunto de datos compuesto por 100 códigos QR, de los cuales 50 correspondían a enlaces seguros y 50 a enlaces maliciosos. Las URLs utilizadas provenían del conjunto de datos de Sahingoz et al. (2023), diseñado específicamente para entrenar modelos de aprendizaje profundo. Este conjunto de datos incluía direcciones IP y dominios similares a los utilizados en campañas de phishing, lo que permitió simular situaciones reales de amenaza. Las métricas utilizadas para la evaluación fueron: precisión, sensibilidad, tasa de error, tiempo de respuesta y tiempo de respuesta promedio.

La precisión se calculó como la proporción de enlaces maliciosos correctamente identificados sobre todos los enlaces clasificados como maliciosos, utilizando la Ecuación 1:

$$P = \frac{VP}{VP + FP} \quad (1)$$

Donde VP son los verdaderos positivos (códigos QR maliciosos correctamente identificados) y FP son los falsos positivos (códigos QR maliciosos identificados incorrectamente como seguros). Una alta precisión es esencial para evitar clasificaciones incorrectas que puedan generar desconfianza en los usuarios.

Por otro lado, la sensibilidad se midió como el porcentaje de enlaces maliciosos correctamente clasificados, mediante la ecuación 2:

$$S = \frac{VP}{VP + FN} \quad (2)$$

Donde FN son los falsos negativos (códigos QR seguros identificados incorrectamente como maliciosos).

Además, la tasa de error se calculó como la proporción de códigos QR clasificados incorrectamente sobre el total de códigos analizados, según la ecuación 3:

$$TE = \frac{FP + FN + CI}{Número\ total\ de\ códigos\ QR\ analizados} \quad (3)$$

Donde CI son los códigos QR clasificados como inciertos, es decir, aquellos que no pudieron ser identificados debido a la falta de información.

Mientras que, el tiempo de respuesta midió el tiempo necesario para que el sistema procesara un código QR, mientras que el tiempo de respuesta promedio se calculó como el promedio de los tiempos de respuesta de todos los códigos QR analizados, utilizando la ecuación 3:

$$TRP = \frac{\sum TR}{Número\ de\ códigos\ QR\ analizados} \quad (3)$$

Donde TR representa el tiempo de respuesta para cada código QR.

3. Resultados

Tras realizar el análisis de desempeño, los resultados obtenidos para la clasificación de los códigos QR se presentan en la Tabla 1.

Tabla 1

Codificación para escanear QR y verificar si la URL es segura

Códigos QR	Cantidad	Seguros	No seguros	Inciertos
Seguros	50	40	0	10
Maliciosos	50	2	46	2

Se destaca que la mayoría de los códigos QR clasificados como inciertos provienen de los códigos etiquetados como seguros. Esto se debe a que la API de VirusTotal no dispone de

suficiente información sobre algunas URLs para determinar si son seguras o no. Esta falta de información provoca que la clasificación de estos códigos QR quede en la categoría de inciertos.

En tanto a las métricas de desempeño obtenidas para la clasificación, la Tabla 2 y la figura 4 muestra los valores de verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP), falsos negativos (FN), precisión, sensibilidad, tasa de error y tiempo de respuesta promedio.

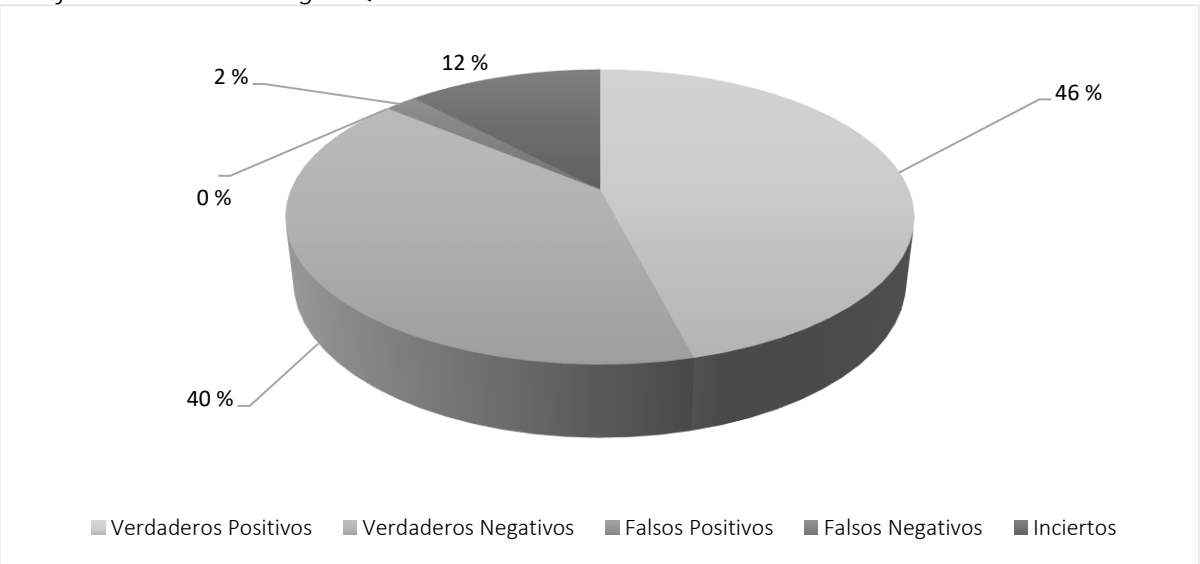
Tabla 2

Valores de verdadero positivo

VP	VN	FP	FN	Precisión	Sensibilidad	Tasa de error	Tiempo de respuesta promedio
46	40	0	2	100 %	95 %	14 %	48,95 ms

Figura 4

Clasificación de los códigos QR analizados



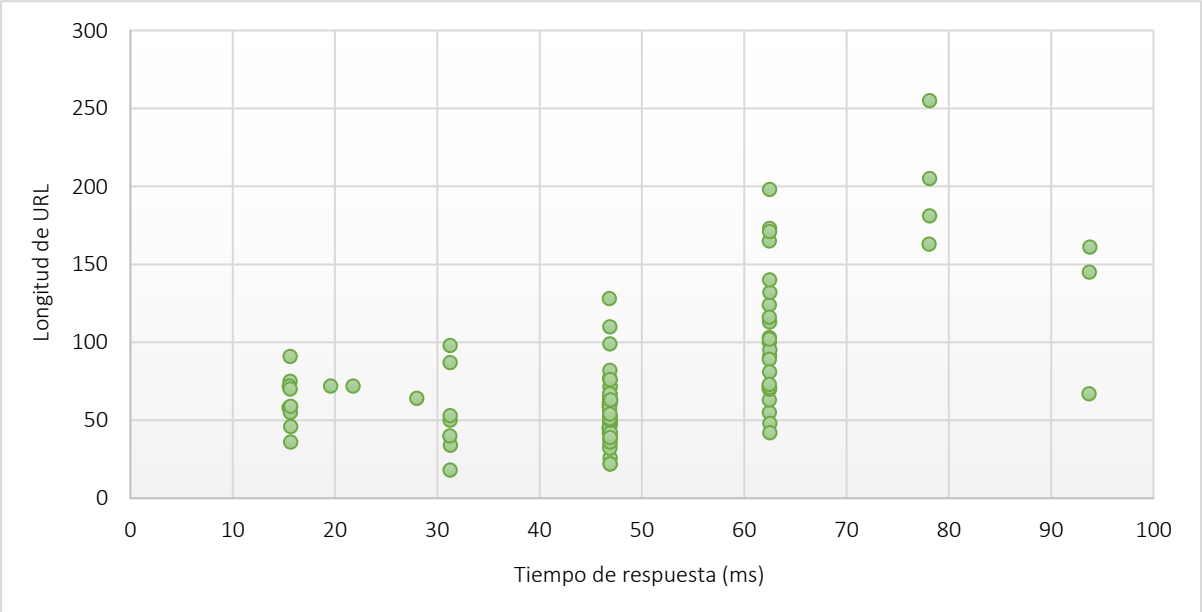
En cuanto a los falsos positivos, no se detectaron en el conjunto de datos utilizado para este análisis. Todos los códigos QR etiquetados como seguros fueron clasificados correctamente como seguros o como inciertos. Esto resalta la efectividad del sistema en la prevención de la clasificación incorrecta de enlaces seguros como maliciosos. Así mismo, se observó que el sistema logró una precisión del 100 % y una sensibilidad del 95 %. Sin embargo, es importante destacar que tanto la precisión como la sensibilidad se calcularon sin considerar los códigos QR clasificados como inciertos. Este detalle puede influir en la interpretación de la efectividad real de las métricas, ya que los códigos inciertos representan una parte significativa de los datos analizados, y su inclusión podría reducir las tasas de precisión y sensibilidad.

Simultáneamente, el sistema presentó una tasa de error del 14 %, que refleja la proporción de clasificaciones incorrectas sobre el total de análisis realizados. Este valor es relativamente bajo, lo que indica que el sistema tiene un buen rendimiento en la clasificación de enlaces, aunque el hecho de no contar con información suficiente para clasificar algunos enlaces como seguros o maliciosos contribuye a este error.

En relación al tiempo de respuesta promedio (TRP), la Figura 6 ilustra la relación entre el tiempo de respuesta del sistema y la longitud de las URLs, destacando la tendencia de que las URLs más largas tienden a generar tiempos de respuesta más altos. Para el análisis el TRP de un código QR fue de 48,95 ms, lo que muestra que el sistema tiene un rendimiento eficiente en términos de velocidad de análisis. El mayor tiempo de respuesta registrado fue de 9338 ms, mientras que el menor fue de 15,51 ms. Se observó que las URLs con mayor longitud tendían a requerir un mayor tiempo de análisis. Esto sugiere que el sistema puede verse afectado por la complejidad de las URLs al momento de realizar el análisis de seguridad.

Figura 6

Gráfico de dispersión entre Tiempo de Respuesta y Longitud de las URLs



4. Discusión

Los resultados son destacables, ya que indican que el sistema es capaz de detectar y clasificar correctamente los códigos QR maliciosos sin generar falsos positivos, y con un número mínimo de falsos negativos. Este desempeño es comparable con otros enfoques actuales en el campo de la detección de amenazas cibernéticas. Por ejemplo, Céspedes (2021) desarrolló un modelo de aprendizaje automático para la detección de URLs maliciosas con una precisión del 93 %. Aunque la precisión del sistema es superior, su enfoque demuestra que las soluciones basadas en machine learning son eficaces en la identificación de amenazas. Sin embargo, la principal diferencia entre ambos enfoques es que el presente sistema incorpora la API de VirusTotal, lo que mejora significativamente las métricas obtenidas al contar con una herramienta de validación adicional que ayuda a reducir los falsos positivos. De manera similar, el estudio de Hernández y Baluja (2021) sobre mecanismos para enfrentar ataques de phishing en redes reveló que las herramientas tradicionales como los antivirus logran tasas de detección cercanas al 90 %. A pesar de que el sistema propuesto en este estudio no utiliza tecnologías tradicionales basadas en firmas estáticas, se observa una mejora en la sensibilidad, que alcanza un 95 %. Esto puede explicarse por la combinación del modelo de aprendizaje automático con la validación en tiempo real que ofrece VirusTotal, lo que permite un análisis más preciso y eficiente. Sin embargo, como se observó, la presencia de códigos clasificados como inciertos sigue siendo un

desafío, especialmente cuando la información disponible en las bases de datos consultadas es insuficiente.

Por otra parte, Pawar et al. (2022), también evidenció alta precisión en la detección de enlaces maliciosos. Aunque el sistema no incluye un mecanismo de autenticación explícito, la integración de VirusTotal, como sugieren Rama y Praveen (2022), mejora significativamente la validación de URLs, proporcionando una capa adicional de seguridad al analizar las URLs en tiempo real, similar a lo que Wahsheh y Luccio (2020) destacan en su trabajo. Sin embargo, la dependencia de una API externa presenta limitaciones, ya que algunos códigos QR fueron clasificados como inciertos debido a la falta de información en las bases de datos de VirusTotal, lo que coincide con la preocupación de Geisler y Pöhn (2024) sobre la falta de controles adecuados en lugares públicos. A pesar de estas limitaciones, el sistema puede mitigar riesgos, especialmente en entornos educativos, donde los códigos QR, como señala Fernández-Rivas et al. (2022), mejoran la experiencia de aprendizaje, pero también presentan riesgos de seguridad.

En comparación con otros enfoques en la detección de phishing, como el de Cabezas y Gutierrez (2023), que utilizaron procesamiento de lenguaje natural para detectar URLs maliciosas en correos electrónicos, el sistema propuesto en este estudio presenta ventajas claras. Mientras que su enfoque alcanzó una precisión del 87 %, el sistema aquí descrito alcanzó un 100 % de precisión y, además, mantiene un tiempo de respuesta promedio de 48,95 ms, lo que lo hace particularmente útil en aplicaciones en tiempo real. Esto refuerza la hipótesis de que la combinación de aprendizaje automático con herramientas externas como VirusTotal mejora considerablemente el rendimiento del sistema. El análisis de desempeño del sistema demuestra que supera a las soluciones tradicionales basadas en firmas estáticas, como los antivirus mencionados por Hernández y Baluja (2021), que a menudo luchan con la detección de amenazas emergentes o URLs ofuscadas. La integración de la API de VirusTotal permite validar los enlaces analizados con múltiples motores antivirus, lo que contribuye a una mejor clasificación de los códigos QR. Además, la baja latencia observada, con tiempos de respuesta promedio de 48,95 ms, es adecuada para el análisis en tiempo real, lo cual es un factor importante en el desarrollo de sistemas de detección de amenazas dinámicas.

No obstante, es importante reconocer que la clasificación de algunos códigos como inciertos pone de manifiesto que la efectividad del sistema está parcialmente condicionada por la disponibilidad de datos en VirusTotal. La falta de información sobre URLs nuevas o poco frecuentes genera esta incertidumbre, lo cual limita la capacidad de clasificación del sistema en estos casos. Aunque el tiempo de respuesta promedio es bajo, la dependencia de una API externa puede introducir retrasos adicionales en condiciones de alta demanda, lo que podría afectar el desempeño en situaciones de alta carga. Para abordar estas limitaciones y mejorar aún más el sistema, se proponen varias optimizaciones. Primero, la integración de bases de datos adicionales, como PhishTank o Google Safe Browsing, así como bases de datos locales actualizadas dinámicamente, podría reducir la cantidad de códigos clasificados como inciertos. La implementación de análisis heurísticos también podría ser útil para detectar patrones sospechosos en URLs maliciosas sin depender exclusivamente de bases de datos preexistentes. Además, se podría mejorar la eficiencia del sistema mediante la optimización de los tiempos de respuesta, implementando un preanálisis local antes de consultar la API externa, lo que reduciría el tiempo de análisis en condiciones de alta demanda. Finalmente, la actualización continua del modelo mediante un sistema de retroalimentación que incorpore enlaces

maliciosos detectados al conjunto de entrenamiento podría mejorar la adaptabilidad del sistema frente a nuevas amenazas.

5. Conclusiones

El sistema antimalware propuesto demostró alta precisión (100 %) y sensibilidad (95 %) para identificar amenazas en códigos QR, proporcionando una capa adicional de seguridad frente a ataques de *quishing*. La baja tasa de error (14 %) observada y el tiempo de respuesta promedio de 48,95 ms refuerzan su potencial. Asimismo, su integración con la API de VirusTotal confirmó su eficacia en la detección y prevención de accesos a enlaces maliciosos, destacándose como una solución práctica para entornos cotidianos como dispositivos móviles y sistemas de pago. Además, su potencial de expansión hacia otros tipos de códigos, como Data Matrix y Aztec, sugiere una aplicabilidad más amplia en sectores como la logística y el control de acceso, garantizando adaptabilidad frente a nuevas tecnologías de codificación visual.

6. Referencias Bibliográficas

Amoah, G. A., y Hayfron-Acquah. (2022). QR code security: Mitigating the issue of quishing (QR code phishing). *International journal of computer applications*, *184*(33), 34–39. https://doi.org/10.5120/ijca2022922425

Cabezas, A., y Gutierrez, J. Z. R. (2023). *Detección de phishing en correos electrónicos mediante procesamiento de lenguaje natural del contenido y URLs ofuscadas* [Tesis de pregrado, Universidad de Lima]. https://repositorio.ulima.edu.pe/handle/20.500.12724/20761

Céspedes, M. M. (2021). *Detección de URLs maliciosas por medio de técnicas de aprendizaje automático* [Tesis de maestría, Universidad Nacional de Colombia]. https://repositorio.unal.edu.co/handle/unal/79722

Geisler, M., y Pöhn, D. (2024). Hooked: A real-world study on QR code phishing. *arXiv [cs.CR]*. https://doi.org/10.48550/ARXIV.2407.16230

Hossain, M. S., Zhou, X., y Rahman, M. F. (2018). Examining the impact of QR codes on purchase intention and customer satisfaction on the basis of perceived flow. *International Journal of Engineering Business Management*, *10*. https://doi.org/10.1177/1847979018812323

Hernández, A., y Baluja, W. (2021). Principales mecanismos para el enfrentamiento al phishing en las redes de datos. *Revista Cubana de Ciencias Informáticas*, *15*(4), 413–441. https://www.redalyc.org/articulo.oa?id=378370462024

Japan Patent Office. (2022, 4 de julio). *QR Code (Denso Wave Incorporated).* https://goo.su/z1UTg

Krombholz, K., Frühwirt, P., Kieseberg, P., Kapsalis, I., Huber, M., y Weippl, E. (2014). QR code security: A survey of attacks and challenges for usable security [conferencia]. *International Conference on Human Aspects of Information Security, Privacy, and Trust*, Viena, Austria. https://doi.org/10.1007/978-3-319-07620-1_8

Pawar, A., Fatnani, C., Sonavane, R., Waghmare, R., y Saoji, S. (2022). Secure QR code scanner to detect malicious URL using machine learning [conferencia]. *2022 2nd Asian*

Conference on Innovation in Technology (ASIANCON), Ravet, India. https://doi.org/10.1109/ASIANCON55314.2022.9908759

Petrova, K., Romaniello, A., Medlin, B. D., y Vannoy, S. A. (2016). QR Codes Advantages and Dangers [conferencia]. *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications*, Lisboa, Portugal. https://doi.org/10.5220/0005993101120115

Rama, M., y Praveen, K. (2022). Prevention of phishing attacks using QR code safe authentication. En S. Smys, V. E. Balas, R. Palanisamy. (Eds) *Inventive Computation and Information Technologies* (pp. 361–372). Springer Nature Singapore. https://doi.org/10.1007/978-981-16-6723-7_27

Saeed, S., Altamimi, S. A., Alkayyal, N. A., Alshehri, E., y Alabbad, D. A. (2023). Digital transformation and cybersecurity challenges for businesses resilience: Issues and recommendations. *Sensors (Basel, Switzerland)*, *23*(15), 6666. https://doi.org/10.3390/s23156666

Slashnext. (2023). *The State of Phishing 2023: An in-depth look at cybersecurity threat trends with insights into how cybercriminals are swiftly advancing and what is required to stop them*. https://goo.su/scbR6Jw

Sahingoz, O. K. S., Buber, E. B., y Kugu, E. K. (2023). *Phishing Attack Dataset* [Data set]. IEEE DataPort. https://dx.doi.org/10.21227/4098-8c60

Sarkhi, M., y Mishra, S. (2024). Detection of QR code-based cyberattacks using a lightweight Deep Learning model. *Engineering technology & Applied science research*, *14*(4), 15209–15216. https://doi.org/10.48084/etasr.7777

Sharevski, F., Devine, A., Pieroni, E., y Jachim, P. (2022). Gone quishing: A field study of phishing with malicious QR codes. En *arXiv [cs.CR]*. https://doi.org/10.48550/ARXIV.2204.04086

Spinelli, O. M., Corrons, F. J., y Dreizzen, E. (2021). QR codes in medical education - part 2. An analog - digital technology cooperation. *Anales. Universidad Nacional de Asuncion. Facultad de Ciencias Medicas*, *54*(3), 129–142. https://doi.org/10.18004/anales/2021.054.03.129

Spinelli, O. M., y Dreizzen, E. (2021). QR codes in medical education - part 1 an analog - digital bridge. *Anales. Universidad Nacional de Asuncion. Facultad de Ciencias Medicas*, *54*(2), 111–120. https://doi.org/10.18004/anales/2021.054.02.111

Tiwari, S. (2016). An introduction to QR code technology [conferencia]. *2016 International Conference on Information Technology (ICIT)*, Bhubaneswar, India. https://doi.org/10.1109/ICIT.2016.021

Wahsheh, H. A. M., y Luccio, F. L. (2020). Security and privacy of QR code applications: A comprehensive study, general guidelines and solutions. *Information (Basel)*, *11*(4), 217. https://doi.org/10.3390/info11040217

Yong, K. S. C., Chiew, K. L., y Tan, C. L. (2019). A survey of the QR code phishing: the current attacks and countermeasures [conferencia]. *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, Sarawak, Malaysia. https://doi.org/10.1109/ICSCC.2019.8843688