Pontificia Universidad Católica del Ecuador

Sede Ibarra

ESCUELA DE INGENIERÍA

INFORME FINAL DEL PROYECTO

TEMA:

ANÁLISIS DE VULNERABILIDADES DE LA APLICACIÓN QUIPUX MEDIANTE LA HERRAMIENTA OWASP ZAP APLICADO AL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE LA PROVINCIA DEL CARCHI

PREVIO A LA OBTENCIÓN DEL TÍTULO DE

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

LÍNEAS DE INVESTIGACIÓN:

CULTURA ORGANIZACIONAL E INFORMÁTICA

AUTOR/A: LUIS KEVIN CANACUÁN PADILLA

ASESOR/A: GALO HERNÁN PUETATE

IBARRA, AGOSTO 2023

Ibarra, 15 de agosto de 2023

Magister

Galo Hernán Puetate Huera

ASESOR

CERTIFICACIÓN

Haber revisado el presente informe final de investigación, el mismo que se ajusta a las normas vigentes en la Escuela de Ingeniería de la Pontificia Universidad Católica del Ecuador Sede Ibarra (PUCESI); en consecuencia, autorizo su presentación para los fines legales pertinentes.

(f:) 

Msc. Galo Hernán Puetate Huera

C.C.: 0401375787

PÁGINA DE APROBACIÓN DEL TRIBUNAL

El jurado examinador, aprueba el presente informe de investigación en nombre de la Pontificia Universidad Católica del Ecuador Sede Ibarra (PUCESI):

(f:) 

Mgs. Galo Hernán Puetate Huera

C.C.: 0401375787

(f): 

Mgs. Darwin Marcelo Pillo

C.C.: 1003319660

(f): 

Mgs. Diego Raúl Mafla

C.C.: 1001698644

ACTA DE CESIÓN DE DERECHOS

Yo, Kevin Canacuán, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: “Se reconoce facultad de los autores y demás titulares de derechos de disponer de sus derechos o autorizar las utilizaciones de sus obras o prestaciones, a título gratuito u oneroso, según las condiciones que determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia”.

Ibarra, 15 de agosto de 2023

f): 

Kevin Canacuán

C.I.: 0450057633

AUTORÍA

Yo, Kevin Canacuán, portador de la cédula de identidad N°0450057633, declaro que la presente investigación es de total responsabilidad de la autora, y eximo expresamente a la Pontificia Universidad Católica del Ecuador Sede Ibarra de posibles reclamos o acciones legales.

f): 

Kevin Canacuán

C.I.: 0450057633

DECLARACIÓN Y AUTORIZACIÓN

Yo: Kevin Canacuán, con CI: 0450057633, autor del trabajo de grado intitulado: "ANÁLISIS DE VULNERABILIDADES DE LA APLICACIÓN QUIPUX MEDIANTE LA HERRAMIENTA OWASP ZAP APLICADO AL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE LA PROVINCIA DEL CARCHI", previo a la obtención del título profesional de "Ingeniero en Tecnologías de la Información", en la Escuela de Ingeniería.

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador Sede- Ibarra, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador Sede Ibarra a difundir a través del Repositorio Digital de la PUCESI el referido trabajo de graduación, respetando las políticas de propiedad intelectual de la Universidad.

Ibarra, 15 de agosto del 2023

(f.) 

Kevin Canacuán

C.I. 1712277522

DEDICATORIA

Dedico mi tesis a mis padres José y Mary, por acompañarme en cada paso que doy en la búsqueda de ser mejor persona y profesional.

También se la dedico a mi hermano Alexis, por todo su apoyo incondicional, espero que le sirva de ejemplo de que todo se puede lograr.

A mi amigo Josué T. por ser un gran apoyo durante toda la carrera y al resto de mi familia, amigos que están al pendiente de mis progresos en la vida.

AGRADECIMIENTO

Este trabajo está dedicado a la Pontificia Universidad Católica del Ecuador Sede Ibarra por mostrarme el camino hacia mi propósito en la vida. A mi tutor y a todos los docentes de la escuela de Ingeniería por compartir sus conocimientos y experiencias durante todo el trayecto guiándome a culminar con éxito una meta más.

RESUMEN

La seguridad de la información busca salvaguardar la integridad de los datos, prevenir accesos no autorizados a sistemas y evitar la divulgación o compromiso de los mismos. En este proyecto de investigación se busca la mitigación de vulnerabilidades que existen en una de las aplicaciones internas que se encuentran en ejecución en el Gobierno Autónomo Descentralizado de la Provincia del Carchi (GADPC). La metodología adoptada abarca diversas etapas. En primer lugar, se analiza la gestión de la seguridad en el entorno de esta entidad gubernamental para comprender su enfoque y necesidades. Posteriormente, se realiza un análisis exhaustivo del código fuente de Quipux, empleando las técnicas y estándares establecidos por OWASP. La síntesis de resultados se basa en la revisión de los informes de escaneo, los cuales detallan las vulnerabilidades y errores identificados en la aplicación de gestión. A través de esta evaluación, se logra determinar que la ejecución de un piloto utilizando la herramienta ZAP de OWASP que permite una identificación precisa de las vulnerabilidades presentes en Quipux, de acuerdo con los niveles críticos de seguridad que imperan en la aplicación web sometida a análisis. En este sentido, el enfoque propuesto revela áreas de mejora en términos de seguridad y ofrece recomendaciones concretas para mitigar las vulnerabilidades descubiertas en la aplicación utilizada por el GADPC.

Palabras Clave: OWASP ZAP, vulnerabilidades, Quipux, Scanning Report

ABSTRACT

Information security seeks to safeguard the integrity of data, prevent unauthorized access to systems and avoid disclosure or compromise. This research project seeks to mitigate vulnerabilities that exist in one of the internal applications that are being implemented in the Decentralized Autonomous Government of the Province of Carchi (GADPC). The methodology adopted includes several stages. First, the security management in the environment of this governmental entity is analyzed to understand its approach and needs. Subsequently, an exhaustive analysis of the Quipux source code is carried out, using the techniques and standards established by OWASP. The synthesis of results is based on the review of the scan reports, which detail the vulnerabilities and errors identified in the management application. Through this evaluation, it is possible to determine that the execution of a pilot using the OWASP ZAP tool allows an accurate identification of the vulnerabilities present in Quipux, according to the critical security levels prevailing in the web application under analysis. In this sense, the proposed approach reveals areas for improvement in terms of security and offers concrete recommendations to mitigate the vulnerabilities discovered in the application used by GADPC.

Keywords: *OWASP ZAP, vulnerabilities, Quipux, Scanning Report*

Tabla de contenido

Índice de Tablas

Índice de Figuras

INTRODUCCIÓN

La creciente digitalización y automatización de los procesos en las organizaciones ha llevado a una mayor dependencia de los sistemas informáticos y, por tanto, un mayor riesgo de vulnerabilidades y ataques cibernéticos.

Según McLean (2015), "las vulnerabilidades de seguridad son debilidades en los sistemas informáticos que pueden ser explotadas por los atacantes para comprometer la integridad, confidencialidad o disponibilidad de los datos". En este sentido, el análisis de vulnerabilidades es una técnica importante para detectar y mitigar estas debilidades. Por su parte, la Fundación *Open Web Application Security Project* (OWASP) (2020) destaca la importancia de las herramientas de análisis de vulnerabilidades en la seguridad de las aplicaciones web. OWASP ZAP (*Open Web Application Security Project Zed Attack Proxy)* es una herramienta de código abierto que permite identificar y explotar vulnerabilidades en aplicaciones web.

En este contexto, la seguridad de la información se ha convertido en un tema prioritario en el ámbito empresarial y gubernamental. El Gobierno Autónomo Descentralizado de la Provincia del Carchi (GADPC) en Ecuador utiliza la aplicación Quipux para gestionar diversos procesos administrativos y operativos. Es por ello que, se considera importante el análisis de vulnerabilidades en dicha aplicación, utilizando la herramienta OWASP ZAP.

Este proyecto de investigación busca la mitigación de vulnerabilidades que existen en una de las aplicaciones internas que se encuentran en ejecución en el GADPC.

Para realizar este estudio, se estableció como objetivo general:

- Analizar las vulnerabilidades de la aplicación Quipux mediante pruebas de penetración utilizando la herramienta OWASP ZAP para la Prefectura del Carchi.

De la misma manera, se fundamentaron los objetivos específicos que permitirán alcanzar el objetivo principal. Estos son:

- Analizar la literatura científica mediante la revisión bibliográfica del análisis de vulnerabilidades de la aplicación Quipux de la Prefectura del Carchi.
- Determinar el tipo de prueba de penetración de la seguridad para evaluar las vulnerabilidades de la aplicación Quipux
- Implementar los mecanismos de seguridad para las vulnerabilidades identificadas.
- Generar reporte del análisis obtenido por la herramienta OWASP ZAP en la aplicación Quipux para la interpretación de los resultados.

Este estudio consta de (3) capítulos: El primero describe las investigaciones bibliográficas y conceptos que sirven como antecedentes para la investigación. El segundo capítulo consta de las herramientas tecnológicas y el desarrollo del proyecto, utilizando la metodología OWASP para el análisis de vulnerabilidades. El tercer capítulo expone los resultados del *pentest* realizado y la propuesta generada para el GADPC. Finalizando con las conclusiones y recomendaciones

CAPÍTULO I: ESTADO DEL ARTE

En este capítulo, se presenta el estado del arte y los conceptos básicos relacionados con el análisis de vulnerabilidades en aplicaciones web y la herramienta OWASP ZAP. Se revisan las tendencias actuales en materia de seguridad informática, las diferentes técnicas de ataques informáticos y las principales vulnerabilidades a las que se enfrentan las aplicaciones web. Además, se aborda la metodología para la identificación de vulnerabilidades y el uso de herramientas de escaneo automatizado, destacando la importancia de OWASP ZAP como una herramienta de código abierto ampliamente utilizada para el análisis de vulnerabilidades en aplicaciones web. Se presentan los conceptos básicos de OWASP ZAP, su arquitectura y su funcionamiento, así como las principales características y herramientas que ofrece para la identificación y mitigación de vulnerabilidades en aplicaciones web. Todo lo anterior, fundamenta las bases para el análisis de vulnerabilidades en la aplicación Quipux, utilizada por el GADPC.

1.1 Seguridad en aplicaciones web:

La seguridad en aplicaciones web es una disciplina crucial para proteger los sistemas en línea contra amenazas y ataques cibernéticos. Implica la implementación de medidas y prácticas para prevenir vulnerabilidades, autenticación deficiente, inyecciones SQL, ataques de denegación de servicio (DoS), entre otros. Su objetivo es salvaguardar la integridad, confidencialidad y disponibilidad de los datos.

1.1.1 Metodologías de pruebas de seguridad:

Las metodologías de pruebas de seguridad son enfoques estructurados para evaluar la robustez de una aplicación web. A continuación, se mencionó los más importantes a destacar:

- OWASP *Testing Guide*: Esta guía de pruebas de seguridad desarrollada por OWASP es ampliamente reconocida y utilizada en la comunidad de seguridad informática. Ofrece una estructura completa para realizar pruebas exhaustivas en aplicaciones web. (*About the OWASP Foundation | OWASP Foundation*, 2023)

- PTES (*Penetration Testing Execution Standard*): PTES es una metodología ampliamente aceptada en el campo de las pruebas de penetración. Proporciona una estructura detallada para llevar a cabo evaluaciones de seguridad de manera efectiva.

- NIST SP 800-115: Publicado por el *National Institute of Standards and Technology* (NIST), este documento ofrece pautas sólidas para la realización de pruebas de seguridad en sistemas de información.

- OSSTMM (*Open Source Security Testing Methodology Manual*): Con un enfoque en la seguridad integral, el OSSTMM es utilizado por profesionales de seguridad para evaluar tanto sistemas de tecnología como de personas y procesos.

- *Crest Core Process*: Especialmente relevante en el ámbito de las pruebas de penetración, esta metodología es favorecida por muchos profesionales y organizaciones de seguridad.

- *Web Application Security Consortium Web Security Testing Guide*: Esta guía se enfoca en la seguridad de aplicaciones web y proporciona directrices claras para la evaluación de vulnerabilidades en este contexto. (*Metodologías Para La Auditoria de La Seguridad*, 2017)

1.1.2 Vulnerabilidades comunes en aplicaciones web:

Inyecciones (SQL, código, OS, etc.): Ataques que permiten a los atacantes insertar comandos maliciosos en entradas de datos para ejecución no autorizada.

Cross-Site Scripting (XSS): Inserción de scripts maliciosos en páginas web visitadas por otros usuarios, lo que puede robar información o ejecutar acciones no deseadas.

Cross-Site Request Forgery (CSRF): Manipulación de sesiones de usuarios legítimos para realizar acciones no autorizadas en su nombre.

Autenticación y autorización defectuosa: Fallos en los sistemas de autenticación y autorización que permiten a los atacantes acceder a recursos sin permiso.

Exposición de datos sensibles: Divulgación no autorizada de información confidencial debido a configuraciones incorrectas o acceso indebido.

Desbordamientos de búfer: Sobrecarga de áreas de memoria para ejecución de código malicioso.

Falta de validación y filtrado de datos: Entrada no validada que permite la ejecución de comandos maliciosos o la manipulación de datos.

Configuraciones incorrectas de seguridad: Ajustes inadecuados que dejan sistemas vulnerables a ataques.

Denegación de servicio (DoS): Sobrecarga de sistemas o servicios para bloquear el acceso legítimo.

Exposición a través de APIs: Vulnerabilidades en interfaces de programación de aplicaciones que permiten a los atacantes acceder a recursos internos.

1.1.3 Proceso de análisis de vulnerabilidades:

El proceso involucra la identificación, evaluación y mitigación de vulnerabilidades. Comienza con el descubrimiento de activos, pruebas de seguridad, análisis de resultados y finaliza con la aplicación de parches y mejoras.

- **Identificación de activos y alcance:** Definir qué componentes serán analizados y establecer los límites del alcance de las pruebas.
- **Recolección de información:** Obtener detalles relevantes sobre la aplicación, como su arquitectura, tecnologías utilizadas y puntos de entrada potenciales.
- **Identificación de vulnerabilidades:** Realizar pruebas exhaustivas para descubrir vulnerabilidades comunes y específicas en la aplicación.
- **Evaluación de riesgos:** Evaluar y clasificar las vulnerabilidades según su gravedad y el potencial impacto en la seguridad.
- **Recomendaciones de mitigación:** Proporcionar soluciones y medidas para corregir o mitigar las vulnerabilidades identificadas.
- **Generación de informes:** Crear un informe detallado que incluya una descripción de las vulnerabilidades, sus riesgos y las recomendaciones para su solución.
- **Implementación de medidas correctivas:** Aplicar las soluciones recomendadas para abordar las vulnerabilidades y mejorar la seguridad.
- **Pruebas de validación:** Realizar pruebas adicionales para asegurarse de que las vulnerabilidades hayan sido corregidas de manera efectiva.

1.1.4 Gestión de vulnerabilidades:

La gestión de vulnerabilidades abarca la detección, evaluación, priorización y mitigación de vulnerabilidades. Se enfoca en administrar riesgos y asegurar que los problemas sean tratados eficazmente

1. **Detección e identificación:**
    - Utilizar herramientas de seguridad y escaneo para identificar posibles vulnerabilidades en sistemas y aplicaciones.

- Monitorear fuentes de información de seguridad, como bases de datos de vulnerabilidades y boletines de seguridad.

2. **Clasificación y priorización:**

    - Evaluar la gravedad y el impacto de las vulnerabilidades identificadas para determinar cuáles requieren atención inmediata.
    - Utilizar sistemas de puntuación, como CVSS (*Common Vulnerability Scoring System*), para asignar prioridades.

3. **Asignación de responsabilidades:**

    - Designar equipos o individuos responsables de abordar y mitigar las vulnerabilidades identificadas.
    - Establecer flujos de trabajo claros para la comunicación y resolución de problemas.

4. **Mitigación y solución:**

    - Implementar soluciones y contramedidas para corregir las vulnerabilidades. Esto puede incluir parches, cambios en la configuración o mejoras en el código.
    - Realizar pruebas de validación para asegurarse de que las soluciones implementadas sean efectivas.

1.2 Estándares y buenas prácticas de seguridad:

Cumplir con estándares como *OWASP Top Ten*, *PCI DSS* (*Payment Card Industry Data Security Standard*) y *CIS Benchmarks* garantiza un nivel mínimo de seguridad. Buenas prácticas incluyen el principio de menor privilegio, separación de responsabilidades y parcheo regular.

1.2.1 Automatización de pruebas de seguridad:

La automatización mediante herramientas como SonarQube y OWASP ZAP agiliza la detección y evaluación de vulnerabilidades. Estas herramientas escanean código y aplicaciones, identificando problemas de seguridad.

1.2.2 Herramientas de análisis de seguridad

SonarQube

Es una plataforma de código abierto diseñada para evaluar y mejorar la calidad del código de software. Aunque su enfoque principal es la calidad del código, también incluye capacidades de análisis de seguridad para identificar vulnerabilidades y problemas de seguridad en el código fuente. SonarQube se integra con diferentes lenguajes de programación y tecnologías, lo que lo hace adecuado para diversas aplicaciones y proyectos.

Características clave:

- Realiza análisis estáticos del código fuente para detectar vulnerabilidades, errores y problemas de calidad.
- Proporciona métricas y estadísticas sobre la calidad del código y la seguridad.
- Integra *plugins* y reglas específicas para lenguajes de programación populares.
- Ofrece informes detallados y tableros de control para visualizar los resultados de las pruebas.
- Facilita la detección temprana de problemas de seguridad y ayuda en la toma de decisiones informadas para mejorar la calidad del código.

OWASP ZAP (Zed Attack Proxy):

OWASP ZAP es una herramienta de seguridad de aplicaciones web de código abierto diseñada para realizar pruebas de seguridad automatizadas y manuales en aplicaciones web. ZAP ofrece un conjunto de características robustas para identificar y mitigar vulnerabilidades en aplicaciones web, y es ampliamente utilizado en la comunidad de seguridad.

Características clave:

- Realiza escaneos automáticos de aplicaciones web para identificar vulnerabilidades comunes, como inyecciones, XSS, CSRF, entre otras.
- Permite la configuración de políticas de seguridad personalizadas y escenarios de ataque.
- Facilita pruebas manuales y exploratorias de aplicaciones web para descubrir vulnerabilidades más complejas.
- Proporciona informes detallados y registros de actividad para documentar las pruebas realizadas.
- Integra con otras herramientas y procesos de seguridad, lo que facilita su inclusión en flujos de trabajo de desarrollo.

Tanto SonarQube como OWASP ZAP fueron herramientas valiosas para la seguridad de aplicaciones web y el análisis de código. Sin embargo, es importante señalar que SonarQube se centra en la calidad del código en general, mientras que OWASP ZAP está específicamente diseñado para pruebas de seguridad de aplicaciones web. Ambas herramientas pueden ser utilizadas de manera complementaria en el proceso de desarrollo y evaluación de aplicaciones web seguras.

1.3 Arquitectura y características de Quipux:

Quipux es una aplicación o sistema utilizado por el GADPC para gestión documental y optimización de diversas funciones administrativas y operativas. Este sistema está desarrollado con una arquitectura de tres capas, básicamente divide la aplicación en presentación, lógica de negocio y capa de datos, lo que permite la escalabilidad y el mantenimiento por separado

Entre sus principales características tenemos que es un software libre, propende al cero gasto de papeles e insumos, eficiencia en el gasto público, uso de firma electrónica, acercamiento ciudadano, gobierno eficaz y eficiente, incremento de la productividad en los servidores públicos al ahorrar tiempos de tratamiento de documentación física y lo más importante es que está desarrollado bajo estándares de la Norma Técnica Ecuatoriana de Gestión de Documentos NTE INEN 2410. (*¿Qué Es Quipux? – Sistema de Gestión Documental Quipux*, 2023)

1.4 Análisis de estudios previos sobre seguridad de aplicaciones web y vulnerabilidades:

El análisis de vulnerabilidades en aplicaciones web es una práctica crucial para garantizar la seguridad de la información en organizaciones gubernamentales y empresariales. Uno de los principales desafíos en este campo es el descubrimiento y la eliminación de vulnerabilidades en aplicaciones web de manera oportuna y efectiva. En este sentido, OWASP ZAP se ha convertido en una herramienta popular para realizar pruebas de seguridad en aplicaciones web debido a su capacidad para detectar vulnerabilidades de seguridad en tiempo real y su amplia gama de funcionalidades.

En la PUCESA (Pontifica Universidad Católica del Ecuador Sede Ambato) realizó una investigación sobre el análisis de vulnerabilidades donde se utilizó OWASP para obtener diversos enfoques para la distinción de estas como también emplea herramientas que facilitan la realización de pruebas de penetración en aplicaciones web. Esta metodología también ofrece métodos para la resolución y mitigación de dichas vulnerabilidades. (Leonardo, 2021)

Villanueva (2021) brinda la importancia del análisis de vulnerabilidades en las empresas:

- Minimizar el impacto de los incidentes de seguridad;
- Mejorar la continuidad operativa del negocio;
- Optimizar la asignación de recursos en infraestructuras;

El resultado clave del análisis de vulnerabilidades es un informe que proporciona detalles esenciales sobre todas las vulnerabilidades identificadas, junto con su evaluación de riesgos correspondiente.

- Facilitar la conformidad con los estándares de cumplimiento

Un estudio de Li et al. (2020) evaluó la efectividad de OWASP ZAP en la detección de vulnerabilidades en aplicaciones web y encontró que la herramienta es capaz de detectar una amplia gama de vulnerabilidades, incluyendo inyección SQL, XSS y vulnerabilidades de autenticación.

Además, un estudio de Kaur y Gupta (2021) destacó la importancia de la integración de herramientas de pruebas de seguridad automatizadas como OWASP ZAP en los procesos de desarrollo de aplicaciones web para garantizar la solidez de la información y reducir los costos asociados con la identificación y corrección de vulnerabilidades.

CAPÍTULO II: MATERIALES Y MÉTODOS

En este capítulo, se definió el procedimiento para examinar las vulnerabilidades presentes en la aplicación Quipux, un sistema utilizado por el gobierno provincial para la gestión de diversos procesos administrativos. La herramienta OWASP ZAP fue empleada como una herramienta de seguridad informática para identificar y evaluar los posibles puntos débiles de la aplicación, con el objetivo de brindar recomendaciones y medidas correctivas para fortalecer la seguridad del sistema y proteger la información sensible del GADPC.

2.1 Generalidades de la investigación

Este trabajo mantuvo una relación con la seguridad informática y se centró en el análisis de vulnerabilidades de la aplicación Quipux en el contexto del GADPC. El objetivo principal fue identificar posibles puntos débiles en la aplicación y brindar recomendaciones para fortalecer la seguridad del sistema. Para lograr esto, se utilizó la herramienta OWASP ZAP, que es una herramienta reconocida en la industria de la seguridad informática para realizar análisis de vulnerabilidades en aplicaciones web.

La investigación es aplicada con un enfoque cualitativo ya que se centró en comprender y evaluar las vulnerabilidades de seguridad presentes en la aplicación y su impacto potencial. Este enfoque implicó considerar aspectos cualitativos relacionados con la naturaliza y características específicas de las vulnerabilidades identificadas, así como su contexto y posibles riesgos asociados.

A continuación, se mencionan algunos elementos del enfoque cualitativo para esta investigación:

1. Identificación y descripción de vulnerabilidades: Implica identificar y describir detalladamente las vulnerabilidades encontradas en la aplicación Quipux. Esto involucra documentar las debilidades de seguridad específicas, como posibles

puntos de acceso, deficiencias en el control de acceso, fallos en la validación de datos, entre otros.

2. Evaluación de la gravedad de las vulnerabilidades: Permite evaluar la gravedad de cada vulnerabilidad identificada. Esto permite considerar su impacto potencial en el GADPC, teniendo en cuenta la confidencialidad, integridad y disponibilidad de los datos y sistemas involucrados.

3. Análisis del contexto y riesgo asociado: Involucra comprender el contexto específico de la aplicación Quipux y su implementación en el GADCP. Esto incluye considerar la infraestructura tecnológica, las políticas de seguridad existentes y otros factores que puedan influir en la explotación de las vulnerabilidades y el riesgo asociado.

4. Propuestas de mitigación y recomendaciones: Basándose en el análisis cualitativo de las vulnerabilidades, se pueden proporcionar propuestas de mitigación y recomendaciones para abordar las debilidades identificadas. Estas propuestas pueden incluir acciones correctivas, mejores prácticas de seguridad, fortalecimiento de controles y medidas preventivas para minimizar el riesgo.

2.1.1 Alcance de la investigación

Esta investigación tuvo como objetivo evaluar las vulnerabilidades solamente del sistema Quipux en el GADPC. Por lo tanto, se realizó la respectiva propuesta para la mitigación de vulnerabilidades en esta aplicación más no la implementación.

2.2 Metodología

El análisis de vulnerabilidades se realizó siguiendo una metodología basada en las mejores prácticas establecidas por OWASP.

A continuación, cómo se observa en la *Ilustración 1*, se proporciona una descripción detallada del proceso metodológico a utilizar para analizar la seguridad del Quipux.

Figura 1: Metodología

Nota: este proceso fue diseñado por el propio autor basado en los lineamientos de Gamboa Y Safla.

Fase I.

En esta investigación se abordó el análisis de los desafíos relacionados con la seguridad informática y la gestión de la información en el GADPC. Se pretendía determinar un conjunto de estándares, herramientas y documentación, utilizando como base los datos e información disponibles.

Fase II.

En esta etapa se estableció el entorno para el experimento centrado en el análisis de seguridad del Quipux. Aquí se preparó cuidadosamente el entorno en que se llevó a cabo la ejecución del análisis, el cual fue sometido a un proceso exhaustivo de análisis de vulnerabilidades a través de rigurosas pruebas. El objetivo principal fue identificar y evaluar las vulnerabilidades de seguridad presentes en el software.

Fase III.

Se ha realizado un análisis exhaustivo del código fuente del Quipux desarrollado por el GADPC. Para llevar a cabo este proceso, se empleó la herramienta OWASP ZAP, que permitió identificar y evaluar posibles vulnerabilidades y amenazas conocidas (CVA). Además, se utilizaron métodos y estándares abiertos de tecnologías de la información, como el Sistema de Puntuación de Vulnerabilidades

Comunes (CVSS), que se aplicó mediante un enfoque estático de evaluación de aplicaciones web. (Portilla, 2020)

Fase IV.

Se proporcionó un marco de trabajo que se centra en el análisis de vulnerabilidades de seguridad en el producto de software Quipux. Se examinaron los resultados de los informes de escaneo de ZAP, específicamente el resumen de alertas que reveló las vulnerabilidades encontradas en el software de registro, control, flujo, organización y trazabilidad de los documentos digitales y/o físicos que se envían y reciben en el GADPC.

2.3 Procedimiento de análisis:

A continuación, se describen los pasos clave del proceso de análisis:

2.3.1 Configuración de OWASP ZAP:

Se realizó la configuración inicial de OWASP ZAP para adaptarlo a las características específicas de la aplicación Quipux y establecer los parámetros de análisis adecuados.

2.3.2 Identificación de las funcionalidades de la aplicación:

Se examinó en detalle la aplicación Quipux para comprender su arquitectura, las funcionalidades que ofrece y los datos que manipula.

2.3.3 Escaneo de seguridad automatizado:

Se realizó un escaneo automatizado de seguridad utilizando OWASP ZAP. Esta etapa permitió identificar de manera rápida y eficiente las vulnerabilidades más comunes, como inyección de SQL, cross-site scripting (XSS), entre otras.

2.3.4 Análisis manual de vulnerabilidades:

Se llevó a cabo un análisis manual exhaustivo para identificar vulnerabilidades específicas que podrían haber pasado desapercibidas en el escaneo automatizado. Esta etapa implica la revisión de código, la manipulación de datos y la interacción directa con la aplicación.

2.3.5 Registro de las vulnerabilidades encontradas:

Todas las vulnerabilidades identificadas durante el análisis, tanto en el escaneo automatizado como en el análisis manual, fueron registradas en un informe detallado. Se proporcionó información sobre la gravedad de cada vulnerabilidad, las posibles consecuencias y las recomendaciones para su mitigación.

2.3.6 Priorización de las vulnerabilidades:

Las vulnerabilidades identificadas fueron priorizadas en función de su gravedad y su impacto potencial en la seguridad de la aplicación Quipux. Esto permitió al Gobierno Autónomo Descentralizado de la Provincia del Carchi tomar medidas inmediatas para abordar las vulnerabilidades más críticas.

2.4 Desarrollo del análisis

Técnicamente se estableció el desarrollo para llevar a cabo el análisis de seguridad del software de registro y control documental. En este proceso, se describieron en detalle cada una de las etapas realizadas para analizar las vulnerabilidades del producto de software denominado Quipux.

2.4.1 Descripción

El diseño del experimento consistió en llevar a cabo una serie de análisis exhaustivos en el código fuente del sistema Quipux con el fin de identificar y catalogar un conjunto de vulnerabilidades de seguridad. El objetivo principal era obtener un conjunto de riesgos y amenazas a los que el software está expuesto, para luego clasificarlos y mitigarlos en base a los registros de vulnerabilidades y amenazas comunes (CWE). Además, se utilizaron estándares abiertos de tecnologías de información (CVSS) y se aplicó un método de evaluación estático de aplicaciones web.

Se estableció que el análisis se realizaría en un entorno de pruebas compuesto por un conjunto de servidores que pertenecían al sistema Quipux de GADPC.

El entorno de pruebas estaba compuesto por los siguientes servidores:

- Servidor de base de datos.
- Servidor de lógica de negocio.
- Servidor de aplicaciones.

El análisis del código fuente fue llevado a cabo específicamente en el servidor de lógica de negocio, el cual contiene los modelos, controladores y vistas que formaban parte integral del sistema Quipux.

2.4.2 Materiales y métodos de análisis

Dado que se trataba de un proyecto de desarrollo de software, a continuación, se describen los materiales y métodos utilizados para el análisis de las vulnerabilidades de seguridad:

Tabla 1: Materiales para el desarrollo del análisis

Fuente: Kevin Canacuán

- CVE (Common Vulnerabilities and Exposures): Se trata de un estándar que permite identificar de manera única cada una de las vulnerabilidades de seguridad que se encuentran registradas y disponibles públicamente.
- CVSS (Common Vulnerability Scoring System): Es un sistema que permite asignar una puntuación a las vulnerabilidades, proporcionando un método abierto y estandarizado para la clasificación de las mismas. El CVSS facilita la priorización de las vulnerabilidades y coordina una respuesta para su mitigación de manera eficiente.
- OWASP ZAP (Zed Attack Proxy) es una herramienta de código abierto perteneciente al proyecto OWASP, que se utiliza para escanear la seguridad de las aplicaciones web. Esta herramienta permite realizar pruebas de penetración mediante el análisis de amenazas y riesgos a nivel de la aplicación. También ofrece la capacidad de establecer niveles y umbrales de aceptación, identificar vulnerabilidades en el acceso y determinar una clasificación del nivel de riesgo e impacto asociado a estas vulnerabilidades en las aplicaciones web.

Se realizó un análisis de los principales riesgos de seguridad a los que puede estar expuesto el sistema Quipux del GADPC utilizando OWASP ZAP. Este análisis tiene como objetivo identificar y evaluar las posibles vulnerabilidades que podrían comprometer la seguridad del sistema, permitiendo así tomar medidas adecuadas para mitigar los riesgos y proteger la integridad, confidencialidad y disponibilidad de los datos y sistemas relacionados con registro y control de los documentos de la prefectura.

Tabla 2: Riesgos de seguridad a ser evaluados

Fuente: OWASP Top 10: 2021

La Tabla 2, se proporciona una descripción detallada de los riesgos de seguridad que serán evaluados utilizando la herramienta Zed Attack Proxy (ZAP).

Para el análisis de las vulnerabilidades del Sistema de registro y control documental se ejecutan las siguientes tareas.

1. Ejecución de escaneo de vulnerabilidades
2. Informe de vulnerabilidades

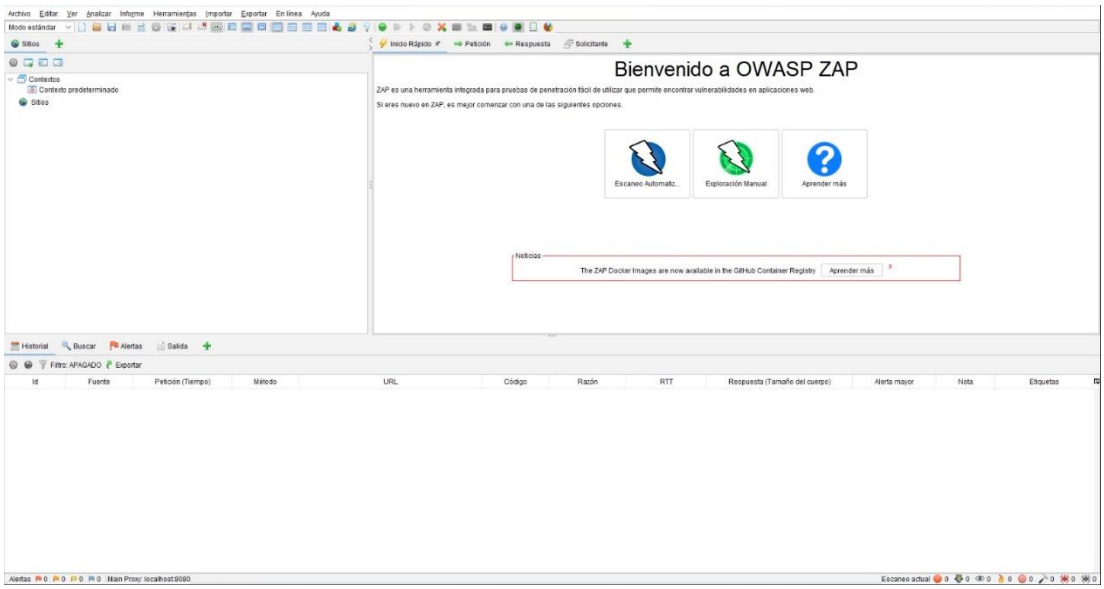
2.4.3 Ejecución de escaneo de vulnerabilidades

Se llevó a cabo un análisis de las vulnerabilidades en el Sistema de registro y control documental Quipux, utilizando la herramienta Zed Attack Proxy (ZAP) en su modo de ejecución de ataque por defecto. Durante el análisis, se proporcionó la URL predeterminada al ZAP, lo que permitió que la herramienta adquiriera una

lista de posibles solicitudes que podría realizar mediante un proceso automático de rastreo (crawling) a partir de dicha URL.

Después de ingresar la URL http://181.112.139.145/quipux, en la herramienta ZAP, se lleva a cabo ataques para evaluar el Quipux. Esto permitió confirmar la existencia de vulnerabilidades, así como analizar las configuraciones de los encabezados http, basándose en el contenido del cuerpo de las respuestas obtenidas.

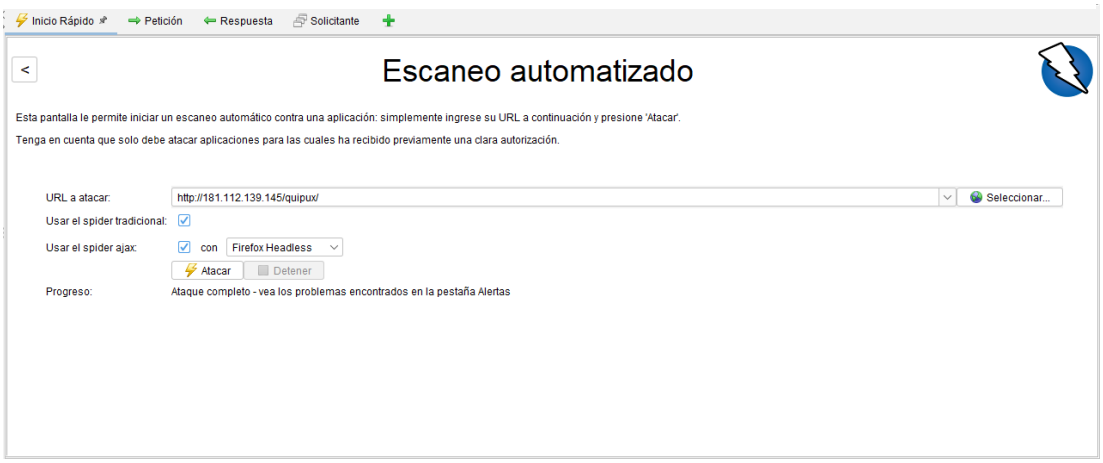
Figura 2: Herramienta OWASP ZAP



Fuente: OWASP ZAP

La Ilustración 2. Se representa el procedimiento utilizado para realizar el escaneo de vulnerabilidades en el Quipux mediante el uso de la herramienta OWASP ZAP en su modo de ejecución automático.

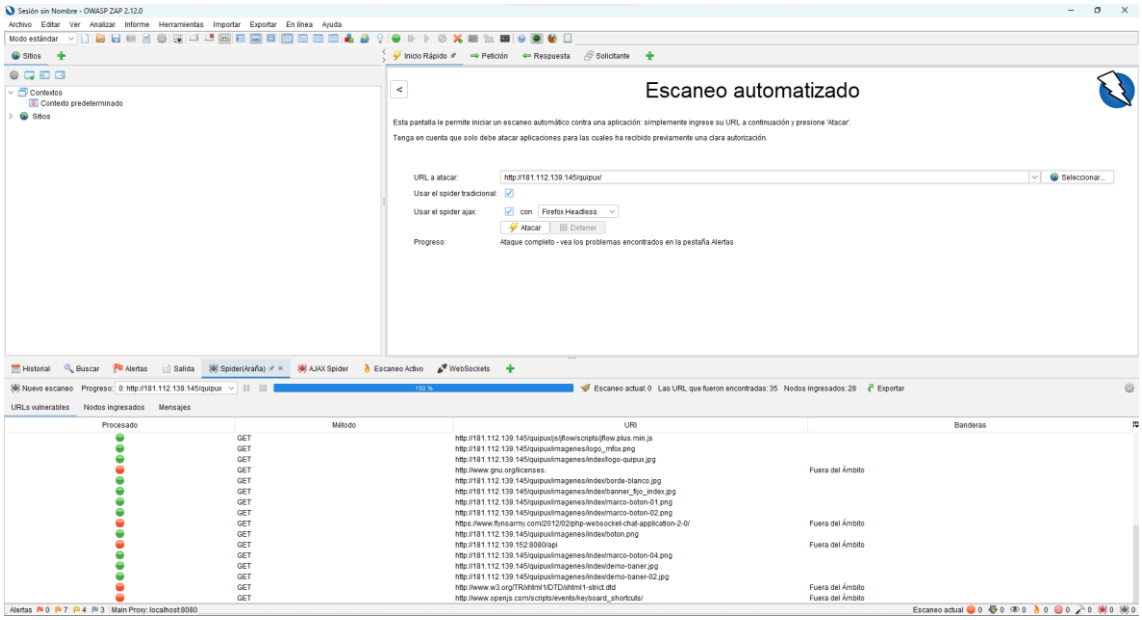
Figura 3: Ejecución de escaneo herramienta OWASP ZAP



Fuente: OWASP ZAP

En la Ilustración 3. Se ilustra el proceso de ejecución del escaneo del Sistema de registro y control documental (Quipux). Primero, se introduce la dirección electrónica del sistema, que en este caso es http://181.112.139.145/quipux/. A continuación, se hace clic en el botón "atacar" para iniciar el escaneo.

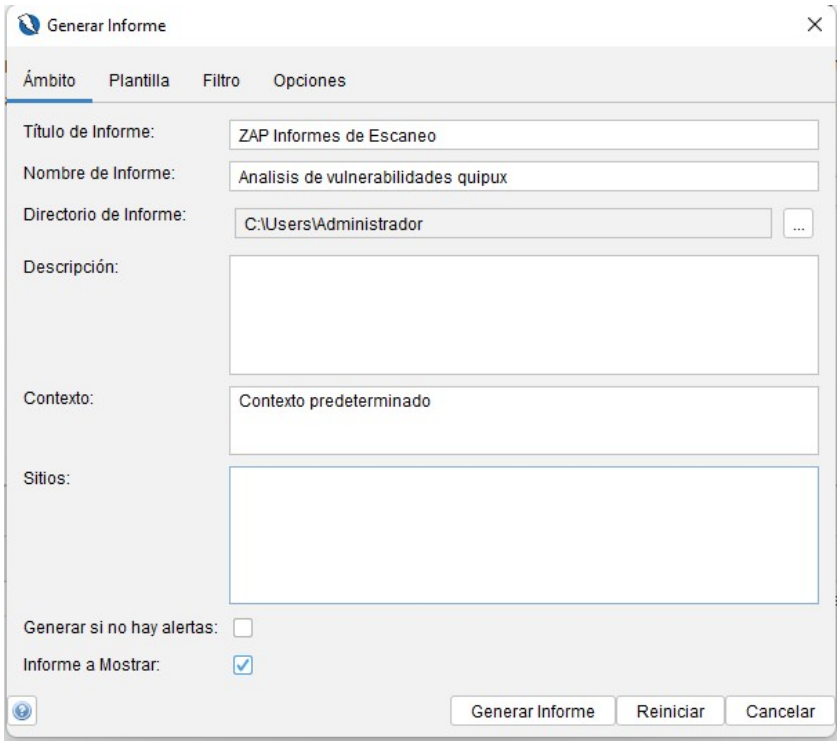
Figura 4: Análisis de escaneo sistema Quipux



Fuente: OWASP ZAP

En la Iluatración 4. Se representa el proceso de análisis y escaneo llevado a cabo por la herramienta OWASP ZAP a partir de la URL proporcionada. Esta herramienta permite detectar y clasificar los distintos tipos de vulnerabilidades identificadas, así como evaluar los riesgos asociados a ellas.

Figura 5: Generación de informe de escaneo sistema Quipux



Fuente: OWASP ZAP

La Ilustración 5. Se observa el proceso para generar él informa del escaneó realizado por la herramienta al sistema de control y registro documental (Quipux).

CAPÍTULO III: RESULTADOS

En este tercer capítulo de resultados, se presenta los hallazgos del análisis de vulnerabilidades de la aplicación Quipux utilizando la herramienta OWASP ZAP. El objetivo es identificar y evaluar los riesgos de seguridad en la aplicación para mejorar su protección y prevenir posibles amenazas.

En estos resultados se ha determinado los tipos de vulnerabilidades, así como el nivel de riesgos a los que está expuesto el sistema.

3.1 Tipos de vulnerabilidades encontradas

Tabla 3: tipos de vulnerabilidades y nivel de riesgo

Nombre	Nivel de riesgo
Archivo Oculto Encontrado	Medio
Ausencia de fichas (tokens) Anti-CSRF	Medio
Cabecera Content Security Policy (CSP) no configurada	Medio
Desconfiguración de Dominio cruzado	Medio
Librería JS Vulnerable	Medio
El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ""X-Powered-By""	Bajo
Strict-Transport-Security Header Not Set	Bajo

Fuente: OWASP ZAP

En la Tabla 7, se especifican los diferentes tipos de vulnerabilidades detectadas durante el análisis realizado por la herramienta ZAP en el Sistema de Gestión Documental (Quipux), junto con el nivel de riesgo asociado a cada una de ellas.

3.2 Funcionamiento

3.2.1 Resultados de escaneo de riesgo medio

Tabla 4: Vulnerabilidades por "Archivo Oculto Encontrado"

Medio	Archivo Oculto Encontrado
Descripción	Se identificó un archivo confidencial como accesible o disponible. Esto puede filtrar información administrativa, de configuración o de credenciales que puede ser aprovechada por un individuo malintencionado para atacar más adelante el sistema o mejorar la manera en que realiza ataques de ingeniería social.
URL	http://181.112.139.145/info.php
Método	GET
Parámetros	
Ataque	
Evidencia	HTTP/1.1 200 OK
Instancia	1

Fuente: OWASP ZAP

Alternativa de solución

Para solventar esta vulnerabilidad detectada en el sistema de gestión documental (Quipux), realiza las siguientes acciones que se detallan a continuación.

Tabla 5: Posible solución de la vulnerabilidad por archivo oculto

Solución	Considera si este componente es realmente necesario en producción; si no es así, desactívelo. Si es así, asegúrate de que el acceso requiera la autenticación y

	autorización adecuadas, o limita la exposición solo a sistemas internos o IPs de origen definidas, etc.
Referencia	https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-on-Web-Servers.html https://www.php.net/manual/en/function.phpinfo.php
CWE Id	538
WASC Id	13
Plugin Id	40035

Fuente: OWASP ZAP

Se logró detectar la vulnerabilidad exitosamente al recuperar una mayor cantidad de datos que fueron originalmente retornados, mediante la manipulación del parámetro.

Una vulnerabilidad informática de tipo "archivo oculto encontrado" en una aplicación web ocurre cuando un atacante puede acceder a archivos que no deberían estar accesibles públicamente. Esto puede conducir a una serie de problemas de seguridad y exposición de datos confidenciales.

A continuación, los puntos clave de cómo se desarrolla esta vulnerabilidad:

1. **Acceso a archivos ocultos**: Una aplicación web normalmente almacena datos y archivos en su servidor, algunos de los cuales deben estar protegidos y no ser accesibles para el público en general. Sin embargo, debido a una configuración incorrecta o un error en el código, algunos archivos pueden quedar expuestos y accesibles.
2. **Identificación de archivos sensibles**: El atacante busca posibles archivos que podrían ser sensibles y estar ocultos para los usuarios no autorizados. Estos archivos pueden incluir contraseñas, datos de configuración, información de la base de datos, archivos de registro o cualquier otro tipo de información confidencial.
3. **Manipulación de URL o parámetros**: El atacante intenta manipular las URL o los parámetros en las solicitudes HTTP enviadas a la aplicación web. Esto se hace para acceder a rutas o directorios que normalmente no serían visibles para un usuario regular.

4. **Enumeración de archivos**: En algunos casos, el atacante puede intentar adivinar nombres de archivo o rutas comunes mediante técnicas de enumeración. Por ejemplo, podría probar "archivo1.txt", "archivo2.txt", "archivo3.txt", y así sucesivamente, para encontrar aquellos que sí estén accesibles.
5. **Escalada de privilegios**: Si el atacante logra acceder a archivos sensibles, puede obtener información privilegiada o incluso utilizarla para escalar sus privilegios y obtener un mayor acceso a la aplicación o al servidor.
6. **Consecuencias**: Una vez que el atacante tiene acceso a archivos ocultos, puede obtener información confidencial del sistema o de los usuarios, lo que podría resultar en robo de datos, suplantación de identidad, compromiso de cuentas, entre otros riesgos de seguridad.

Para evitar esta vulnerabilidad, se implementó buenas prácticas de seguridad, tales como:

- Control de acceso adecuado: Asegurarse de que los archivos y rutas estén configurados para ser accesibles solo por usuarios autorizados. Implementar mecanismos de autenticación y autorización para restringir el acceso a archivos sensibles.
- Validación de entrada: Sanitizar y validar todas las entradas de datos proporcionadas por los usuarios para evitar ataques de manipulación de URL o parámetros.
- Configuración segura del servidor: Asegurarse de que la configuración del servidor web esté bien ajustada para prevenir la exposición de archivos sensibles.
- Uso de autenticación de dos factores (2FA): Implementar 2FA puede dificultar aún más el acceso no autorizado incluso si un atacante logra acceder a ciertos archivos.
- Pruebas de seguridad regulares: Realizar pruebas de penetración y auditorías de seguridad de manera periódica para identificar y corregir posibles vulnerabilidades en la aplicación web.

Explicación

Esta vulnerabilidad se refiere a la presencia de archivos o recursos ocultos en el sistema que pueden ser accesibles públicamente. Estos archivos podrían contener información sensible o códigos maliciosos.

Impacto

Un atacante podría descubrir y acceder a archivos confidenciales que no deberían estar disponibles para el público. Esto podría llevar a la exposición de datos sensibles o incluso a la ejecución de acciones maliciosas.

OWASP ZAP, también encontró otro Riesgo Medio: Ausencia de fichas (tokens) Anti-CSRF

Tabla 6: Vulnerabilidades por Ausencia de fichas (tokens) Anti-CSRF

Medio	**Ausencia de fichas (tokens) Anti-CSRF**
Descripción	No se encontraron fichas (tokens) Anti-CSRF en un formulario HTML. Una solicutud falsa entre sitios en un ataque que compromete y obliga a una víctima a enviar su solicitud HTTP a un destino objetivo sin su conocimiento o intención para poder realizar una acción como víctima. La causa oculta es la funcionalidad de la aplicación utilizando acciones de URL/formulario que pueden ser adivinados de forma repetible. La naturaleza del ataque es que CSRG explota la confianza que un sitio web proporciona a un usuario. Por el contrario, las cadenas de comandos de los sitios cruzados (XSS) explotan la confianza que un usuario proporciona en un sitio web. Al igual que XSS, los ataques CSRG no son de forma necesaria de sitios cruzados, pero hay la posibilidad de que si pueden serlo. La falsificación de las solicitudes ente los sitios también se conoce como CSRF, XSRG, ataques con un solo clic, montaje de sesión, diputado confundido y navegación en alta mar. Los ataques de CSRG son muy efectivos en varias situaciones, que incluyen: *La victima tiene una sesión activa en el sitio de destino.

	*La víctima se autoriza por medio de la autenticación HTTP en el sitio de destino. *La víctima se encuentra en la misma red local que el sitio de destino. CSRF se ha utilizado especialmente para poder realizar una acción contra un sitio objetivo utilizando los privilegios de la víctima, pero se han revelado técnicas recientes para difundir información al obtener el acceso a la respuesta. El riesgo de divulgación de información aumenta de forma drástica cuando el sitio de destino se encuentra vulnerable a XSS, porque XSS se puede utilizar como una plataforma para CSRF, lo que le permite al atacante que opere desde adentro de los límites de la misma política de origen.
URL	http://181.112.139.145/quipux/login.php
Método	GET
Parámetros	
Ataque	
Evidencia	<form name="form_login" action="" method="post" onSubmit="return validar_login();">
Instancia	1

Fuente: OWASP ZAP

Alternativa de solución

Para contrarrestar esta vulnerabilidad detectada en el sistema de gestión documental (Quipux), se realizó las siguientes acciones que se detallan a continuación.

Tabla 7: Posibles soluciones de la vulnerabilidad por Ausencia de fichas (tokens) Anti-CSRF

Solución	Frase: Arquitectura y Diseño Utilice una biblioteca o marco comprobado que no acepte que ocura esta debilidad o que proporcione construcciones que permitan que esta debilidad sea mas sencilla de

	evitar. Por ejemplo, utilice el paquete anti-CSRG como el CSRGuard de OWASP. Fase: Implementación Asegúrese de que su aplicación esté libre de fallas de secuencias de comandos entre sitios, ya que la mayoría de las defensas de CSRF pueden detenerse por alto por medio del uso de secuencias de comandos manejadas por el atacante. Fase: Arquitectura y Diseño Origina un nonce único para cada uno de los formularios, coloque el nonce en el formularo y confirme la independencia al obtener el formulario. Asegúrese de que el nonce no sea predecible (CWE-330). Usted tiene que tener en cuenta que esto puede pasar desapercibido utilizando XSS. Identificar las operaciones que sean especialmente peligrosas. Cuando el usuario desarrolla una operación peligrosa, envíe una solicitud de confirmación de forma separada para poder garantizar que el usuario tenga la intención de desarrollar esa operación. Usted tiene que tener en cuenta que esto puede pasar desapercibido utilizando XSS. Utilice el control de gestión de la sesión de ESAPI. Este control introduce un elemento para CSRF. No utilice el método GET para ninguna de las solicitudes que puedan desencadenar un cambio de estado.

	Fase: Implementación Revise que la solicitud se creó en la página esperada. Esto podría quebrar la funcionalidad auténtica, ya que los usuarios o los representantes puede ser que hayan desactivado el envío de Referer por motivos de privacidad.
Referencia	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Plugin Id	10202

Fuente: OWASP ZAP

Explicación

Esta vulnerabilidad se relaciona con la falta de implementación de fichas de seguridad (tokens) para prevenir ataques de falsificación de solicitudes entre sitios (CSRF). Sin fichas, un atacante podría engañar a un usuario para que realice acciones no deseadas en el sistema.

Impacto

Los atacantes podrían manipular a los usuarios para que realicen acciones no autorizadas, como cambiar contraseñas, realizar transacciones o modificar configuraciones, lo que comprometería la integridad y seguridad de la aplicación.

De riesgo medio también se tuvo una vulnerabilidad de Librería JS Vulnerable

Tabla 8: Vulnerabilidad por Librería JS Vulnerable

Medio	Librería JS Vulnerable
Descripción	La librería identificada jquery, versión 1.8.0 es vulnerable.
URL	http://181.112.139.145/quipux/js/jquery.min.js

Método	GET
Parámetros	
Ataque	
Evidencia	,jquery:"1.8.0"
Instancia	1

Fuente: OWASP ZAP

Alternativa de Solución

Con el objetivo de resolver la vulnerabilidad detectada en el Sistema de Gestión Documental (Quipux), se llevarán a cabo las siguientes acciones que se describen a continuación.

Tabla 9: Posible solución a la vulnerabilidad Librería JS Vulnerable

Solución	Actualiza a la última versión de jquery.
Referencia	https://nvd.nist.gov/vuln/detail/CVE-2012-6708 https://github.com/jquery/jquery/issues/2432 http://research.insecurelabs.org/jquery/test/ https://nvd.nist.gov/vuln/detail/CVE-2019-11358 https://bugs.jquery.com/ticket/11974 https://github.com/jquery/jquery.com/issues/162 https://nvd.nist.gov/vuln/detail/CVE-2020-7656 http://blog.jquery.com/2016/01/08/jquery-2-2-and-1-12-released/ http://bugs.jquery.com/ticket/11290 https://blog.jquery.com/2019/04/10/jquery-3-4-0-released/ https://nvd.nist.gov/vuln/detail/CVE-2015-9251 https://github.com/advisories/GHSA-q4m3-2j7h-f7xw https://github.com/jquery/jquery/commit/753d591aea698e57d6db58c9f722cd0808619b1b https://blog.jquery.com/2020/04/10/jquery-3-5-0-released/
CWE Id	829
WASC Id	
Plugin Id	10003

Fuente: OWASP ZAP

Explicación

Esta vulnerabilidad se refiere al uso de librerías JavaScript desactualizadas o con vulnerabilidades conocidas, lo que podría permitir a los atacantes explotar estas debilidades.

Impacto

Los atacantes podrían aprovechar las vulnerabilidades en las librerías para ejecutar código malicioso en el navegador de los usuarios, lo que podría llevar a la toma de control del sistema o la exposición de información sensible.

De igual manera se reconoció la vulnerabilidad de riesgo medio denominada Cabecera *Content Security Policy (CPS)* no configurada.

Tabla 10: Vulnerabilidad de Cabecera Content Security Policy (CPS) no configurada

Medio	Cabecera Content Security Policy (CSP) no configurada
Descripción	La Política de seguridad de contenido (CSP) es una capa adicional de seguridad que ayuda a detectar y mitigar ciertos tipos de ataques, incluidos Cross Site Scripting (XSS) y ataques de inyección de datos. Estos ataques se utilizan para todo, desde el robo de datos hasta la desfiguración del sitio o la distribución de malware. CSP proporciona un conjunto de encabezados HTTP estándar que permiten a los propietarios de sitios web declarar fuentes de contenido aprobadas que los navegadores deberían poder cargar en esa página; los tipos cubiertos son JavaScript, CSS, marcos HTML, fuentes, imágenes y objetos incrustados como applets de Java, ActiveX, archivos de audio y video.
URL	http://181.112.139.145/favicon.ico
Método	GET
Parámetros	
Ataque	
Evidencia	

URL	http://181.112.139.145/quipux
Método	GET
Parámetros	
Ataque	
Evidencia	
URL	http://181.112.139.145/quipux/
Método	GET
Parámetros	
Ataque	
Evidencia	
URL	http://181.112.139.145/quipux/js/fancywebsocket.js
Método	GET
Parámetros	
Ataque	
Evidencia	
URL	http://181.112.139.145/quipux/js/jquery171.js
Método	GET
Parámetros	
Ataque	
Evidencia	
URL	http://181.112.139.145/quipux/login.php

Método	GET
Parámetros	
Ataque	
Evidencia	
URL	http://181.112.139.145/robots.txt
Método	GET
Parámetros	
Ataque	
Evidencia	
URL	http://181.112.139.145/sitemap.xml
Método	GET
Parámetros	
Ataque	
Evidencia	
Instancia	8

Fuente: OWASP ZAP

Alternativa de solución

Para solventar esta vulnerabilidad detectada en el sistema de gestión documental (Quipux), realiza las siguientes acciones que se detallan a continuación.

Tabla 11: Posible solución a la vulnerabilidad Content Security Policy (CSP) no configurada

Solución	Asegúrese de que su servidor web, servidor de aplicaciones, balanceador de carga, etc. esté configurado para establecer la cabecera Content-Security-Policy.
Referenci	https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy

a	https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html http://www.w3.org/TR/CSP/ http://w3c.github.io/webappsec/specs/content-security-policy/csp-specification.dev.html http://www.html5rocks.com/en/tutorials/security/content-security-policy/ http://caniuse.com/#feat=contentsecuritypolicy http://content-security-policy.com/
CWE Id	693
WASC Id	15
Plugin Id	10038

Fuente: OWASP ZAP

Explicación

La falta de configuración de una Política de Seguridad de Contenido (CSP) permite que los recursos de una página web se carguen desde cualquier origen, lo que puede exponer a riesgos de ataques como XSS.

Impacto

Los ataques de Cross-Site Scripting (XSS) podrían ser más efectivos, lo que permitiría a los atacantes inyectar scripts maliciosos en la aplicación y comprometer la seguridad de los usuarios y sus datos.

Por último, en el conjunto de vulnerabilidades de riesgo medio se obtuvo otra denominada "Desconfiguración de Dominio cruzado".

Tabla 12: Vulnerabilidad Desconfiguración de Dominio cruzado

Medio	**Desconfiguración de Dominio cruzado**
Descripción	Descargas de datos del navegador web podría ser posible, debido a una desconfiguración del intercambio de recursos cruzados de origen (CORS) en el servidor web
URL	https://www.hostingcloud.racing/ua4L.js
Método	GET

Parámetros	
Ataque	
Evidencia	Access-Control-Allow-Origin: *
Instancia	1

Fuente: OWASP ZAP

Alternativa de Solución

Con el objetivo de resolver la vulnerabilidad detectada en el Sistema de Gestión Documental (Quipux), se llevarán a cabo las siguientes acciones que se describen a continuación.

Tabla 13: Posible mitigación de la vulnerabilidad Desconfiguración de Dominio cruzado

Solución	Asegúrese que los datos sensibles no están disponibles de manera no autenticada (usando dirección IP listado-blanco, por ejemplo). Configurar el encabezado HTTP ""Access-Control-Allow-Origin" a un conjunto de dominios más restrictivo, o remover completamente todos los encabezados CORS, para permitir que el navegador web refuerce la política de mismo origen (SOP) en una manera mas restrictiva.
Referencia	https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy
CWE Id	264
WASC Id	14
Plugin Id	10098

Fuente: OWASP ZAP

Explicación

Esta vulnerabilidad ocurre cuando la configuración del dominio cruzado (CORS) permite que sitios no autorizados accedan a recursos restringidos en el servidor.

Impacto

Un atacante podría hacer uso indebido de esta configuración para acceder a recursos privados y sensibles en el servidor, lo que podría resultar en la filtración de datos o el acceso no autorizado.

3.2.2 Resultados de escaneo de riesgo bajo

A continuación, se detallan los resultados encontrados en el escaneo realizado con la herramienta ZAP en el entorno de producción del Sistema de Gestión Documental (Quipux), específicamente abordando las vulnerabilidades con riesgo de nivel bajo

Tabla 14: Vulnerabilidad Strict-Tranport-Security Header Not Set

Bajo	**Strict-Transport-Security Header Not Set**
Descripción	HTTP Strict Transport Security (HSTS) es un mecanismo de política de seguridad web mediante el cual un servidor web declara que los agentes de usuario que cumplen (como un navegador web) deben interactuar con él utilizando solo conexiones HTTPS seguras (es decir, HTTP en capas sobre TLS / SSL). HSTS es un protocolo de seguimiento de estándares IETF y se especifica en RFC 6797.
URL	https://www.hostingcloud.racing/ua4L.js
Método	GET
Parámetros	
Ataque	
Evidencia	
Instancia	1

Fuente: OWASP ZAP

Con el objetivo de resolver la vulnerabilidad detectada en el Sistema de Gestión Documental (Quipux), se llevarán a cabo las siguientes acciones que se describen a continuación.

Tabla 15: Posible solución a la Vulnerabilidad Strict-Tranport-Security Header Not Set

Solución	Asegúrese de que su servidor web, servidor de aplicaciones, equilibrador de carga, etc. está configurado para aplicar Strict-Transport-Security.
Referencia	https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html https://owasp.org/www-community/Security_Headers http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security http://caniuse.com/stricttransportsecurity http://tools.ietf.org/html/rfc6797
CWE Id	319
WASC Id	15
Plugin Id	10035

Fuente: OWASP ZAP

Explicación

Si el encabezado HTTP *'Strict-Transport-Security'* (HSTS) no está configurado, los navegadores no estarán obligados a establecer conexiones seguras a través de HTTPS, lo que podría dejar al sistema vulnerable a ataques de intercepción.

Impacto

Los atacantes podrían aprovechar la falta de HSTS para realizar ataques de intermediario y escuchar el tráfico entre el cliente y el servidor, comprometiendo la confidencialidad de la información transmitida.

Como última vulnerabilidad encontrada se registró la siguiente:

Tabla 16: Vulnerabilidad por: El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ""X-Powered-By""

Bajo	**El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ""X-Powered-By""**
Descripción	El servidor de la web/aplicación está divulgando información mediante uno o más encabezados de respuesta HTTP ""X-Powered-By"". El

	acceso a tal información podría facilitarle a los atacantes la identificación de otros marcos/componentes de los que su aplicación web depende y las vulnerabilidades a las que pueden estar sujetos tales componentes.
URL	http://181.112.139.145/quipux
Método	GET
Parámetros	
Ataque	
Evidencia	X-Powered-By: PHP/5.6.40
URL	http://181.112.139.145/quipux/
Método	GET
Parámetros	
Ataque	
Evidencia	X-Powered-By: PHP/5.6.40
URL	http://181.112.139.145/quipux/login.php
Método	GET
Parámetros	
Ataque	
Evidencia	X-Powered-By: PHP/5.6.40
Instancia	3

Fuente: OWASP ZAP

Para solventar esta vulnerabilidad detectada en el sistema de gestión documental (Quipux), realiza las siguientes acciones que se detallan a continuación.

Tabla 17: Solución alternativa a la vulnerabilidad por: El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ""X-Powered-By""

Solución	Asegúrese que su servidor web, servidor de aplicación, equilibrador de carga, etc. está configurado para suprimir encabezados ""X-Powered-By".
Referencia	http://blogs.msdn.com/b/varunm/Archive/2013/04/23/Remove-Unwanted-http-Response-headers.aspx http://www.troyhunt.com/2012/02/shhh-don ' t-deje-la-respuesta-headers.html
CWE Id	200
WASC Id	13
Plugin Id	10037

Fuente: OWASP ZAP

Explicación

La presencia del encabezado HTTP *'X-Powered-By'* en las respuestas del servidor puede revelar información sobre las tecnologías y versiones utilizadas, lo que facilita a los atacantes identificar posibles puntos de entrada y vulnerabilidades.

Impacto

Los atacantes podrían utilizar esta información para planificar ataques específicos dirigidos a las tecnologías conocidas, aumentando el riesgo de explotación y compromiso del sistema.

4 CONCLUSIONES

1. La seguridad del software es un aspecto crítico en el desarrollo tecnológico actual. La detección y mitigación de amenazas y vulnerabilidades deben ser consideradas como una prioridad en cada etapa del ciclo de vida del software.

2. La elección adecuada de herramientas de seguridad puede marcar la diferencia en la calidad del producto final. Es esencial invertir tiempo y recursos en la evaluación de diversas opciones para seleccionar la más adecuada a las necesidades y características específicas de la empresa.

3. La inversión en medidas preventivas desde las primeras etapas del desarrollo del software puede resultar en ahorros significativos a largo plazo. Identificar y resolver problemas de seguridad tempranamente reduce la probabilidad de enfrentar costosos riesgos en el futuro.

4. La accesibilidad de la tecnología para un público más amplio también conlleva mayores peligros potenciales. Por lo tanto, es imperativo que las empresas estén proactivamente preparadas para enfrentar posibles amenazas de seguridad.

5. La capacitación del personal en seguridad informática es fundamental para crear una cultura de seguridad dentro de la empresa. Los desarrolladores deben estar equipados con los conocimientos necesarios para identificar y abordar problemas de seguridad desde el inicio del desarrollo.

6. El monitoreo constante de la seguridad del software en producción es esencial para mantener un nivel óptimo de protección. Las amenazas cambian y evolucionan con el tiempo, por lo que es vital estar siempre alerta.

7. Cumplir con estándares y buenas prácticas reconocidas, como los proporcionados por OWASP, es un enfoque sólido para garantizar la seguridad del software. Estos estándares son actualizados por la comunidad de expertos en seguridad y ofrecen pautas valiosas.

8. La seguridad del software no debe ser vista como un costo adicional, sino como una inversión en la calidad del producto final y en la reputación de la empresa. Los costos asociados con garantizar la seguridad deben ser asumidos para evitar mayores pérdidas económicas y daños a la marca.

9. Las pruebas de vulnerabilidad deben ser exhaustivas y periódicas, utilizando herramientas especializadas para identificar de manera eficiente posibles debilidades en el código fuente y en el producto final.

10. A pesar de utilizar metodologías ágiles como Extreme Programming, es importante reconocer que siempre habrá fallos que surjan. La adopción de estándares y buenas prácticas puede ayudar a reducir estos fallos y minimizar las consecuencias económicas.

La seguridad del software es una responsabilidad compartida por todo el equipo de desarrollo y la empresa en general. Adoptar prácticas sólidas, utilizar herramientas adecuadas y mantener una cultura de seguridad contribuirá a mejorar la calidad del software y proteger los activos de la empresa frente a posibles amenazas y vulnerabilidades.

5 RECOMENDACIONES

1. Mejorar el proceso de selección de herramientas: Asegurarse de realizar una evaluación exhaustiva de las distintas herramientas disponibles para la detección de amenazas y vulnerabilidades en el código fuente. Buscar aquellas que mejor se adapten a las necesidades y características específicas del GADPC.

2. Esto ayudará a reducir costos y riesgos futuros al encontrar y resolver problemas de seguridad antes de que se conviertan en amenazas mayores.

3. Capacitación en seguridad informática: Brinda a los desarrolladores y al personal involucrado en el desarrollo de software capacitación en seguridad informática. Esto les permitirá identificar y abordar de manera proactiva posibles vulnerabilidades, mejorando la calidad del software final.

4. Monitoreo constante: Establece un proceso de monitoreo continuo de la seguridad del software una vez que se encuentre en producción. Esto ayudará a detectar y mitigar rápidamente nuevas amenazas que puedan surgir con el tiempo.

5. Cumplimiento de estándares y buenas prácticas: Asegúrate de seguir estándares reconocidos y buenas prácticas en el desarrollo de software, como los proporcionados por OWASP, para garantizar que los productos cumplan con los requisitos de seguridad más actualizados.

6. Integración de seguridad en el desarrollo ágil: Si se sigue la metodología Extreme Programming u otra metodología ágil, es importante integrar aspectos de seguridad desde el principio y a lo largo de todo el proceso de desarrollo.

7. Pruebas exhaustivas de vulnerabilidad: Realiza pruebas de vulnerabilidad periódicas y exhaustivas utilizando herramientas especializadas como ZAP, con el fin de identificar posibles debilidades y fallos en el software.

8. Asignación de recursos adecuados: Asegúrate de asignar recursos suficientes para garantizar la seguridad del software, tanto en términos de personal capacitado como en herramientas y tecnologías.

9. Actualización y parcheo regular: Mantén el software y las bibliotecas de terceros siempre actualizados, aplicando parches y correcciones de seguridad de manera regular para evitar vulnerabilidades conocidas.

10. Cultura de seguridad: Fomenta una cultura organizacional donde la seguridad de la información y del software sea una prioridad para todos los miembros del equipo. La seguridad debe ser un compromiso compartido en toda la empresa.

Implementar estas recomendaciones ayudará a fortalecer la seguridad del software desarrollado por GADPC, mejorando la calidad de sus productos y reduciendo los riesgos asociados a amenazas y vulnerabilidades.

Bibliografía

¿Qué es Quipux? – Sistema de Gestión Documental Quipux. (2023). Gestiondocumental.gob.ec. https://web.gestiondocumental.gob.ec/que-es-quipux/

About the OWASP Foundation | OWASP Foundation. (2023). Owasp.org. https://owasp.org/about/

Group, T. (04 de diciembre de 2021). cpl.thalesgroup.com. Obtenido de https://cpl.thalesgroup.com/es/software-monetization/what-is-software-security

Hernández, M. (21 de Marzo de 2022). Bidaidea | Ciberseguridad e Inteligencia. Obtenido de ciberseguridadbidaidea.com: https://ciberseguridadbidaidea.com/fases-del-pentesting/

Kaur, G., & Gupta, N. (2021). A survey on web application security testing tools. Journal of Information Security and Applications, 59, 102798. https://doi.org/10.1016/j.jisa.2020.102798

Khan, A. M., & Khan, M. A. (2019). Web application security: A comprehensive guide for beginners. CRC Press.

Leonardo, D. (2021). Vulnerabilidades en aplicaciones web utilizando la metodología de "proyecto abierto de seguridad de aplicaciones web." Pucesa.edu.ec. https://repositorio.pucesa.edu.ec/handle/123456789/3175

Li, Z., Han, X., Zhan, Y., & Yang, Y. (2020). The effectiveness of OWASP ZAP in detecting vulnerabilities of web applications. Journal of Cybersecurity, 6(1), tyaa003. https://doi.org/10.1093/cybsec/tyaa003

McLean, D. (2015). Understanding and preventing software vulnerabilities. CRC Press.

Metodologías para la auditoria de la seguridad. (2017). Linkedin.com. https://es.linkedin.com/pulse/metodolog%C3%ADas-para-la-auditoria-de-seguridad-kevin-rodriguez-lago

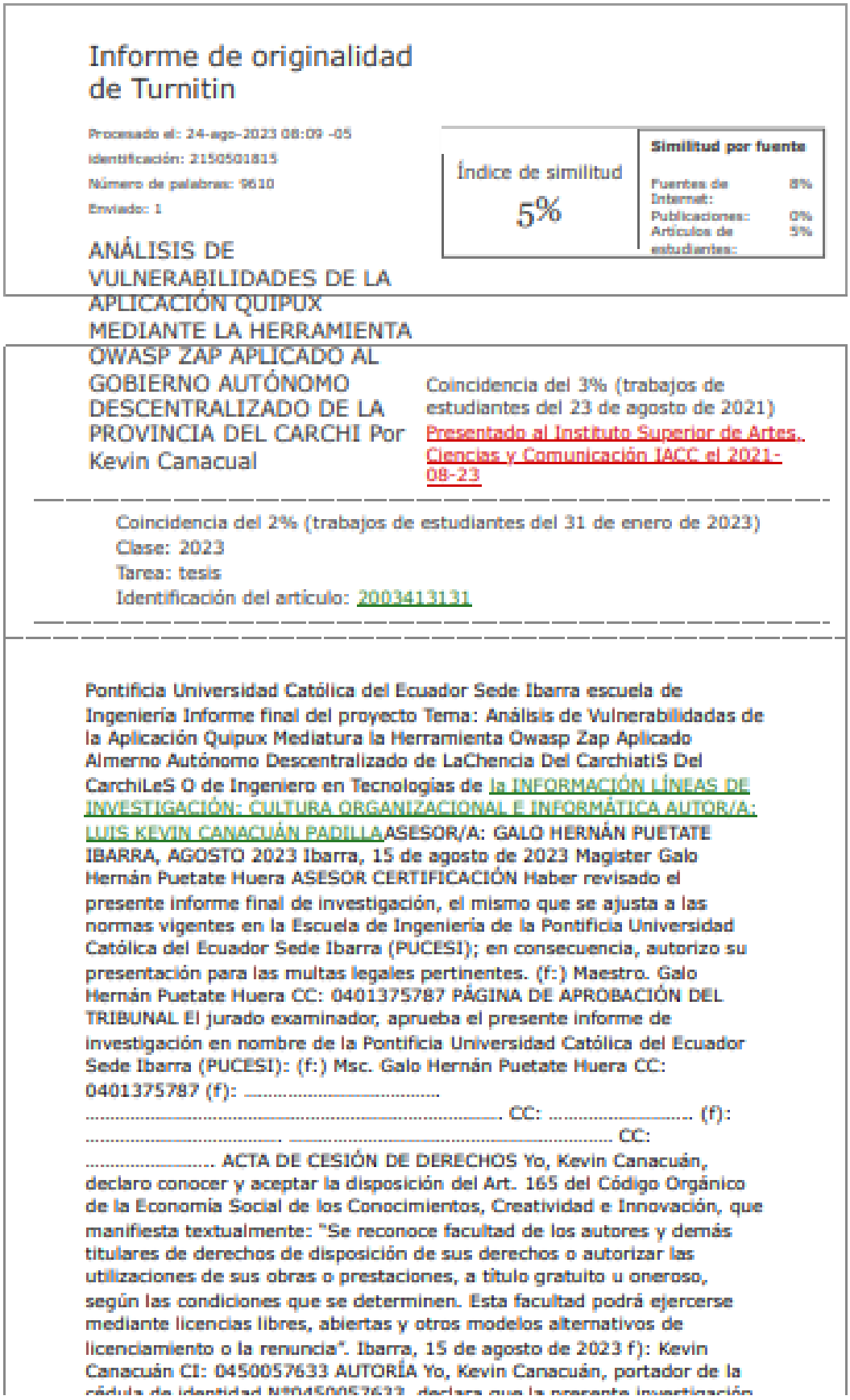
OWASP Top Ten | OWASP Foundation. (2020). Owasp.org. https://owasp.org/www-project-top-ten/

OWASP. (2020). OWASP Zed Attack Proxy Project. Recuperado de https://owasp.org/www-project-zap/

Portilla, A. D. (16 de Junio de 2020). ANÁLISIS DE SEGURIDAD DEL SOFTWARE DE GESTIÓN ACADÉMICA PARA ECUALATINO S.A. Ibarra, Imbabura, Ecuador.

Team, A. (10 de noviembre de 2020). ambit Bulding solutions together. Obtenido de www.ambit-bst.com: https://www.ambit-bst.com/blog/tipos-de-vulnerabilidades-y-amenazas-inform%C3%A1ticas

Anexos



Informe de originalidad de Turnitin

Procesado el: 24-ago-2023 08:09 -05
Identificación: 2150501815
Número de palabras: 9610
Enviado: 1

ANÁLISIS DE VULNERABILIDADES DE LA APLICACIÓN QUIPUX MEDIANTE LA HERRAMIENTA OWASP ZAP APLICADO AL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE LA PROVINCIA DEL CARCHI Por Kevin Canacual

Índice de similitud
5%

Similitud por fuente
Fuentes de Internet: 8%
Publicaciones: 0%
Artículos de estudiantes: 5%

Coincidencia del 3% (trabajos de estudiantes del 23 de agosto de 2021)
Presentado al Instituto Superior de Artes, Ciencias y Comunicación IACC el 2021-08-23

Coincidencia del 2% (trabajos de estudiantes del 31 de enero de 2023)
Clase: 2023
Tarea: tesis
Identificación del artículo: 2003413131

Pontificia Universidad Católica del Ecuador Sede Ibarra escuela de Ingeniería Informe final del proyecto Tema: Análisis de Vulnerabilidadas de la Aplicación Quipux Mediatura la Herramienta Owasp Zap Aplicado Almerno Autónomo Descentralizado de LaChencia Del CarchiatiS Del CarchiLeS O de Ingeniero en Tecnologías de la INFORMACIÓN LÍNEAS DE INVESTIGACIÓN: CULTUBA ORGANIZACIONAL E INFORMÁTICA AUTOR/A: LUIS KEVIN CANACUÁN PADILLAASESOR/A: GALO HERNÁN PUETATE IBARRA, AGOSTO 2023 Ibarra, 15 de agosto de 2023 Magister Galo Hernán Puetate Huera ASESOR CERTIFICACIÓN Haber revisado el presente informe final de investigación, el mismo que se ajusta a las normas vigentes en la Escuela de Ingeniería de la Pontificia Universidad Católica del Ecuador Sede Ibarra (PUCESI); en consecuencia, autorizo su presentación para las multas legales pertinentes. (f:) Maestro. Galo Hernán Puetate Huera CC: 0401375787 PÁGINA DE APROBACIÓN DEL TRIBUNAL El jurado examinador, aprueba el presente informe de investigación en nombre de la Pontificia Universidad Católica del Ecuador Sede Ibarra (PUCESI): (f:) Msc. Galo Hernán Puetate Huera CC: 0401375787 (f): .... CC: ..... (f): .... CC: ..... ACTA DE CESIÓN DE DERECHOS Yo, Kevin Canacuán, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: "Se reconoce facultad de los autores y demás titulares de derechos de disposición de sus derechos o autorizar las utilizaciones de sus obras o prestaciones, a título gratuito u oneroso, según las condiciones que se determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia". Ibarra, 15 de agosto de 2023 f): Kevin Canacuán CI: 0450057633 AUTORÍA Yo, Kevin Canacuán, portador de la cédula de identidad N°0450057633, declaro que la presente investigación

CARCHI PREFECTURA
construimos bienestar

GOBIERNO AUTÓNOMO DESCENTRALIZADO DE LA PROVINCIA DEL CARCHI

CARTA DE ACEPTACIÓN

Oficio No GADPC-DATH-GC-0040-2023

Tulcán, 03 de marzo de 2023

Señor
Luis Kevin Canacuán Padilla
ESTUDIANTE DE LA CARRERA DE TECNOLOGIAS DE LA INFORMACIÓN DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE-IBARRA
Presente.

De mi consideración:

En atención al oficio s/n de fecha 28 de febrero de 2023, en donde solicita la autorización para la elabroación del proyecto de tesis denominado "Análisis de vulnerabilidades de la aplicación Quipux, mediante la herramienta OWASP ZAP aplicado al Gobierno Autónomo Descentralizado de la provincia del Carchi."; me permito indicar que cuenta con la autorización y el visto favorable mediante memorando N° GADPC-UTIC-JC-2023-003-M.

Particular que pongo en su conocimiento para los fines pertinentes.

Atentamente,

Psic. Gioyanny Campoverde
DIRECTOR ADMINISTRATIVO Y TALENTO HUMANO (E)
GC/ae

CARTA DE ENTREGA

Tulcán, 14 de agosto 2023

Señores UTIC del GAD de la provincia del Carchi
Presente. -

Estimados miembros del departamento de UTIC del GADPC, me dirijo a ustedes con el propósito de hacer entrega formal del proyecto de investigación titulado **"Análisis de vulnerabilidades de la aplicación Quipux mediante la herramienta OWASP ZAP aplicado al Gobierno Autónomo Descentralizado de la Provincia Del Carchi"** realizado como parte de mi trabajo de titulación en la Pontifica Universidad Católica del Ecuador Sede Ibarra.

El proyecto tiene como objetivo la mitigación de vulnerabilidades que existen en el Quipux. En este sentido, el enfoque propuesto revela áreas de mejora en términos de seguridad y ofrece recomendaciones concretas para mitigar las vulnerabilidades descubiertas en la aplicación utilizada por el GADPC.

Quisiera expresar mi gratitud por la oportunidad brindada para llevar a cabo esta investigación y por la valiosa colaboración y apoyo proporcionados durante el proceso. Mi compromiso ha sido abordar este proyecto con la mayor seriedad y profesionalismo, con el objetivo de contribuir a la mejora y fortalecimiento de la seguridad en las operaciones y sistemas del Gobierno Autónomo Descentralizado de la Provincia del Carchi.

Adjunto a esta carta, encontrarán una copia completa y detallada del proyecto de investigación, que incluye análisis, resultados, conclusiones y recomendaciones pertinentes. Estoy a disposición para cualquier consulta o aclaración adicional que puedan requerir con respecto al contenido del proyecto.

Agradezco sinceramente su tiempo y consideración, y quedo a la espera de cualquier orientación adicional que puedan proporcionarme en relación a la revisión y evaluación del proyecto.

Para constancia de elaboración, ejecución y entrega se firma de manera conjunta.

Atentamente,

Luis Kevin Canacuán
ESTUDIANTE PUCE-I

Msc. Armando Freire
COORDINADOR UTIC GAD CARCHI