Universidad CENFOTEC

Maestría en Ciberseguridad

Documento final de Proyecto de Investigación Aplicada

Desarrollo de un plan de mejoras para mitigar riesgos y vulnerabilidades de Ciberseguridad en una plataforma web de crédito en una Institución Financiera Nacional.

Pérez Medina Carlos Andrés,

Molina Zúñiga Josué David.

marzo - 2024

Declaratoria de derechos de autor

Nosotros, Josue David Molina Zúñiga y Carlos Pérez Medina, titular de los derechos de autor del presente documento denominado “Desarrollo de un plan de mejoras para mitigar riesgos y vulnerabilidades de Ciberseguridad en una plataforma web de crédito en una Institución Financiera Nacional”, por la presente declaramos que somos los creadores originales de dicho trabajo y que poseemos todos los derechos de autor asociados con la misma.

Dedicatoria y/o agradecimientos.

El presente trabajo investigativo es dedicado principalmente a Dios, por ser el inspirador y darnos fuerza para continuar en este proceso de obtener uno de los anhelos más deseados por ambos.

A nuestras familias, por su amor, comprensión, paciencia y apoyo en todos estos largos meses del proceso de investigación, gracias a ustedes hemos logrado llegar hasta aquí y convertirnos en lo que somos.

Y a todas las personas que nos han apoyado desde docentes, compañeros y amigos que han logrado hacer que el trabajo se realice con éxito, sus valiosos consejos y aportes fueron fundamentales para dar forma y mejorar la calidad en el contenido de esta investigación.

Por último y no menos importante, queremos brindar un agradecimiento especial al director Miguel Pérez Montero por proporcionar tanto el conocimiento como la guía necesaria para llevar a cabo esta investigación, sin duda el profesor ha sido fundamental para la realización de este proyecto.

Hoja de aprobación del proyecto, firmada por los miembros del Comité Examinador. Sin esta hoja, el documento NO ES VÁLIDO.

Tabla de contenido

Contenido

Lista de figuras, gráficos, cuadros y/o ilustraciones.

Lista de Tablas

Lista de Ilustraciones

Resumen ejecutivo.

El proyecto de investigación aborda un tema tan crucial en el ámbito país como lo es la ciberseguridad en las páginas web, tema que hoy en día es moda, incluso es un hecho, de que es uno de los objetivos más buscados por los ciberdelincuentes y más aún cuando se trata de un aplicativo web de entidades bancarias. Si bien es cierto, nuestro país cuenta con un entorno financiero cada vez más globalizado y digitalizado, donde día a día se apunta a ofrecer nuevos servicios digitales para el consumo de los clientes en la búsqueda de generar más utilidades, también es un hecho que las instituciones han tenido que asumir una postura mucho más responsable en los últimos años respecto al manejo de la seguridad de la información en todas las áreas y departamentos.

Dado esto, aspectos tan importantes para cualquier institución financiera como el optar por una estrategia de evaluación constante en el manejo de riesgos informáticos, generar estrategias de defensa, aumentar la capacitación continua del personal o incluso implementar respuestas efectivas ante incidentes, se vuelven temas vitales.

Esta investigación se enfocó principalmente en analizar uno de los sistemas que integran la gama de servicios de una institución financiera activa en el país, específicamente un sistema web denominado “CREDIWEB”, donde al igual que todas las plataformas web, este se encuentra expuesto a diversas amenazas cibernéticas cada vez más evolucionadas e innovadoras, esto sin dejar de mencionar todo el crecimiento exponencial al que han sometido estos métodos de delitos cibernéticos y a la creciente sofisticación de los ataques que se generan hoy en día; Por lo tanto, el propósito de este trabajo es identificar y evaluar vulnerabilidades en la seguridad de este aplicativo con el fin de asegurar que el sitio web se encuentre con una seguridad aceptable contra posibles ataques, manteniendo durante un incidente de ciberseguridad o seguridad informática, la confidencialidad, integridad y disponibilidad de los datos y su servicio antes, durante y después del ataque, sin que se vea afectado, o en dado caso, el objetivo es identificar posibles brechas en la seguridad para que no puedan ser explotadas por ciberdelincuentes, con esto,

poder generar diversas recomendaciones correctivas para ser aplicadas el en sistema y evitar que ocurra un evento de seguridad real.

En conclusión, el proyecto de investigación resalta la necesidad de implementar una ciberseguridad sólida en el aplicativo web de esta institución, debido a que los riesgos son reales y las consecuencias pueden ser devastadoras tanto para los clientes como para la reputación de la entidad financiera, esto en caso de que se materializara algún evento. Por lo tanto, la implementación de estrategias de seguridad adecuadas y una mentalidad proactiva, puede ayudar no solo a esta institución, sino que también a muchas otras organizaciones a salvaguardar la confianza de sus clientes y garantizar la seguridad de cada una de las transacciones en línea en sus transacciones.

Frases clave: Pérdida de Control de Acceso, Inyección, Diseño Inseguro, Configuración de Seguridad Incorrecta, Componentes Vulnerables y Desactualizados, Fallas de Identificación y Autenticación, Fallas en el Software y en la Integridad de los Datos, Fallas en el Registro y Monitoreo, Falsificación de Solicitudes del Lado del Servidor (SSRF)

Capítulo 1. Introducción.

1.1 Generalidades:

A lo largo de los años, los sistemas bancarios han evolucionado significativamente y han pasado de implementar sus sistemas de gestión de la información del típico desarrollo de aplicaciones de escritorio a desarrollo de aplicaciones basadas en navegadores web.

Esto ha permitido a estas entidades poder implementar infraestructura más robusta, con mayor tolerancia a fallos y con menos uso de recursos, además de brindar beneficios a los clientes como poder acceder a sus cuentas de forma remota, sin embargo, esta nueva modalidad también ha traído nuevos desafíos, entre los cuales se presentan los riesgos de ciberseguridad, peligros ante la privacidad de datos personales o confidenciales y la potenciación de actividades fraudulentas en estos sistemas.

Las instituciones financieras ahora deben ser proactivas en sus medidas de seguridad con relación a la gestión de datos para poder garantizar una seguridad adecuada en las transacciones y transferencia de información que se realizan a diario en sus sistemas o aplicaciones web de posibles actores maliciosos.

Otro aspecto para considerar por parte de las entidades financieras es que, se debe asegurar que las aplicaciones sean accesibles para todos los colaboradores, independientemente de su experiencia técnica, por otra parte, estas también deben mantenerse actualizadas ante el panorama tecnológico de su tiempo.

Lo anterior se considera como generalidades de las aplicaciones web ya que estas están en constante riesgo. Los principales riesgos relacionados con las aplicaciones web son los riesgos de ciberseguridad, fuga, destrucción o modificación y secuestro de la información.

Estos riesgos pueden presentarse debido a debilidades en las medidas de seguridad de la aplicación, mal diseño del sistema, monitoreo, parchado o

mantenimiento inadecuado, lo que puede permitir a los delincuentes informáticos acceder a los datos confidenciales del cliente.

Otro factor para considerar son los riesgos de privacidad de los datos, los cuales pueden presentarse por métodos de cifrado obsoletos o poco seguros, lo que podría desencadenar que estos datos puedan ser decodificados exponiendo a estas empresas financieras a riesgos reputacionales.

Por último, las actividades fraudulentas se pueden presentar debido a un monitoreo inadecuado o un registro de eventos deficiente, lo que imposibilita detectar este tipo de acciones dentro del sistema de forma oportuna, permitiendo que los delincuentes usen el sistema para su propio beneficio.

1.2 Antecedentes del Problema:

Como es de conocimiento general, el cibercrimen se ha incrementado en Costa Rica de manera exponencial en los últimos años. Esto incluye ataques de phishing, ransomware, fraude en línea y robo de información personal, lo que ha llevado al país a reconsiderar un mayor enfoque en la seguridad cibernética a nivel gubernamental, corporativo e institucional.

Desde 2014, Costa Rica ha puesto en marcha una Estrategia Nacional de Seguridad Cibernética liderada por el Ministerio de Ciencia, Tecnología y Telecomunicaciones (MICITT), destinada a establecer un marco de cooperación y de coordinación entre diferentes entidades públicas y privadas, para fortalecer la seguridad cibernética del país. La estrategia incluye la creación de capacidad técnica, el fortalecimiento de la legislación y la sensibilización sobre ciberseguridad.

Sin embargo, el riesgo de los ciberataques es latente, de hecho, una muestra de esto es que, a principios de 2018, y en los años posteriores, se reportaron varios incidentes de ciberseguridad bastante agresivos que afectaron diferentes instituciones públicas, como por ejemplo el Banco de Costa Rica y el Instituto Costarricense de Electricidad, el Ministerio de Hacienda y a la CCSS, por mencionar algunos.

Estos incidentes resaltaron la importancia indiscutible de fortalecer las medidas de seguridad y concienciación en todas las organizaciones, promoviendo la educación y capacitación en este campo, para enfrentar los diversos desafíos de ciberseguridad con los que nos podremos encontrar a diario en las diferentes instituciones del país, ya que a medida que la tecnología continúa avanzando, las instituciones deben continuar mejorando sus capacidades y medidas de seguridad para enfrentar los desafíos en evolución respecto al dominio cibernético.

En resumen, la ciberseguridad es esencial para proteger los datos, prevenir ataques cibernéticos, salvaguardar la reputación, cumplir con regulaciones y proteger a los usuarios contra diferentes fraudes en línea. Para lograr esto, es indispensable adoptar prácticas y medidas de ciberseguridad robustas, con el fin de garantizar un entorno en línea seguro y confiable en cualquier institución u organización.

1.3 Definición y Descripción del Problema:

En la era digital actual en la que vivimos, las plataformas web desempeñan un papel fundamental en la comunicación, el comercio y el intercambio de información en diversos ámbitos. Sin embargo, el creciente uso de estas plataformas y de las diferentes tecnologías que se involucran en el despliegue de estas, también ha llevado a un aumento en los ciberataques y a la exposición de vulnerabilidades de seguridad que han logrado comprometer, en muchas ocasiones, la integridad, la confidencialidad y la disponibilidad de los datos dentro de las instituciones.

La necesidad constante de realizar pruebas de los sitios web en búsqueda de riesgos y vulnerabilidades se ha convertido en un tema de preocupación, tanto para las organizaciones, directivos y usuarios finales. Dado esto, a pesar de los avances tecnológicos en temas de seguridad, los ciberdelincuentes continúan encontrando formas mucho más sofisticadas de explotar las vulnerabilidades y comprometer la seguridad de las plataformas web.

El análisis práctico, en busca de mejoras para mitigar diversos riesgos y vulnerabilidades de los sistemas web, en temas de ciberseguridad, implica la evaluación exhaustiva de su infraestructura, aplicaciones, servicios y políticas de seguridad internas, entre otros. Dicho proceso evaluativo tiene como objetivo identificar y comprender las posibles vulnerabilidades y brechas de seguridad presentes en las plataformas (en caso de existir), donde, para efectos del presente documento, se centrará en una plataforma web a la cual se le denominará “CREDIWEB”, la cual pertenece a un sistema de gestión de procesos denominada BPM (Business Process Management), esto implica que esta metodología adoptó una serie de pasos o acciones sobre la forma de trabajar de la empresa y su interacción de la aplicación con los usuarios, esto con el objetivo de mejorar los procesos y facilitar la colaboración con un enfoque hacia el cliente.

Es debido a esto que el aseguramiento, ya sea del sistema mencionado anteriormente o de cualquier otra plataforma web destinada a la continuidad del servicio es indispensable, ya que se debe asegurar su correcto funcionamiento en todo momento, sin embargo, en muchos casos, las organizaciones no cuentan con los conocimientos, recursos o enfoques adecuados para llevar a cabo un análisis exhaustivo en temas de ciberseguridad.

Existe una necesidad apremiante para cualquier institución, de investigar y desarrollar metodologías efectivas para la evaluación de seguridad de las plataformas web, así como identificar y mitigar las vulnerabilidades existentes. Por lo tanto, estas metodologías deben abordar aspectos clave, como la identificación de vulnerabilidades, ya sean conocidas o desconocidas, la evaluación de la robustez de las medidas de seguridad implementadas (escudo de sistema) y la recomendación de medidas preventivas y correctivas para fortalecer la ciberseguridad de las plataformas (fortalecimiento).

En este trabajo de investigación, se propuso realizar un análisis de ciberseguridad de una plataforma web específica, con el fin de identificar las vulnerabilidades presentes y proponer soluciones para mitigar dichas vulnerabilidades. El estudio se centra en la identificación y evaluación de vulnerabilidades conocidas y desconocidas, así como en la implementación de

medidas preventivas y correctivas para mejorar la ciberseguridad de la plataforma. El objetivo final es proporcionar a la organización una guía práctica y efectiva para proteger la plataforma web contra posibles ataques cibernéticos.

1.4 Justificación:

En la actualidad la mayoría de las empresas privadas y entidades de gobierno están recurriendo al uso de aplicaciones web de forma más frecuente, ya que estas facilitan el acceso a los recursos y herramientas para los diferentes clientes internos y externos.

Estas aplicaciones de navegador o explorador web permiten reducir el costo de mantenimiento e instalación, además, ofrecen ventajas en la compatibilidad de hardware o software ya que se ejecutan a través del portal web y no de un agente preinstalado en los equipos de cómputo, lo que permite una administración y mantenimiento menos complejos.

No obstante, la implementación de este tipo de aplicaciones conlleva riesgos y “CREDIWEB”, no es la excepción a la regla y tampoco se encuentra exenta de usuarios malintencionados o ciberataques, por lo tanto, la implementación del proyecto en una entidad financiera requiere del análisis de esta plataforma para poder identificar vulnerabilidades relacionadas.

Ya que esta herramienta está diseñada para los colaboradores, pretende manejar información sensible o confidencial, lo cual expone a la entidad financiera a riesgos como la fuga, robo, destrucción o modificación de información sensible o confidencial, reprocesos administrativos, demandas, sanciones o, afectación del nivel de servicio; lo que podría llevar a la institución a grandes pérdidas financieras y una inminente pérdida de confianza de sus clientes.

Por lo anterior el equipo de trabajo se da a la tarea de realizar un análisis de ciberseguridad de la plataforma web desde un enfoque práctico, en búsqueda de mejoras para mitigar riesgos y vulnerabilidades del sistema.

Ofreciendo apoyo en la detección de brechas de seguridad, brindando en la medida de lo posible un producto más confiable y seguro en beneficio de la institución, sus colaboradores y sus clientes.

1.5 Viabilidad. Técnica, operativa y económica:

Se cuenta con la disponibilidad tanto de los recursos técnicos, cómo el acceso a la información necesaria para realizar el trabajo de investigación. A continuación, se menciona algunos de los recursos más importantes para la investigación:

Acceso a información y datos: Se cuenta con acceso a diversas fuentes de información propias del sistema, así mismo se tienen disponibles recursos digitales como los diagramas UML, manuales y documentación del proyecto, así como de diversos conjuntos de datos relevantes para la investigación.

Recursos tecnológicos: Se cuenta con la autorización y el acceso a diversas herramientas y tecnologías para poder realizar una investigación profunda de la plataforma web. Entre ellos podemos mencionar:

Editores de código e IDEs: (Visual Studio Code y Visual Studios Professional), herramientas de modelado de bases de datos (Microsoft SQL Server Management Studio), herramientas de prototipado y diseño de interfaces de usuario (UI) (Adobe XD,Miro, Visio, entre otros).

Herramientas de gestión de Software: Se cuenta con el acceso a las herramientas institucionales para la generación de informes, gestión de requisitos, gestión de proyectos, compilaciones automatizadas, pruebas, oficialmente se trata de Microsoft Azure DevOps.

Herramientas de gestión de versiones: Acceso al sistema de control de versiones como (Git) en AzureDevOs, lo cual permite acceder eficientemente a las versiones del software y el ambiente colaborativo actual de los desarrolladores.

Acceso a la plataforma web (CREDIWEB): se cuenta con el acceso a la plataforma para realizar análisis y pruebas desde el lado del cliente. Se coordina con los encargados, un usuario modelo para poder realizar la gestión de las pruebas de seguridad en el sistema, así como los permisos respectivos para dicho proceso.

En consideración de la viabilidad operativa los miembros del grupo tienen el conocimiento necesario y las habilidades para poder desarrollar un marco de análisis adecuado; interpretando la literatura existente en el campo como marcos de trabajo, artículos, foros, entre otros de forma idónea, permitiendo a estos aplicar diferentes metodologías o métodos de investigación para realizar diversos análisis de código, pruebas de penetración, revisión de código fuente. Esto permite que el equipo de trabajo pueda identificar brechas, vulnerabilidades o puntos de mejora, tanto a nivel de código fuente de la aplicación web, como en el análisis de ciberseguridad.

Por otra parte, el trabajo también se considera viable a nivel económico, debido a que este no hará incurrir a la entidad financiera en ningún de los siguientes costes:

Recurso Humano: el trabajo a desarrollar no requiere del involucramiento o participación de recurso humano adicional propio de la entidad financiera, por lo que no se deberán considerar los costos relacionados a horas extras o cualquier otro tipo de incentivo económico.

Hardware y Software (licencias): el desarrollo de esta investigación no implica la necesidad de realizar ningún tipo de adquisición de hardware o software adicional, debido a que el equipo de trabajo utilizará según lo acordado, los recursos con los que ya cuenta la institución.

Infraestructura y Seguridad: Dado que la entidad financiera ya cuenta con un ambiente de pruebas aislado, no se deberá invertir en costes de configuración relacionados a la plataforma web.

Por último, se considera importante mencionar, que el equipo de trabajo asume una postura de responsabilidad ética y moral con la entidad financiera,

lo cual permite y asegura entre ambas partes un manejo adecuado de la privacidad de los datos y aspectos de confidencialidad de la información.

1.6 Objetivo General y Específicos:

1.6.1 Objetivo General:

Desarrollar un plan de mejoras para mitigar los riesgos y las vulnerabilidades de ciberseguridad en la plataforma web de crédito en la Institución Financiera Nacional.

1.6.2 Objetivos Específicos:

- Revisar los marcos de ciberseguridad comúnmente aceptados y establecer las mejores pruebas que se deban de realizar.
- Realizar pruebas de penetración contra la plataforma web de crédito en la Institución Financiera Nacional.
- Identificar las vulnerabilidades comunes de la plataforma web de crédito en la Institución Financiera Nacional.
- Proponer un plan de implementación a las vulnerabilidades comunes basados en los hallazgos de las pruebas y análisis de vulnerabilidades realizados.

1.7 Alcances y Limitaciones:

1.7.1 Alcance:

Este trabajo está limitado únicamente al análisis e identificación de brechas de seguridad, así como la identificación de vulnerabilidades y oportunidades de mejora relacionadas a la plataforma web denominada como “CREDIWEB”.

Esto conlleva a la realización de tareas como: el análisis de vulnerabilidades comunes presentes en la plataforma web, análisis de vulnerabilidades de relacionadas al código fuente; considere esto como la

identificación del uso de bibliotecas o librerías obsoletas, configuraciones incorrectas en la plataforma web, falta de validación de datos, pruebas de inyección de código, entre otras técnicas de validación de seguridad relacionadas a las aplicaciones web.

1.7.2 Limitaciones:

La principal limitante es el uso restringido de herramientas por parte de la institución, ya que esto podría exponer a la entidad financiera a posibles riesgos relacionados a la exposición de datos, por este motivo no se utilizan herramientas de carácter invasivo.

1.8 Marco de Referencia. Organizacional y Socioeconómico:

El sistema financiero en Costa Rica está compuesto por una variedad de instituciones y participantes que trabajan en conjunto para facilitar el flujo de capital, es decir, son las encargadas de generar, captar, administrar y dirigir el ahorro, además de brindar diversos servicios financieros tanto a los individuos como a las empresas y entidades del país.

El sistema financiero está supervisado por diferentes legislaciones y órganos que regulan las transacciones de activos financieros, entre las entidades que lo regulan podemos nombrar:

El Banco Central de Costa Rica (BCCR), quien es la entidad encargada de regular la política monetaria y velar por la estabilidad del sistema financiero, remite la moneda nacional (colón costarricense) y establece las tasas de interés de referencia.

La Superintendencia General de Entidades Financieras (SUGEF): Es el ente regulador encargado de supervisar y regular todas las instituciones financieras en Costa Rica. Su objetivo principal es asegurar la solidez y transparencia del sistema financiero, proteger los derechos de los consumidores y prevenir actividades ilícitas, como el lavado de dinero.

Bolsa Nacional de Valores (BNV): Es el mercado de valores de Costa Rica, donde se realizan transacciones de compra y venta de valores, como acciones y bonos. La BNV proporciona un espacio para la inversión y financiamiento de empresas a través de la emisión de valores.

Comisión Nacional de Valores (CONAV): Es el organismo encargado de regular y supervisar el mercado de valores en Costa Rica. La CONAV vela por la protección de los inversionistas, promueve la transparencia y el buen funcionamiento del mercado y aprueba la emisión de valores.

Regulaciones y Legislación: El sistema financiero en Costa Rica está respaldado por un marco legal y regulador que establece las normas y requisitos para la operación de las instituciones financieras y la protección de los usuarios. Algunas leyes importantes incluyen la Ley Orgánica del Banco Central, la Ley Reguladora del Mercado de Valores y la Ley General de Entidades Financieras.

Además, el sistema financiero costarricense se divide en dos grandes grupos: el Sistema Financiero Nacional (**SFN**) y el Sistema Financiero Internacional (**SFI**). El SFN está compuesto por las entidades financieras que operan en el país y que están reguladas por los entes anteriormente mencionados, entre estas entidades podemos mencionar:

Bancos Comerciales: Son entidades que ofrecen una amplia gama de servicios financieros, como cuentas de ahorro, cuentas corrientes, préstamos, tarjetas de crédito, entre otros. Algunos de los bancos comerciales más importantes en el país son el Banco Nacional de Costa Rica, el Banco de Costa Rica y el Banco Popular y de Desarrollo Comunal.

Cooperativas de Ahorro y Crédito: Son instituciones financieras cooperativas que brindan servicios similares a los bancos comerciales, pero con una estructura de propiedad y gobierno diferente. Las cooperativas de ahorro y crédito tienen un enfoque en servir a sus miembros y a menudo ofrecen tasas de interés más competitivas.

Instituciones Financieras No Bancarias: Además de los bancos y las cooperativas de ahorro y crédito, hay otras instituciones financieras no

bancarias en Costa Rica. Estas incluyen compañías de seguros, administradoras de pensiones, sociedades de inversión y empresas emisoras de tarjetas de crédito.

Por último, es importante destacar que el sistema financiero costarricense enfrenta diversos riesgos que pueden afectar su estabilidad y funcionamiento adecuado.

Entre ellos, se pueden mencionar los riesgos crediticios que suceden cuando los prestatarios no cumplen con sus obligaciones, el riesgo de liquidez que pueden generar las instituciones financieras, diversos riesgos de mercado, riesgos de cumplimiento y legalidad, entre otros, sin embargo, como parte de este trabajo de investigación, se hace énfasis en el riesgo operativo que se asocia a diferentes actividades internas de una institución financiera, y que pueden ser errores en los procesos, fraudes, **fallas tecnológicas**, desastres naturales u otros eventos que puedan interrumpir o afectar su funcionamiento normal, afectando la continuidad del negocio.

Estos riesgos pueden generar incontables pérdidas financieras y dañar drásticamente la reputación e imagen de cualquier institución sin importar su nombre o su posición en el mercado.

1.9 Estado de la Cuestión:

Si bien es cierto, el concepto de ciberseguridad en las empresas se refiere a las buenas prácticas y diversas medidas que se implementan con el fin de proteger los diferentes sistemas, redes, datos y activos digitales de una organización contra amenazas cibernéticas. La ciberseguridad va más allá y es fundamental en el entorno institucional financiero actual, ya que las instituciones dependen cada vez más de la tecnología y la conectividad para gestionar sus operaciones y el manejo de información sensible en cada transacción.

La ciberseguridad genera un enfoque holístico que requiere una combinación de tecnología, políticas y buenas prácticas de seguridad, así como una mentalidad proactiva para poder identificar y mitigar los riesgos

cibernéticos. Para mitigar este riesgo es necesario implementar medidas de ciberseguridad sólidas y contar con un personal capacitado para hacer frente a diferentes recomendaciones de seguridad, solo así las empresas pueden administrar el riesgo y proteger su información confidencial, salvaguardar su reputación, mantener la continuidad de sus transacciones en un entorno digital y continuar con sus operaciones diarias sin problemas.

La gestión y administración de sitios o páginas web para realizar transacciones financieras debe manejarse con mucho cuidado, ya que a nivel nacional e internacional nos enfrentamos a un aumento de amenazas y continuos ataques cibernéticos dirigidos a estos sitios en todo el mundo, de la misma manera, se puede ver en diversos medios, que una de las actividades económicas con mayor cantidad de ataques es el sector financiero, especialmente las plataformas tecnológicas de estas instituciones, dicho esto, y adentrándose al tema principal de la presente investigación, se procede a examinar diversos marcos de referencias, guías e investigaciones relacionadas con la protección de los sitios web contra amenazas cibernéticas, ya que a pesar de que existen muchos estándares y normas a nivel internacional, las instituciones carecen, en muchas ocasiones, de una guía metodológica específica, la cual permite que estas empresas puedan blindarse o aplicar de forma adecuada, diversas medidas ajustadas a su entorno y organización, lo cual pueda ayudar a enfrentar las amenazas y ataques que sufren estas empresas del sector financiero nacional.

Para esta investigación, se considera como una de las claves principales evaluar de forma rigurosa el sistema web de la institución financiera seleccionada, especialmente en búsqueda de diferentes amenazas que puedan exponerla, así como los métodos de protección, administración y la forma cómo gestionar las posibles amenazas y desafíos actuales con lo que se pueda enfrentar en el tema de Ciberseguridad.

Es debido a lo expuesto, que en este proyecto de investigación se pretende realizar una evaluación del sistema “CREDIWEB”, el cual pertenece a un sistema de gestión de procesos, mediante la ejecución de una serie de guías y marcos de referencia, con el fin de poder identificar las vulnerabilidades presentes y proponer soluciones para mitigar dichas vulnerabilidades.

1.9.1 Planificación de la revisión.

Se realiza una búsqueda de la documentación existente sobre el objeto de estudio, con el fin de explorar y conocer los aportes académicos que se hayan emitido al respecto, identificando, las áreas débiles y oportunidades de mejora. Finalmente, y una vez realizado el análisis de dicha documentación, se procede a desarrollar una serie de recomendaciones que sea simple, flexible y de fácil implementación.

1.9.2 Formulación de la pregunta

Con el objetivo de encontrar respuestas que demuestren el aporte de información e investigación de este trabajo, se realiza la formulación de la pregunta, la cual ayudará a delimitar los esfuerzos de búsqueda de información.

1.9.2.1 Enfoque del a pregunta

Se determina para esta investigación, que la búsqueda de información se debe realizar con base en documentos técnicos que detallan el uso y aplicación del marco para la mejora de la seguridad del sistema web “CREDIWEB”.

1.9.2.2 Amplitud y calidad de la pregunta

En esa sección se define la pregunta de investigación que se desea responder de forma clara y concisa, basados en un problema a resolver. Para la cual se realiza un listado de términos clave relevantes para la búsqueda de información y se consideran componentes fundamentales como son la población específica, capacidad técnica y guías de interés. Se definen medidas

a utilizar para medir el efecto con base en la pregunta a responder y el diseño de los estudios.

1.9.3 Problema:

Dado el alto nivel de especialización que se requiere, tanto en materia de continuidad de negocios, como en tecnología, las empresas se enfrentan a un gran reto de defenderse efectivamente de ataques cibernéticos.

1.9.4 Pregunta

Dado el problema mencionado con anterioridad, se plantea la siguiente pregunta de investigación:

¿Cuenta el sistema web “CREDIWEB” con una adecuada defensa de seguridad integral para hacerle frente a las vulnerabilidades más comunes, esto según los ciberataques más comunes detectados a nivel mundial?

1.9.5 Palabras claves y sinónimos.

Se realiza un listado de palabras claves que se van a utilizar para la búsqueda e identificación de documentos y trabajos relacionados con la investigación. En su gran mayoría, estas palabras están en el idioma inglés debido a que existe una gran cantidad de trabajos relaciones y publicaciones en este idioma, las cuales se muestran en la tabla 1.

Tabla 1: Listado de palabras

Palabra	Equivalente en Ingles
Evaluación	Evaluation
Riesgo	Risk

Modelo	Model
Autorización	Authorization
Servicios web	Web services
Desarrolladores de Software	Software Developers
Especialistas de Seguridad Informática	Information Security Specialists
Herramientas de Software	Software Tools
Amenaza	threats
Objetivos de las Pruebas	Test Objectives
Requerimientos de Seguridad	Security requirements
Contramedidas	countermeasures
Diccionario	Dictionary
Control de Acceso	Access control
Cuenta de usuario	User account
Ciberseguridad	Cybersecurity
Marco de referencia	Framework
NIST (Instituto Nacional de Normas y Tecnología)	NIST (National Institute of Standards and Technology)
Identificación de puntos de entrada de la aplicación	Identification of application entry points
Análisis de códigos de error	Error code analysis
Gestión de configuración	configuration management
Fuerza bruta	Brute force
Autenticación	Authentication
Galletas	Cookies
Escalación de privilegios	privilege escalation

Validación de datos	Data validation
Secuencias de comandos entre sitios	Cross site scripting
Inyección SQL	Injection SQL
Inyección de código	code injection
desbordamiento de búfer	buffer overflow
Cadenas de formato	format strings
HTTP (Protocolo de transferencia de hipertexto)	HTTP (Hypertext Transport Protocol)
HTTPS (Protocolo de transferencia de hipertexto seguro)	HTTPS (Hypertext Transport Protocol Secure)
Denegación de servicio	Denial of service
código fuente	Source code
Intrusión	Intrusion
criptografía	Cryptography
Pruebas de Seguridad, Pruebas de penetración	Security Tests, Penetration Testing
Cifrado	Encryption
Vulnerabilidad	Vulnerability
Hacking web	Hacking web
Interfaz de Programación de Aplicaciones	Application Programming Interface - api
caza de insectos	bug bounting hunting
integridad	integrity
disponibilidad	availability
confiabilidad	reliability
SAST	static application security test

DAST	dynamic application security test
Caja blanca	White box
caja negra	black box
caja gris	White box
Enumeración de debilidades comunes (CWE)	Common Weakness Enumeration (CWE)
Vulnerabilidades y exposiciones comunes (CVE)	Common Vulnerabilities and Exposures (CVE)
Sistema de puntuación de vulnerabilidad común (CVSS)	Common Vulnerability Scoring System (CVSS)

(Pérez y Molina. 2023. Lista de palabras).

1.9.6 Intervención

Ver los resultados de cómo la aplicación de guías y metodologías para la mejora de la seguridad del sistema CREDIWEB contribuye a aumentar su defensa en ciberseguridad.

1.9.7 Control

Al iniciar la investigación, no se cuenta con ninguna base de información. Se empieza con una búsqueda de cero a partir de las palabras clave definidas.

1.9.8 Efectos

Se espera tener documentación suficiente con las búsquedas realizadas para entender cuáles guías y metodologías existen, con el fin de realizar una evaluación del sistema CREDIWEB en el ámbito de la ciberseguridad.

Para la documentación encontrada se realiza una revisión de la calidad de ésta en sitios web especializados para tal fin.

1.9.9 Aplicación

Esta investigación es de gran utilidad para todo el departamento de Tecnologías de la Información de la institución financiera, como por ejemplo los desarrolladores e ingenieros a cargo de administrar los sistemas, así como también administradores de bases de datos, quienes deben resguardar el activo más valioso de una organización, y que deseen realizar una evaluación de su estado actual de resiliencia y ciberseguridad.

1.9.10 Diseño experimental

Durante el diseño experimental se realiza un análisis y clasificación de los estudios obtenidos basándose en la calidad del contenido y relevancia para la investigación. Con lo anterior, se garantiza, no solo contar con la documentación de confianza para la investigación, sino también la suficiente para evitar tener un rango muy amplio de estudios que pudieran generar resultados no deseados.

1.10 Selección de fuentes:

Autor: Ángel Lenin Salgado Yánez.
Año: 2014.
Institución: Universidad de las Fuerzas Armadas.
Título: Análisis de las aplicaciones web de la Superintendencia de Bancos y Seguros, utilizando las recomendaciones Top Ten de OWASP para determinar los riesgos más críticos de seguridad e implementar buenas prácticas de seguridad para el desarrollo de sus aplicativos.
Descripción: Tiene como objetivo el análisis de riesgos de las aplicaciones web utilizando las recomendaciones OWASP Top 10 – 2010 para descubrir las

vulnerabilidades que se presentan durante el desarrollo de un software y estimar el riesgo asociado para el negocio. A partir de los resultados obtenidos donde se identificaron la ocurrencia de almacenamiento criptográfico inseguro y protección insuficiente en la capa de transporte se realizó una propuesta de buenas prácticas para asegurar las aplicaciones, corregir los riesgos detectados y asegurar el proceso de desarrollo de nuevas funcionalidades y existentes.

Autor: Rubén Jorge Fusario.
Año: 2017.
Institución: Universidad de Buenos Aires.
Título: Vulnerabilidades en la Seguridad de las Transacciones Interactivas de Comercio Electrónico a través de la Web.
Descripción: Este trabajo de tesis investiga las vulnerabilidades en la seguridad de las transacciones interactivas de comercio electrónico trazable a través de la web y pretende responder el siguiente interrogante; ¿Qué componente del sistema de comercio electrónico trazable presenta mayor nivel de vulnerabilidad para la seguridad del sistema, y cuál es el orden de importancia de los factores que afectan dicha seguridad, según la evaluación efectuada por futuros profesionales de TIC?

Autor: Edderson Jair Hernández Mechate.
Año: 2020.
Institución: Universidad Andina del Cusco.
Título: Vulnerabilidades Informáticas en el Portal Web de la Universidad Andina del Cusco.
Descripción: se desarrolló teniendo como objeto de pruebas el portal Web de la Universidad Andina del Cusco (www.uandina.edu.pe); el punto de partida este dado en la afirmación de que no se puede garantizar la seguridad o la estabilidad total de un sistema informático; es así cómo se desarrolló persiguiendo el objetivo de identificar vulnerabilidades informáticas mediante la aplicación de las herramientas de detección de vulnerabilidades en el portal Web de la Universidad Andina del Cusco.

Capítulo 2. Marco Conceptual

A continuación, se presentan los conceptos relacionados con el presente trabajo de investigación; estos conceptos pretenden estandarizar el uso de estos para que no exista interpretación diferenciada entre los lectores.

2.1 Conceptos Generales:

2.1.1 Amenaza: Las amenazas en el ámbito de la ciberseguridad deben considerarse como una acción que puede ser ejercida mediante la explotación de vulnerabilidad o engaños con el fin de acceder a información confidencial o recursos de un sujeto u objeto.

2.1.2 Autorización: La Real Academia Española (RAE) define la autorización como “Acto de una autoridad por el cual se permite a alguien una actuación en otro caso prohibida” (Real Academia Española. 2023. Diccionarios. https://www.rae.es/).

2.1.3 Contramedida: Las contramedidas son parte de la gestión relacionadas a la seguridad de la información con el fin de reducir el riesgo de la ocurrencia relacionadas con incidentes de seguridad.

2.1.4 Control de Acceso: Según Microsoft una definición adecuada para el control de acceso seria la siguiente: “El control de acceso es un elemento esencial de la seguridad que determina quién tiene permiso para tener acceso a determinados datos, aplicaciones y recursos y en qué circunstancia” (Microsoft, 2023).

2.1.5 Cuenta de usuario: IBM define las cuentas de usuario como “Las cuentas de usuario se definen por grupos, permisos y políticas de contraseña para ayudar a proporcionar un entorno seguro. Este tipo de definición de cuenta de usuario se define como un modelo de seguridad basado en rol” (IBM, 2023).

2.1.6 Desarrolladores de Software: IBM (2023) en su artículo "¿Qué es el desarrollo de software?", define a los desarrolladores de software como personas que tiene un rol de participación en áreas específicas de proyectos, los cuales brindan su conocimiento mediante la escritura de código para la creación de requisitos funcionales, pueblas y mantenimiento de aplicaciones. (IBM, 2023).

2.1.7 Diccionario: La Real Academia Española (RAE) define la palabra diccionario como el "Repertorio en forma de libro o en soporte electrónico en el que se recogen, según un orden determinado, las palabras o expresiones de una o más lenguas, o de una materia concreta, acompañadas de su definición, equivalencia o explicación." (Real Academia Española, 2023).

2.1.8 Especialistas de Seguridad Informática: Se puede considerar como una persona especializada en el campo de la tecnología, el cual tiene como responsabilidad proteger la información de las empresas a las cuales presta servicios.

2.1.9 Evaluación: La Real Academia Española (RAE) define evaluación como la acción o efecto de evaluar (Real Academia Española, 2023), esto se puede interpretar como la acción de ejecutar pruebas para medir el nivel de cumplimiento de un determinado evento, objeto o actividad.

2.1.10 Herramientas de Software: La Real Academia Española (RAE) define software como un "Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora" (Real Academia Española, 2023), por otra parte, define la palabra herramienta de la siguiente manera: "Instrumento, por lo común de hierro o acero, con que trabajan los artesanos" (Real Academia Española, 2023), por esto, se puede decir que una herramienta de software puede

considerarse como una aplicación o un programa de computadora que puede ejecutar una serie de tareas definidas.

2.1.11 Modelo: M. Felicísimo, en 1997 presenta en su publicación “Curso sobre Modelos Digitales del Terreno” dos definiciones de modelos, la primera definición: un modelo es "una representación simplificada de la realidad en la que aparecen algunas de sus propiedades" (Joly, 1988:111). La segunda definición: "un modelo es un objeto, concepto o conjunto de relaciones que se utiliza para representar y estudiar de forma simple y comprensible una porción de la realidad empírica" (Ríos, 1995:23). (M. Felicísimo. 1997).

2.1.12 Objetivos de las Pruebas: La Real Academia Española (RAE) define Objetivo como “Fin o intento a que se dirige o encamina una acción u operación” (Real Academia Española, 2023). Por otra parte, también se puede encontrar la definición de prueba como el “Ensayo o experimento que se hace de algo, para saber cómo resultará en su forma definitiva” (Real Academia Española, 2023).

2.1.13 Requerimientos de Seguridad: Pérez Porto, J., Gardey, A. (2014) definen como un requisito “una condición o una circunstancia necesaria para cumplir un cierto objetivo o para obtener un resultado determinado”, además, la Real Academia Española (RAE), define seguridad como “Cualidad de seguro” (Real Academia Española, 2023).

2.1.14 Riesgo: Según la definición de Ambit (2023), se puede considerar el riesgo como “… la posibilidad de que un sistema sufra un incidente de seguridad y que una amenaza se materialice causando una serie de daños”. También debe considerarse como riesgo la definición publicada por la Escuela Europea de la excelencia (2023) en su blog **“ISO 31000. Términos y definiciones”** el cual lo define como el “Efecto de la incertidumbre sobre nuestros objetivos.”

2.1.15 Servicios web: La Real Academia Española (RAE) (2023) reconoce el término web como un “sustantivo femenino, escrito con mayúscula inicial, designa, por abreviación de la expresión inglesa World Wide Web, la red mundial de comunicaciones denominada Internet”, también considera esta definición “Como adjetivo significa 'de la Red o de Internet'. Se usa normalmente en la expresión página web, que significa 'documento de Internet, al que se accede mediante enlaces de hipertexto”. La RAE (2023) también define la palabra servicio como “Acción y efecto de servir”.

2.2 Conceptos Técnicos:

2.1.16 Análisis de códigos de error: Según la definición de la Wikipedia (2023), un código de error puede entenderse como “mensajes numerados que corresponden a errores en una aplicación específica”.

2.2.1 Autenticación: La RAE (2023) define el término como “Certificar la autenticidad de algo, especialmente un documento”.

2.2.2 Cadenas de formato: La cadena de formato existe a nivel de ataque, Owasp, define este tipo de ataque como la evaluación de datos enviados a través de una cadena de código en el cual se puede generar un fallo o modificación que modifique el comportamiento esperado. (Owasp, 2023).

2.2.3 Caja blanca: Es un término empleado para las pruebas de aplicaciones o donde se tiene acceso al código fuente. (zaptest, 2023-1).

2.2.4 Caja gris: Este término acoge tanto la definición mencionada para caja blanca, como el de la caja negra, ya que fusiona ambos contextos en la ejecución de pruebas. (zaptest, 2023-2).

2.2.5 Caja negra: Las pruebas de caja negra están basadas en la simulación de un atacante al sistema o aplicativo, ya que los encargados de realizar dicha prueba, no tiene información, más que la ofrecida de forma libre en internet para poder realizar el análisis mencionado. (zaptest, 2023-3).

2.2.6 Caza de fallos: También conocida como casería de amenazas es una actividad relacionada al análisis de software y hardware con la intención de identificar amenazas y cobrar por una remuneración por esto. (Orange, 2022).

2.2.7 Ciberseguridad: El enclave define el término como “Aplicación de medidas de seguridad para proteger los diferentes componentes de los sistemas de información y comunicaciones de un ciberataque”. (Enclave de Ciencia, 2023).

2.2.8 Cifrado: La RAE (2023) define el cifrado como “Transcribir en guarismos, letras o símbolos, de acuerdo con una clave, un mensaje o texto cuyo contenido se quiere proteger”.

2.2.9 Código fuente: Es un conjunto de instrucciones codificadas, las cuales dan origen a las aplicaciones o programas computacionales, estas instrucciones se ejecutan a través de un compilador de código.

2.2.10 Confiabilidad: El término según Ostec (2023) se puede definir como “no de los pilares fundamentales de la seguridad de la información según la norma ISO 27001. Este pilar tiene como objetivo garantizar que la información sea accesible solo para personas autorizadas, protegiéndola contra la divulgación no autorizada y el uso indebido”.

2.2.11 Criptografía: La RAE (2023) define el término como el “Arte de escribir con clave secreta o de un modo enigmático”.

2.2.12 Denegación de servicio: esto es conocido con sus siglas “DDoS” y consiste en un ataque distribuido, el cual tiene como objetivo sobrecargar los servidores que atienden peticiones con relación al sistema o aplicación con el fin de botar el servicio. (Akamai, 2023). **Desbordamiento de búfer:** También conocido como un desbordamiento de memoria; tiene como objetivo exceder los recursos asignados por el sistema operativo, si esto es permitido, el atacante podrá ejecutar código arbitrario mediante la modificación de las cabeceras de las peticiones realizadas al servidor de aplicación. (Pérez, 2014).

2.2.13 Disponibilidad: Insitech en su sitio web define el término en relación con lo mencionado en ITILv4 de la siguiente manera: “De acuerdo con ITIL 4, la disponibilidad es “la capacidad de un servicio de TI u otro elemento de soporte para realizar sus funciones cuando sea necesario”. (Insitech, 2022).

2.2.14 Enumeración de debilidades comunes (CWE): Esta es una categoría que enumera las vulnerabilidades de software de forma genérica. (Wikipedia, 2023).

2.2.15 Escalación de privilegios: La definición encontrada en la Wikipedia describe adecuadamente el término definiéndolo como “"La escalada de privilegios es el acto de explotar un error, un fallo de diseño o una supervisión de la configuración en un sistema operativo o una aplicación de software para obtener un acceso elevado a los recursos que normalmente están protegidos de una aplicación o un usuario. El resultado es que una aplicación con más privilegios de los previstos por el desarrollador de la aplicación o el administrador del sistema puede realizar acciones no autorizadas". (Wikipedia. 2023).

2.2.16 Fuerza bruta: Es una forma de obtener acceso de manera forzada que tiene como objetivo ingresar a una aplicación o sistema por la fuerza,

otra forma de interpretar esta definición es la siguiente: “forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso. Dicho de otro modo, define al procedimiento por el cual a partir del conocimiento del algoritmo de cifrado empleado y de un par texto claro/texto cifrado, se realiza el cifrado (respectivamente, descifrado) de uno de los miembros del par con cada una de las posibles combinaciones de clave, hasta obtener el otro miembro del par”. (Wikipedia, 2021).

2.2.17 Galletas: La RAE (2023) define el término con vocablo de la lengua inglesa el cual se debe interpretar de la siguiente manera: “cookie: 1. Voz inglesa empleada en informática con el sentido de 'pequeño archivo de datos que queda instalado en el disco duro de un ordenador cuando este accede a una página web'. Por tratarse de un extranjerismo crudo, debe escribirse con resalte tipográfico”.

2.2.18 Gestión de configuración: “La gestión de la configuración es un proceso de ingeniería de sistemas que sirve para establecer la coherencia de los atributos de un producto a lo largo de su vida. En el mundo de la tecnología, la gestión de la configuración es un proceso de gestión de TI que supervisa los elementos de configuración individuales de un sistema de TI”. (Buchanan, 2023).

2.2.19 Hacking web: Este término se debe interpretar más como un conjunto de pruebas que tiene la intención de medir y probar qué tan segura es una aplicación, servicio o infraestructura tecnológica ante la amenaza de un ciberataque, por otra parte, el blog de “KeepCoding” lo define de la siguiente manera: “El hacking web consiste en poner a prueba los sistemas de seguridad informática de una aplicación. De este modo, es posible saber si existen fallos de seguridad, cómo podrían explotarse y qué soluciones se deberían aplicar. El hacking ético de páginas web es uno de los servicios más demandados en el área de ciberseguridad,

debido a que la mayoría de los softwares que se han desarrollado hasta el momento están en este formato". (KeepCoding, 2022).

2.2.20 HTTP (Protocolo de transferencia de hipertexto): Es el protocolo en el cual se basa la red mundial o "internet", este es un protocolo de transferencia de información, que mantiene una estructura tipo cliente servidor, este protocolo trabaja en el número de puerto 80 y sobre la pila "TCP/IP". Una definición más técnica puede ser la mencionada en la página de "Oracle" la cual lo define como "un protocolo (un conjunto de reglas que describen cómo se intercambia información en una red) que permite que un navegador web y un servidor web "hablen" entre sí utilizando el alfabeto ISO Latin1, que es ASCII". (Oracle, 2004.).

2.2.21 HTTPS (Protocolo de transferencia de hipertexto seguro): Es la versión segura de HTTP, este transfiere información mediante el puerto número 443.

2.2.22 Identificación de puntos de entrada de la aplicación: Se puede definir como el escaneo o análisis de todos los receptores o emisores de información que tiene disponible un sistema o aplicativo por los cuales se puede intentar alterar su comportamiento con un fin específico.

2.2.23 Instituto Nacional de Normas y Tecnología (NIST): "NIST es el Instituto Nacional de Estándares y Tecnología del Departamento de Comercio de los Estados Unidos. El marco de seguridad cibernética del NIST ayuda a las empresas de todos los tamaños a comprender, administrar y reducir mejor su riesgo de seguridad cibernética y proteger sus redes y datos. El Marco es voluntario. Le brinda a su empresa un resumen de las mejores prácticas para ayudarlo a decidir dónde concentrar su tiempo y dinero para la protección de la seguridad cibernética.". (FTC en español, 2023).

2.2.24 Integridad: Iso27000.es define la integridad de la siguiente manera: “Propiedad de la información relativa a su exactitud y completitud”. (ISO27000.es, 2005).

2.2.25 Interfaz de Programación de Aplicaciones: La página de Definición propone el término como: “API es una sigla que procede de la lengua inglesa y que alude a la expresión “Application Programming Interface” (cuya traducción es Interfaz de Programación de Aplicaciones). El concepto hace referencia a los procesos, las funciones y los métodos que brinda una determinada biblioteca de programación a modo de capa de abstracción para que sea empleada por otro programa informático”. (Pérez Porto, J., Gardey, 2015).

2.2.26 Intrusión: "Acción de introducirse de forma indebida o ilegal en una propiedad”. (Gran Diccionario de la Lengua Española, 2016). La RAE (2023) define el término cómo el “Apropiarse, sin razón ni derecho, de un cargo, una autoridad, una jurisdicción, etc.”.

2.2.27 Inyección de código: “Un defecto de inyección de código ocurre cuando es posible enviar datos inesperados a un intérprete. Estos son muy comunes en código antiguo. Se encuentran frecuentemente en consultas SQL, LDAP, Xpath o NoSQL; comandos de sistema operativo; analizadores sintácticos de XML; cabeceras SMTP; parámetros de funciones; etc. Estos defectos son fáciles de encontrar cuando se examina el código, sin embargo, son difíciles de descubrir mediante pruebas funcionales.” (Wikipedia, 2023).

2.2.28 Inyección SQL: “es un tipo de vulnerabilidad en la que un atacante usa un trozo de código SQL (lenguaje de consulta estructurado) para manipular una base de datos y acceder a información potencialmente valiosa. Es uno de los tipos de ataques más frecuentes y amenazadores,

ya que puede atacar prácticamente cualquier sitio o aplicación web que use una base de datos basada en SQL". (Kaspersky, 2023).

2.2.29 Marco de referencia: Según la definición de enclave, un marco de referencia se define de la siguiente manera: "Framework: Informática Conjunto estandarizado de criterios, prácticos, conceptos y herramientas para abordar una problemática particular". (Enclave de ciencia, 2023).

2.2.30 Prueba de seguridad de aplicaciones dinámicas (DAST): Pathak en la página web de "GeekFlare" define el término como: "otro método de prueba que utiliza un enfoque de caja negra, suponiendo que los probadores no tienen acceso o conocimiento del código fuente de la aplicación o su funcionalidad interna. Prueban la aplicación desde el exterior utilizando las salidas y entradas disponibles. La prueba se asemeja a un pirata informático tratando de acceder a la aplicación). (Amrita Pathak, 2022).

2.2.31 Prueba de seguridad de aplicaciones estáticas (SAST): Pathak en la página web de "GeekFlare" define el término como: "un método de prueba para asegurar una aplicación al revisar su código fuente estadísticamente para identificar todas las fuentes de vulnerabilidad, incluidas las debilidades y fallas de la aplicación como inyección SQL". (Amrita Pathak, 2022).

2.2.32 Pruebas de Seguridad o Pruebas de penetración: La página web de Distillery define las pruebas de penetración como: "Las pruebas de seguridad de aplicaciones (AST) son un proceso de identificación, análisis y corrección de las vulnerabilidades de seguridad de una aplicación web. Incluye probar la aplicación para detectar vulnerabilidades conocidas y examinar el código para detectar posibles problemas de seguridad. El proceso consiste en probar el código de la aplicación y su entorno para detectar fallos de seguridad y posibles

vulnerabilidades. Una vez identificados, los problemas se abordan y solucionan". (Distillery, 2023).

2.2.33 Secuencias de comandos entre sitios (XSS): Según la definición de Báez se puede interpretar el término como: "un tipo de ataque que aprovecha fallas de seguridad en sitios web y que permite a los atacantes implantar scripts maliciosos en un sitio web legítimo (también víctima del atacante) para ejecutar un script en el navegador de un usuario desprevenido que visita dicho sitio y afectarlo, ya sea robando credenciales, redirigiendo al usuario a otro sitio malicioso". (Báez, 2021).

2.2.34 Sistema de puntuación de vulnerabilidad común (CVSS): La definición que ofrece IBM sobre el término es la siguiente: "El sistema CVSS (Common Vulnerability Scoring System) se utiliza para evaluar la gravedad y el riesgo de la seguridad del sistema informático". (IBM, 2023).

2.2.35 Validación de datos: Según la definición ofrecida por la Wikipedia la validación de datos se puede interpretar como: "En las ciencias de la computación, validación de datos es el proceso de asegurar que un programa funcione en datos limpios, correctos y útiles. Utiliza rutinas, a menudo llamadas "reglas de validación" "restricciones de validación" o "rutinas de comprobación", que comprueban la corrección, significación y seguridad de los datos que se introducen en el sistema. Las reglas pueden implementarse a través de las instalaciones automatizadas de un diccionario de datos, o mediante la inclusión de una lógica de validación explícita". (Wikipedia, 2021).

2.2.36 Vulnerabilidad: Según lo publicado en la web de Ambit, "Se considera una vulnerabilidad a una debilidad propia de un sistema que permite ser atacado y recibir un daño. Las vulnerabilidades se producen de forma habitual por una baja protección contra ataques externos, falta de

actualizaciones, fallos de programación, y otras causas similares". (Ambit, 2023).

2.2.37 Vulnerabilidades y exposiciones comunes (CVE): La CVE o "Common Vulnerabilities and Exposures (CVE, traducción: «Vulnerabilidades y exposiciones comunes»), es una lista de información registrada sobre vulnerabilidades de seguridad conocidas, en la que cada referencia tiene un número de identificación CVE-ID, descripción de la vulnerabilidad, que versiones del software están afectadas, posible solución al fallo (si existe) o como configurar para mitigar la vulnerabilidad y referencias a publicaciones o entradas de foros o blog donde se ha hecho pública la vulnerabilidad o se demuestra su explotación. Además, suele también mostrarse un enlace directo a la información de la base de datos de vulnerabilidades del NIST (NVD), en la que pueden conseguirse más detalles de la vulnerabilidad y su valoración". (Wikipedia, 2023).

Capítulo 3. Marco Metodológico

3.1 Tipo de Investigación:

El propósito del presente proyecto se enfoca en resolver un problema práctico y emplear los conocimientos teóricos en una aplicación de una institución financiera. Con el fin de proporcionar una solución concreta y directamente aplicable a esta herramienta. Por esto, se considera esta investigación aplicada.

3.2 Alcance Investigativo:

Esta investigación tuvo un alcance de tipo explicativo, el cual va más allá de las descripciones y busca establecer las causas de los eventos o fenómenos, además, pretende generar mayor entendimiento de estos por medio de la identificación de variables, uso de metodologías, análisis y discusión, por último, su alcance está claramente delimitado en la investigación.

3.3 Enfoque:

El enfoque fue cualitativo y siguió un método de inferencia de resultados llamado “ideográfico”, en el cual no se comprueban leyes universales. Se usa el método inductivo (de lo específico a lo general). Esto marca una diferencia muy grande en cuanto al nivel de generalización que se puede alcanzar, más parecido a las posibilidades que al alto umbral de certeza del enfoque cuantitativo. El paradigma base es el naturalismo.

3.4 Diseño:

El diseño de la investigación se definió con la siguiente serie de pasos:

- Para identificar los procedimientos que se utilizaron en el análisis de vulnerabilidades se realizó una entrevista al técnico experto encargado.
- Se aplicó un escaneo de vulnerabilidades a la muestra que se seleccionó para la investigación.

- Se interpretaron los resultados.
- Se identificaron las oportunidades de mejora y puntos fuertes de la metodología empleada.
- Se realizo un análisis final con sus conclusiones.
- Se brindaron las recomendaciones correspondientes a los interesados.

3.5 Población y Muestreo:

En este proyecto se entendió como población el sistema “CREDIWEB”, el cual pertenece a una institución financiera nacional, donde se generaron pruebas de penetración a este único sistema, dado esto, no hubo muestreo.

3.5.1 Instrumentos de Recolección de Datos

Los instrumentos de recolección de datos permiten conseguir datos “crudos”, que deberán ser analizados, para fines estadísticos como los conteos y mediciones, o bien para ser sujetos de un proceso de interpretación. En este caso, los instrumentos de evaluación utilizados fueron los siguientes:

Herramienta y Scanner de Detección de Vulnerabilidades:

Estas herramientas permitieron analizar objetivos específicos con el fin de encontrar vulnerabilidades de todo tipo de criticidad, configuraciones incorrectas en los equipos, contraseñas predeterminadas establecidas que sean comunes o ausentes en algunas cuentas del sistema, entre otros. Además de emitir reportes con el análisis correspondiente y algunas recomendaciones.

Para efecto de esta investigación se utilizaron las siguientes herramientas:

- “Burp Suite” de la compañía “PortSwigger” la cual actúa como un proxy para la intercepción de tráfico a través de la red de datos, además como se utilizó una versión de prueba por 30 días, se tuvieron a disposición las características de escaneos automatizados y auditoría.

- “Zen Attack Proxy (ZAP)” del Proyecto “OWASP Foundation”, la cual permite buscar dentro de aplicaciones web vulnerabilidades basadas en el “Top 10” de la fundación.

- “Nikto” del proyecto puesto a disposición en “GitHub”, esto en el perfil del alias “Sullo", este proyecto fue escrito por Chris Sullo y David Lodge y se puede descargar desde su sitio “https://cirt.net/Nikto2”. Esta es una herramienta para escaneos de servidores web, este instrumento es de código abierto, basada en la licencia “GPL”, por lo que permite realizar escaneos de forma libre de pago.

Capítulo 4. Análisis de la situación

Mediante la utilización de las herramientas seleccionadas, el equipo de trabajo inicio las pruebas de escaneos en búsqueda de vulnerabilidades con el fin de identificar mejoras en la aplicación web objetivo. Estas pruebas iniciaron el lunes 10 de julio del 2023 y culminaron el lunes 31 de julio del 2023.

El método utilizado para el análisis consistió en lanzar los diferentes sistemas de escaneos bajo un patrón de rastreo basado en los términos de caja blanca y caja gris, omitiendo el patrón de caja negra, esto debido a que el equipo de investigación tenía acceso previo al código fuente de la aplicación, lo que descarta una prueba de caja negra objetiva, ya que el equipo no podía aludir desconocimiento del ámbito y arquitectura de la aplicación web, por lo que implicaría que los investigadores podrían aplicar pruebas sesgadas por la información obtenida al conocer la arquitectura y diseño de la solución web y de esta forma contaminar los resultados debido a que estos ya manejaban la información mencionada.

El equipo de investigación presentó un resumen de actividades de tipo bitácora donde se define el día y la hora en que se realizaron los diferentes sondeos realizados sobre “CREDIWEB”.

4.1 Actividades Realizadas

Aclarado lo anterior, el equipo inició con sus operaciones el día 10 de julio del 2023 al ser aproximadamente las 18:00 horas, en este punto inicial se accedió al entorno controlado del aplicativo web, donde se revisó que el servidor de aplicación estuviera en funcionamiento, que el aplicativo web definido como “CREDIWEB” estuviera corriendo adecuadamente dentro del servidor web, se validó que el equipo auditor y el equipo auditado se pudieran comunicar por medio de la red, entre las acciones más comunes para esta última actividad se lanzaron comandos como “ping”, “traceroute”, “nslookup”,

con el fin de comprobar una adecuada trazabilidad y que el objetivo estuviera completamente aislado.

Se verificó que no se pudiera acceder a ningún otro recurso de la institución financiera, para no comprometer información de ningún tipo, más que a la que se le autorizó al equipo de investigación, esto para mantener un ámbito de trabajo limitado únicamente al alcance propuesto, sin caer en posibles infracciones por extrapolar el alcance de los escaneos a otros servicios o componentes para los cuales no se contaba con autorización.

Al ser aproximadamente las 18:00 horas del día 11 de julio del 2023 se inicia un escaneo de vulnerabilidades sobre la solución web definida como “CREDIWEB”, esto utilizando la herramienta “Zen Attack Proxy (ZAP)” del Proyecto “OWASP Foundation”, en modo “Attack Mode”, este modo se utilizó debido a que se contaba con un entorno aislado el cual permitía este nivel de explotación, sin embargo para entornos de producción es recomendable utilizar escaneos como “Safe Mode”, “protected Mode” (Solo escanea la página objetivo) que son menos agresivos que un escaneo de tipo “Standard Mode” (Escanea la página objetivo y sus enlaces).

Al ser aproximadamente las 22:00 horas del día 11 de julio del 2023 se pausa el escaneo de vulnerabilidades lanzado desde la herramienta “Zen Attack Proxy (ZAP)” hacia la solución web definida como “CREDIWEB”, esto debido a que el escaneo extendió su tiempo de ejecución.

Al ser aproximadamente las 18:00 horas del día 12 de julio del 2023 se retoma el escaneo de vulnerabilidades sobre la solución web definida como “CREDIWEB”, esto utilizando la herramienta “Zen Attack Proxy (ZAP)” del Proyecto “OWASP Foundation”, en modo “Attack Mode”.

Al ser aproximadamente las 22:00 horas del día 12 de julio del 2023 se pausa el escaneo de vulnerabilidades lanzado desde la herramienta “Zen Attack Proxy (ZAP)” hacia la solución web definida como “CREDIWEB”, esto debido a que el escaneo extendió su tiempo de ejecución.

Al ser aproximadamente las 18:00 horas del día 13 de julio del 2023 se retoma el escaneo de vulnerabilidades sobre la solución web definida como

“CREDIWEB”, esto utilizando la herramienta “Zen Attack Proxy (ZAP)” del Proyecto “OWASP Foundation”, en modo “Attack Mode”.

Al ser aproximadamente las 21:15 horas del día 13 de julio del 2023 finaliza el escaneo de vulnerabilidades sobre la solución web definida como “CREDIWEB”, esto utilizando la herramienta “Zen Attack Proxy (ZAP)” del Proyecto “OWASP Foundation”, en modo “Attack Mode”, por lo que se procede a guardar la información obtenida para un análisis posterior.

Concluido el escaneo mediante la utilización de “Zen Attack Proxy (ZAP)”, al ser aproximadamente las 18:00 horas del día 17 de julio del 2023 se inicia el escaneo de vulnerabilidades sobre la solución web definida como “CREDIWEB”, esto utilizando la herramienta “Burp Suite” 2023 de la compañía “PortSwigger” en su versión de prueba de 30 días. Para el contexto de sondeo propuesto sobre “Burp Suite” se utilizó al igual que con la herramienta anterior, un escaneo profundo, porque el fin que se buscó fue entregar todas las vulnerabilidades no identificadas al momento de esta investigación.

Al ser aproximadamente las 22:00 horas del día 17 de julio del 2023 se pausó el escaneo de vulnerabilidades lanzado desde la herramienta “Burp Suite” versión 2023 hacia la solución web definida como “CREDIWEB”, esto debido a que el escaneo extendió su tiempo de ejecución.

Al ser aproximadamente las 18:00 horas del día 18 de julio del 2023 se retoma el escaneo de vulnerabilidades sobre la solución web definida como “CREDIWEB”, esto utilizando la herramienta “Burp Suite” versión 2023 de la compañía “PortSwigger”, en su configuración más pesada “Deep”.

Al ser aproximadamente las 22:00 horas del día 18 de julio del 2023 se pausó el escaneo de vulnerabilidades lanzado desde la herramienta “Burp Suite” versión 2023 hacia la solución web definida como “CREDIWEB”, esto debido a que el escaneo extendió su tiempo de ejecución.

Al ser aproximadamente las 18:00 horas del día 19 de julio del 2023 se retoma el escaneo de vulnerabilidades sobre la solución web definida como “CREDIWEB”, esto utilizando la herramienta “Burp Suite” versión 2023 de la compañía “PortSwigger”, en su configuración más pesada “Deep”.

Al ser aproximadamente las 22:00 horas del día 19 de julio del 2023 se pausó el escaneo de vulnerabilidades lanzado desde la herramienta “Burp Suite” versión 2023 hacia la solución web definida como “CREDIWEB”, esto debido a que el escaneo extendió su tiempo de ejecución.

Al ser aproximadamente las 18:00 horas del día 20 de julio del 2023 se retoma el escaneo de vulnerabilidades sobre la solución web definida como “CREDIWEB”, esto utilizando la herramienta “Burp Suite” versión 2023 de la compañía “PortSwigger”, en su configuración más pesada “Deep”.

Al ser aproximadamente las 22:00 horas del día 20 de julio del 2023 se pausó el escaneo de vulnerabilidades lanzado desde la herramienta “Burp Suite” versión 2023 hacia la solución web definida como “CREDIWEB”, esto debido a que el escaneo extendió su tiempo de ejecución.

Al ser aproximadamente las 18:00 horas del día 21 de julio del 2023 se retoma el escaneo de vulnerabilidades sobre la solución web definida como “CREDIWEB”, esto utilizando la herramienta “Burp Suite” versión 2023 de la compañía PortSwigger”, en su configuración más pesada “Deep”.

Al ser aproximadamente las 20:15 horas del día 21 de julio del 2023 finaliza el escaneo de vulnerabilidades sobre la solución web definida como “CREDIWEB”, esto utilizando la herramienta “Burp Suite” versión 2023 de la compañía “PortSwigger”, en su configuración más pesada “Deep”, por lo que se procede a guardar la información obtenida para un análisis posterior.

Concluido el escaneo mediante la utilización de “Burp Suite”, al ser aproximadamente las 08:00 horas del día 22 de julio del 2023 se inicia el escaneo de vulnerabilidades sobre la solución web definida como “CREDIWEB”, esto utilizando la herramienta “Nikto” escrita por los señores Chris Sullo y David Lodge. Para el contexto de sondeo propuesto sobre “Nikto” se utilizó al igual que con las herramientas anteriores, un escaneo agresivo, con el fin de identificar el mayor número de vulnerabilidades o riesgos relacionados a la solución web de la entidad financiera.

Al ser aproximadamente las 19:48 horas del día 22 de julio del 2023 finaliza el escaneo de vulnerabilidades sobre la solución web definida como

“CREDIWEB”, esto utilizando la herramienta “Nikto” escrita por los señores Chris Sullo y David Lodge, por lo que se procede a guardar la información obtenida para un análisis posterior.

Concluidos los diferentes sondeos o escaneos realizados a la aplicación web denominada como “CREDIWEB” se procede a realizar el análisis de los datos obtenidos, con el fin de descartar la mayor cantidad de falsos negativos o positivos que por consecuencia crearon ruido a los diferentes escaneos realizados.

Al ser aproximadamente las 18:00 horas del día 24 de julio del 2023 el equipo de investigación se reúne de forma virtual para el inicio del análisis de vulnerabilidades o riesgos identificados en los resultados obtenidos de la herramienta “Zen Attack Proxy (ZAP)”.

Al ser aproximadamente las 22:00 horas del día 24 de julio del 2023 el equipo de investigación finaliza el análisis de vulnerabilidades o riesgos identificados en los resultados obtenidos de la herramienta “Zen Attack Proxy (ZAP)”.

Al ser aproximadamente las 18:00 horas del día 25 de julio del 2023 el equipo de investigación se reúne de forma virtual para el inicio del análisis de vulnerabilidades o riesgos identificados en los resultados obtenidos de la herramienta “Nikto”.

Al ser aproximadamente las 22:00 horas del día 25 de julio del 2023 el equipo de investigación finaliza el análisis de vulnerabilidades o riesgos identificados en los resultados obtenidos de la herramienta “Nikto”.

Al ser aproximadamente las 18:00 horas del día 26 de julio del 2023 el equipo de investigación se reúne de forma virtual para el inicio del análisis de vulnerabilidades o riesgos identificados en los resultados obtenidos de la herramienta “Burp Suite”.

Al ser aproximadamente las 23:50 horas del día 26 de julio del 2023 el equipo de investigación finaliza el análisis de vulnerabilidades o riesgos identificados en los resultados obtenidos de la herramienta “Burp Suite”.

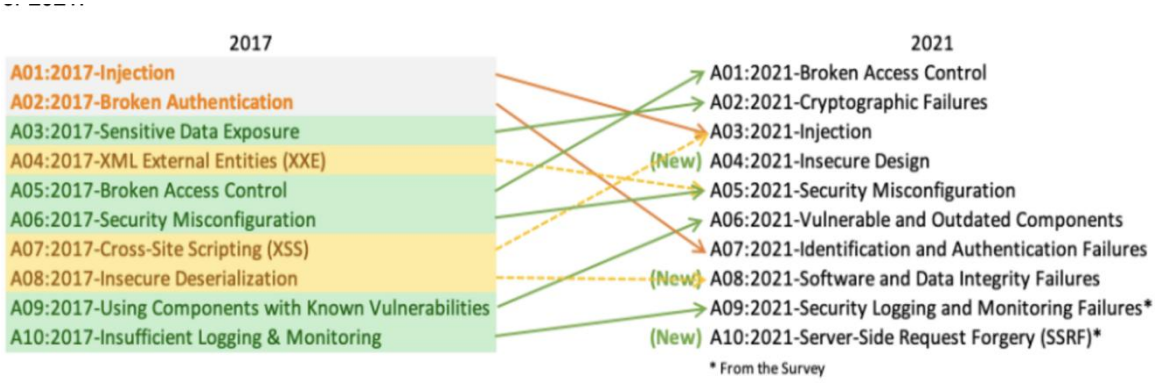
Una nota importante para considerar es que los escaneos realizados con las herramientas “Zen Attack Proxy (ZAP)” y “Burp Suite” no se dejaban ejecutar de forma ininterrumpida debido a que el equipo de investigación era responsable de cualquier anomalía, contratiempo o evento no planificado que pudiera ocurrir durante el sondeo o manipulación de la copia del código fuente suministrado por la entidad financiera, por lo que se acordó que el equipo de trabajo estaría siempre presente a la hora de realizar cualquier actividad sobre la solución web.

Otra consideración importante fue que los escaneos realizados con “Nikto” no se podían pausar, por lo que se designó un día fuera de la semana laboral para realizar el testeo con esta herramienta.

Por otra parte, las vulnerabilidades identificadas en la aplicación web denominada “CREDIWEB” fueron categorizadas según su eje en el “Top 10” de “OWASP Foundation”.

Este está priorizado en una lista, la cual define cada categoría con base a la frecuencia de ocurrencia en que se van identificando o se han identificado las diferentes vulnerabilidades encontradas en los diferentes sitios web, las cuales son consideradas como los riesgos más comunes en este tipo de aplicaciones. A continuación, se muestra el “Top 10” de “OWASP Foundation”:

Ilustración 1: Top Ten 2021 de OWASP



(OWASP. 2021. Top 10 Web Application Security Risks. https://owasp.org/www-project-top-ten/).

A continuación, se muestra una conciliación de las diferentes vulnerabilidades identificadas con las distintas herramientas utilizadas en el escaneo de vulnerabilidades realizado sobre la aplicación web denominada “CREDIWEB”, la cual pertenece a una entidad financiera nacional.

La presentación de estas vulnerabilidades estará definida por la siguiente estructura:

1. Título de la categoría según el “Top 10” de “OWASP Foundation”:
2. Breve descripción de ítem.
3. Vulnerabilidades relacionadas al ítem según el “Top 10”.
4. Vulnerabilidades identificadas en la solución denominada “CREDIWEB”.

4.2 A01:2021 Pérdida de Control de Acceso

La pérdida de control de acceso o “Broken Access Control” está relacionada al control de acceso y define los límites de acción de los usuarios dentro del sistema o aplicación.

Vulnerabilidades de este tipo pueden desencadenar o terminar en la divulgación, modificación o destrucción no autorizada de información para la empresa o en este caso entidad financiera, ya que el poder eludir estos controles puede permitir la manipulación de metadatos, manipulación de tokens o cookies, escalaciones de privilegios o movimientos horizontales y acceso desde orígenes no autorizados.

Por otra parte, se ve afectado el principio del menor privilegio o el de denegación por defecto, los cuales basan su actuar en que el usuario debería tener únicamente los accesos necesarios para su labor o el sistema debería negar todo y solo permitir lo que se ha establecido como permitido en su configuración.

4.2.1 Vulnerabilidades relacionadas con la pérdida de control de acceso

Según la publicación del “Top 10” de “OWASP” las siguientes son las vulnerabilidades comunes identificadas que tiene relación con la pérdida del control de acceso:

- CWE-22 Limitación incorrecta de un nombre de ruta a un directorio restringido.
- CWE-23 Travesía de la trayectoria relativa.
- CWE-35 Travesía del Ruta '... /... //'
- CWE-59 Resolución de enlace incorrecta antes del acceso al archivo.
- CWE-200 Exposición de información confidencial a un actor no autorizado
- CWE-201 Exposición de información confidencial a través de datos Enviados
- CWE-219 Almacenamiento de archivos con datos confidenciales bajo raíz web.
- CWE-264 Permisos, privilegios y controles de acceso.
- CWE-275 Problemas de permiso.
- CWE-276 Permisos predeterminados incorrectos.
- CWE-284 Control de acceso inadecuado.
- CWE-285 Autorización incorrecta.
- CWE-352 Falsificación de solicitudes entre sitios (CSRF).
- CWE-359 Exposición de información personal privada a un actor no autorizado.
- CWE-377 Archivo temporal inseguro.
- CWE-402 Transmisión de recursos privados a una nueva esfera.
- CWE-425 Solicitud directa ("navegación forzada")
- CWE-441 Poder o intermediario no deseado.
- CWE-497 Exposición de información confidencial del sistema a una esfera de control no autorizada.
- CWE-538 Inserción de información confidencial en un archivo o directorio de acceso externo.
- CWE-540 Inclusión de información confidencial en el código fuente.
- CWE-548 Exposición de información a través de la lista de directorios.

- CWE-552 Archivos o directorios accesibles a partes externas.
- CWE-566 Omitir la autorización a través de la clave principal SQL controlada por el usuario.
- CWE-601 Redirección de URL de a un sitio no confiable ("Redirección abierta").
- CWE-639 Omitir la autorización a través de la clave controlada por el usuario.
- CWE-651 Exposición del archivo WSDL que contiene información confidencial.
- CWE-668 Exposición de recursos a la esfera equivocada.
- CWE-706 Uso de un nombre o referencia resuelto incorrectamente.
- CWE-862 Falta la autorización.
- CWE-863 Autorización incorrecta.
- CWE-913 Control inadecuado de los recursos de código gestionados dinámicamente.
- CWE-922 Almacenamiento inseguro de información confidencial.
- CWE-1275 Cookie sensible con atributo incorrecto del mismo sitio.

4.2.2 Vulnerabilidades relacionadas con la pérdida de control de acceso que fueron identificadas en la solución denominada “CREDIWEB”

1. Se identifico que la solicitud enviada por el aplicativo parece ser vulnerable a los ataques de falsificación de solicitudes entre sitios (CSRF) contra usuarios autenticados ya que la solicitud original contiene parámetros que parecen ser tokens anti-CSRF. Sin embargo, la solicitud tiene éxito si se eliminan estos parámetros. Esto ocurre en diez rutas del sistema las cuales se detallan a continuación:

Tabla 2: Falsificación de solicitud entre sitios.

Asunto:	Falsificación de solicitud entre sitios
Gravedad:	**Medio**
Confianza:	**Tentativo**

Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/CREDIWEB /Vista_Parametros**
Ruta:	**/CREDIWEB /Mantenimiento_CatálogoServicios**
Ruta:	**/CREDIWEB /Mantenimiento_CatálogoNotarios**
Ruta:	**/CREDIWEB /Mantenimiento_CatálogoGarantías**
Ruta:	**/CREDIWEB /Mantenimiento_CatálogoAsistentesNotarios**
Ruta:	**/ CREDIWEB / CompromisoPago_Reporte**
Ruta:	**/ CREDIWEB / Registro_Llamada**
Ruta:	**/ CREDIWEB / Carga_Parametros**
Ruta:	**/ CREDIWEB / DatosAsignacionCasos**

(Pérez y Molina. 2023. Falsificación de solicitud entre sitios)

Esto puede ser debido a que este tipo de falsificación de solicitudes entre sitios web puede usar únicamente autenticación de cookies tipo HTTP para la identificación de usuarios. Lo anterior permite que un usuario malicioso pueda autenticarse con la reutilización de tokens relacionados a la sesión del usuario, oh que el usuario sea capaz de determinar por el análisis de patrones los mecanismos necesarios para la construcción de una solicitud e inyectarla de forma exitosa.

2. Se localizaron e identificaron numerosas direcciones de correo electrónico al lanzar un ataque de “Footprinting” contra varias rutas del aplicativo. Esto incrementa el número de ataques de “Phisihng” o de ingeniería social contra el personal interno de la entidad financiera, identificar nombres de usuarios y con esto crear patrones de direcciones de correo que sean utilizados para elaborar diccionarios y utilizarlos en ataques de fuerza bruta.

   A continuación, se muestran las cuentas de correo electrónicos identificadas dentro del sistema y que son de usuarios específicos:

Tabla 3: Direcciones de correo electrónico reveladas.

Asunto:	Direcciones de correo electrónico reveladas
Gravedad:	**Información**
Confianza:	**Cierto**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/**
Correo Electrónico:	**gerente100@b---.f-.c-**
Correo Electrónico:	**MVILLALOBOSb---.f-.c-**
Correo Electrónico:	**OBARRANTESb---.f-.c-**
Correo Electrónico:	**GURENAb---.f-.c-**
Correo Electrónico:	**WRODRIGUEZBb---.f-.c-**
Correo Electrónico:	**osarmientob---.f-.c-**
Correo Electrónico:	**achavesb---.f-.c-**
Correo Electrónico:	**szeledonb---.f-.c-**
Correo Electrónico:	**MVAGUILARFb---.f-.c-**
Correo Electrónico:	**jfonsecab---.f-.c-**
Correo Electrónico:	**LCHACONb---.f-.c-**
Correo Electrónico:	**jhernandezrb---.f-.c-**
Correo Electrónico:	**dlopezb---.f-.c-**
Correo Electrónico:	**wgomezsb---.f-.c-**
Correo Electrónico:	**wespinozab---.f-.c-**
Correo Electrónico:	**JMORASAb---.f-.c-**
Correo Electrónico:	**DARIASb---.f-.c-**
Correo Electrónico:	**CACUNAb---.f-.c-**

Correo Electrónico:	**EABARCAb---.f-.c-**

(Pérez y Molina. 2023. Direcciones de correo electrónico reveladas).

3. Se identificaron múltiples cadenas de consulta en la raíz del aplicativo web, lo cual puede provocar una fuga de información a través de los diferentes dominios, ya que al sitio web realizar solicitudes de recursos a otras referencias, esta consulta se carga en los encabezados HTTP, esta referencia muestra la URL de origen y posiblemente un token de sesión e información del sistema o de la página web que llama el recurso.

   Ahora bien, si el recurso solicitado está en un dominio externo, el cual podría ser o no inseguro, la información mencionada será transmitida a este poniendo en peligro la información transmitida entre los dominios, considerando esto como un riesgo potencial.

   A continuación, se muestran todos los enlaces que se identificaron. Estos enlaces presentaron el comportamiento mencionado, por lo tanto, las siguientes rutas del sistema transmiten información entre dominios distintos al de la entidad financiera:

Tabla 4: Fuga de referencia entre dominios.

Asunto:	Fuga de referencia entre dominios
Gravedad:	**Información**
Confianza:	**Cierto**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/**
Enlace:	**https://github.com/mozilla/blurts-servidor**
Enlace:	**https://support.mozilla.org/kb/firefox-monitor-faq**
Enlace:	**https://assets-prod.sumo.prod.webservices.mozgcp.net/media/uploads**

	/products/2020-04-14-08-36-13-8dda6f.png
Enlace:	**https://activos-prod.sumo.prod.webservices.mozgcp.net/static/354.3a7ddcb120703df0.js**
Enlace:	**https://activos-prod.sumo.prod.webservices.mozgcp.net/static/672.2aa5f418a3ec2171.js**
Enlace:	**https://activos-prod.sumo.prod.webservices.mozgcp.net/static/901.5c1bf4c53aef45f5.js**
Enlace:	**https://assets-prod.sumo.prod.webservices.mozgcp.net/static/apple-touch-icon.bdc11b610d791a16.png**
Enlace:	**https://activos-prod.sumo.prod.webservices.mozgcp.net/static/common.6968c88d57044d85.js**
Enlace:	**https://activos-prod.sumo.prod.webservices.mozgcp.net/static/common.fx.download.7c161292b0ad1beb.js**
Enlace:	**https://assets-prod.sumo.prod.webservices.mozgcp.net/static/default-FFA-avatar.2f8c2a0592bda1c5.png**
Enlace:	**https://activos-prod.sumo.prod.webservices.mozgcp.net/static/document.b5035c05ca0ac300.js**
Enlace:	**https://assets-prod.sumo.prod.webservices.mozgcp.net/static/favicon-16x16.352a0a2cef154dda.png**
Enlace:	**https://activos-prod.sumo.prod.webservices.mozgcp.net/static/favicon-**

	32x32.2143cdf0c7e3f377.png
Enlace:	**https://assets-prod.sumo.prod.webservices.mozgcp.net/static/favicon.03d97697df808c0e.ico**
Enlace:	**https://assets-prod.sumo.prod.webservices.mozgcp.net/static/gtm-snippet.c572a96f8b784ca0.js**
Enlace:	**https://assets-prod.sumo.prod.webservices.mozgcp.net/static/jsi18n/es/djangojs-min.js**
Enlace:	**https://assets-prod.sumo.prod.webservices.mozgcp.net/static/placeholder.688345f843bb37ed.gif**
Enlace:	**https://activos-prod.sumo.prod.webservices.mozgcp.net/static/screen.5bf883e5e9b34141.css**
Enlace:	**https://assets-prod.sumo.prod.webservices.mozgcp.net/static/spinner.18d6c26adc937688.gif**
Enlace:	**https://assets-prod.sumo.prod.webservices.mozgcp.net/static/volunteer.a3be8d331849774b.png**
Enlace:	**https://blog.avast.com/2015/05/25/explaining-avasts-https-scanning-feature/**
Enlace:	**https://es.wikipedia.org/wiki/Autoridad_de_certificaci..n_ _**
Enlace:	**https://github.com/mozilla/kitsune/**
Enlace:	**https://support.avast.com/es-es/article/189/**
Enlace:	**https://support.microsoft.com/es-es/kb/2965142**

Enlace:	**https://support.microsoft.com/help/14210/security-essentials-download**
Enlace:	**https://twitter.com/**
Enlace:	**https://www.bitdefender.com/support/repairing-or-removing-bitdefender-free-edition-1160.html**
Enlace:	**https://www.instagram.com/**
Enlace:	**https://www.kaspersky.com/descargas**
Enlace:	**https://www.microsoft.com/windows/seguridad-integral**
Enlace:	**https://www.ssllabs.com/ssltest**

(Pérez y Molina. 2023. Fuga de referencia entre dominios).

4.3. A02:2021 Fallas Criptográficas

Las fallas criptográficas o “Cryptographic Failures” están basadas en proteger los datos que se encuentran transitando la red que son sensibles o confidenciales, o deberían serlo dentro del sistema que gestiona esta información.

Algunos ejemplos de estos datos son los números de tarjetas de crédito, registros de salud, contraseñas de la aplicación, información personal del usuario, o cualquier otro que se encuentre definido en la Ley de Protección de la Persona frente al tratamiento de sus datos personales (ley 8968) o su Reglamento a la Ley de Protección de la Persona frente al Tratamiento de sus Datos Personales (N° 37554-JP).

Para poder determinar si los datos están protegidos y que no viajan por la red en texto claro, con un cifrado obsoleto o vulnerable se deben considerar:

1. Existen algoritmos o protocolos criptográficos obsoletos o descontinuados como MD5 o SHA1 o se utilizan métodos de encriptación como PKCS en su versión 1.

2. Las claves criptográficas tienen una longitud adecuada según la información que protegen.
3. Los encabezados HTTP o HTTPS cuentan con las directivas de seguridad adecuadas.
4. Se validan las llaves y los certificados del servidor web.
5. Se verifica que los vectores de inicialización se están ejecutando de forma aleatoria y adecuada, evitando la reutilización de estos.

4.3.1 Vulnerabilidades relacionadas con las fallas criptográficas

Según la publicación del “Top 10” de “OWASP” las siguientes son las vulnerabilidades comunes identificadas que tiene relación con las fallas criptográficas:

- CWE-261 Codificación débil para la contraseña.
- CWE-296 Seguimiento inadecuado de la cadena de confianza de un certificado.
- CWE-310 Problemas criptográficos.
- CWE-319 Texto claro Transmisión de información confidencial.
- CWE-321 Uso de una clave criptográfica codificada.
- CWE-322 Intercambio de claves sin autenticación de entidad.
- CWE-323 Reutilización de un Nonce, par de claves en cifrado.
- CWE-324 Uso de una clave después de su fecha de caducidad.
- CWE-325 Falta el paso criptográfico requerido.
- CWE-326 Fuerza de cifrado inadecuada.
- CWE-327 Uso de un algoritmo criptográfico roto o arriesgado.
- CWE-328 Hash unidireccional reversible.
- CWE-329 No utiliza un IV aleatorio con modo CBC.
- CWE-330 Uso de valores insuficientemente aleatorios.
- CWE-331 Entropía insuficiente.
- CWE-335 Uso incorrecto de semillas en el generador de números pseudoaleatorios (PRNG).

- CWE-336 Misma semilla en el generador de números pseudoaleatorios (PRNG).
- CWE-337 Semilla predecible en generador de números pseudoaleatorios (PRNG).
- CWE-338 Uso de un generador de números pseudoaleatorios criptográficamente débil (PRNG).
- CWE-340 Generación de números o identificadores predecibles.
- CWE-347 Verificación incorrecta de la firma criptográfica.
- CWE-523 Transporte de credenciales sin protección.
- CWE-720 OWASP Top Ten 2007 Categoría A9 - Comunicaciones inseguras.
- CWE-757 Selección de un algoritmo menos seguro durante la negociación ("Degradación del algoritmo").
- CWE-759 Uso de un hash unidireccional sin salto.
- CWE-760 Uso de un hash unidireccional con un salto predecible.
- CWE-780 Uso del algoritmo RSA sin OAEP.
- CWE-818 Protección insuficiente de la capa de transporte.
- CWE-916 Uso de hash de contraseñas con un esfuerzo computacional insuficiente.

4.3.2 Vulnerabilidades relacionadas con las fallas criptográficas que fueron identificadas en la solución denominada “CREDIWEB”

1. Se identificó que la aplicación permite a los usuarios conectarse a ella a través de conexiones no cifradas, esto permite que un atacante con acceso a la red pueda monitorear y ver el tráfico de red de un usuario legítimo, logrando registrar y monitorear las interacciones de este con el sistema para obtener cualquier información que el usuario suministre. Además, un atacante capaz de modificar el tráfico podría utilizar la aplicación como plataforma para ataques contra sus usuarios y sitios web de terceros.

   Considerando que este aplicativo web está en la red interna, se considera difícil la explotación de esta desde el exterior, no obstante, si es explotable desde una red interna.

Por lo que se debe considerar un factor de ataque el teletrabajo, ya que este incrementa el riesgo de la materialización de esta vulnerabilidad debido a que las redes domésticas normalmente poseen contraseñas de acceso a los puntos de acceso débiles o inseguras y que estos puntos de acceso brindados por el proveedor de servicios (ISP) pueden poseer algoritmos de cifrado débiles u obsoletos, lo cual podrían ser blanco fácil para un ataque de hombre en el medio (MitM), provocando la explotación mencionada.

Teniendo estos escenarios claros, a continuación, se muestran las rutas explotables:

Tabla 5: Comunicaciones sin cifrar.

Asunto:	Comunicaciones sin cifrar
Gravedad:	**Bajo**
Confianza:	**Cierto**
Anfitrión:	**http://c----------.o-------.org**
Anfitrión:	**http://o---.d-------.com**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	/

(Pérez y Molina. 2023. Comunicaciones sin cifrar).

4.4 A03:2021 Inyección

La Inyección de datos o "Injection" ocurre cuando se envía a través de las URLs, “Scripts” o formularios, datos que no son válidos y estos son interpretados o devueltos por la aplicación de forma errónea, arrojando información valiosa.

Estos datos pueden ser enviados por medio de concatenación o de forma directa, estos son estructurados en relación con el objetivo, con el fin de

hacer una búsqueda o mapeo relacional de objetos (ORM), con el fin de obtener alguna clase de registros que sean útiles para el atacante.

Las inyecciones más comunes ocurren sobre consultas o base de datos de tipo SQL, NoSQL, comando OS, Object Relational Mapping (ORM), LDAP y Expression Language (EL) o Object Graph Navigation Library (OGNL).

4.4.1 Vulnerabilidades relacionadas con la Inyección

Según la publicación del “Top 10” de “OWASP” las siguientes son las vulnerabilidades comunes identificadas que tiene relación con la Inyección:

- CWE-20 Validación de entrada incorrecta.
- CWE-74 Neutralización incorrecta de elementos especiales en la salida utilizados por un componente.
- CWE-75 Fallo para desinfectar elementos especiales en un plano diferente (inyección de elementos especiales).
- CWE-77 Neutralización incorrecta de elementos especiales utilizados en un comando ("Inyección de comando").
- CWE-78 Neutralización incorrecta de elementos especiales utilizados en un comando del sistema operativo ("Inyección de comandos del sistema operativo").
- CWE-79 Neutralización incorrecta de la entrada durante la generación de páginas web.
- CWE-80 Neutralización incorrecta de etiquetas HTML relacionadas con scripts en una página web (Basic XSS).
- CWE-83 Neutralización incorrecta de la secuencia de comandos en atributos en una página web.
- CWE-87 Neutralización incorrecta de la sintaxis alternativa XSS.
- CWE-88 Neutralización incorrecta de delimitadores de argumentos en un comando ("Inyección de argumentos").
- CWE-89 Neutralización incorrecta de elementos especiales utilizados en un comando SQL ("SQL Injection").

- CWE-90 Neutralización incorrecta de elementos especiales utilizados en una consulta LDAP ("Inyección LDAP").
- Inyección XML CWE-91 (también conocida como inyección ciega de XPath).
- CWE-93 Neutralización inadecuada de secuencias de CRLF ("Inyección de CRLF").
- CWE-94 Control inadecuado de la generación de código ("Inyección de código").
- CWE-95 Neutralización incorrecta de directivas en código evaluado dinámicamente ("Inyección de evaluación").
- CWE-96 Neutralización incorrecta de directivas en código guardado estático ("Inyección de código estático").
- CWE-97 Neutralización incorrecta de los incluyes del lado del servidor (SSI) dentro de una página web.
- CWE-98 Control inadecuado del nombre de archivo para la declaración Include/Require en el programa PHP ("Inclusión de archivos remotos PHP").
- CWE-99 Control inadecuado de los identificadores de recursos ("Inyección de recursos").
- CWE-100 En desaprobación: fue un problema para los problemas de validación de entrada.
- CWE-113 Neutralización incorrecta de secuencias CRLF en encabezados HTTP ("División de respuestas HTTP").
- CWE-116 Codificación incorrecta o escape de la salida.
- CWE-138 Neutralización inadecuada de elementos especiales.
- CWE-184 Lista incompleta de entradas no permitidas.
- CWE-470 Uso de entrada controlada externamente para seleccionar clases o código ("Reflexión insegura").
- CWE-471 Modificación de datos inmutables asumidas (MAID).
- CWE-564: Inyección SQL por Hibernar.
- CWE-610 Referencia controlada externamente a un recurso en otra esfera.
- CWE-643 Neutralización incorrecta de datos dentro de las expresiones XPath ("Inyección XPath").
- CWE-644 Neutralización incorrecta de los encabezados HTTP para la sintaxis de secuencias de comandos.

- CWE-652 Neutralización incorrecta de datos dentro de las expresiones de XQuery ("Inyección de XQuery").
- CWE-917 Neutralización incorrecta de elementos especiales utilizados en una declaración de lenguaje de expresión ("Inyección de lenguaje de expresión").

4.4.2 Vulnerabilidades relacionadas con la inyección que fueron identificadas en la solución denominada “CREDIWEB”

1. Se identificó que los parámetros “Filtro_TipoCaso” y “Filtro_Etapa” pueden ser vulnerables a los ataques de inyección SQL, esto debido a que se enviaron mediante las herramientas de escaneo de vulnerabilidades web utilizadas varios “payloads” y se detectó que en los parámetros “Filtro_TipoCaso” y “Filtro_Etapa” fue devuelto un mensaje de error de la base de datos, determinando que esta es una base de datos de tipo relacional e identificando el motor de base de datos como: “Microsoft SQL Server”.

A continuación, se muestran las rutas detectadas como vulnerables a inyección de tipo SQL en el aplicativo:

Tabla 6: inyección SQL.

Asunto:	inyección SQL
Gravedad:	**Alto**
Confianza:	**Firme**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/CREDIWEB/DatosAsignacionCasos**
Ruta:	**/CREDIWEB/AsignarCaso_ConFiltros**

(Pérez y Molina. 2023. inyección SQL).

2. Se identificó que la aplicación es vulnerable en la manipulación de enlaces basada en DOM. Los datos se leen desde “document.location.href” y “location.href” lo cual permite pasar a la propiedad 'href' del elemento de tipo DOM.

Esto debido a que se identificaron “scripts” que permiten la escritura de datos en un objetivo de navegación dentro de los enlaces, permitiendo manipular el comportamiento de un enlace de los formularios identificados.

Esto puede facilitar a un ciberdelincuente realizar:

a. redirecciones de páginas web a URLs externas de forma arbitraria para facilitar un ataque de phishing.
b. Redireccionas la información que el usuario envía desde un formulario a un servidor controlado por el atacante.
c. Evadir las defensas anti-XSS que se impusieron en el navegador, esto mediante la inyección de enlaces en el sitio que contienen vulnerabilidades XSS, esto debido a que las defensas anti-XSS del navegador no funcionan en los enlaces del sitio.

A continuación, se muestran las rutas de los formularios con los scripts que permiten la escritura de datos en un objetivo de tipo DOM:

Tabla 7: Manipulación de enlaces basados en DOM.

Asunto:	Manipulación de enlaces (basada en DOM)
Gravedad:	**Bajo**
Confianza:	**Firme**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	/
Ruta:	**/CREDIWEB/VisorFileOn**
Ruta:	**/CREDIWEB/ReglaSolicitudHija**
Ruta:	**/CREDIWEB/RegistroLlamadas**

Ruta:	**/CREDIWEB/ReglaCasos**
Ruta:	**/CREDIWEB/MantenimientoUsuario**
Ruta:	**/CREDIWEB/PendientesTramitar**
Ruta:	**/CREDIWEB/MantenimientoAdmin**
Ruta:	**/CREDIWEB/MantenimientoGeneral**
Ruta:	**/CREDIWEB/MantenimientoReglas**
Ruta:	**/CREDIWEB/Índice**
Ruta:	**/CREDIWEB/FileOn_Reporte**
Ruta:	**/CREDIWEB/CobroAdministrativo_AsignacionColas**
Ruta:	**/CREDIWEB/TareasEspeciales**

(Pérez y Molina. 2023. Manipulación de enlaces basados en DOM).

3. Se identificó que la aplicación web quita el escape de las secuencias de escape de la barra invertida cuando procesa el valor del nombre de un archivo en la ruta de URL de la página web, también se identifica que en algunos casos procesa esta secuencia de escape de una manera inesperada, dando como resultado de ambos eventos, un reenvío de la solicitud.

En ambos casos se envió mediante los escaneos automatizados varias secuencias de escape, utilizando cargas útiles como la siguiente: “653icmato3\\l8f47mwo7m” en las URLs de las siguientes rutas:

Tabla 8: Transformación de entrada sospechosa

Asunto:	Transformación de entrada sospechosa (reflejada)
Gravedad:	**Información**
Confianza:	**Firme**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/Content/ DataTablePaginacion /datatables.min.js**

Ruta:	**/Contenido/complementos/bootstrap/ js /bootstrap.bundle.min.js**
Ruta:	**/Contenido/complementos/ jquery /jquery.min.js**
Ruta:	**/ CREDIWEB / Cargar Catálogos Principales**
Ruta:	**/ CREDIWEB / CierrePlanesInversion_Reporte**
Ruta:	**/Content/ DateRangePicker /moment.min.js**
Ruta:	**/Content/ DateRangePicker /daterangepicker.js**
Ruta:	**/Contenido/complementos/bootstrap/ js /bootstrap.bundle.min.js**
Ruta:	**/ CREDIWEB / ComandosRepotesExcel**
Ruta:	**/Scripts/bootstrap.js**
Ruta:	**/Scripts/jquery-3.4.1.min.js**
Ruta:	**/Contenido/ DualListBoxSoporte /jquery.bootstrap-duallistbox.js**
Ruta:	**/ CREDIWEB /Agenda**
Ruta:	**/robots.txt**

(Pérez y Molina. 2023. Transformación de entrada sospechosa).

Este tipo de comportamiento indica que la aplicación se encuentra evaluando la entrada dentro de algún contexto interpretado, lo que podría dar lugar a la inyección de código u otros problemas, debido a que la aplicación está realizando algún tipo de procesamiento adicional.

4. Se identificó que en algunas rutas se copia el valor del nombre del archivo en la “URL” de la respuesta generada por la aplicación, además, se identifica que en la ruta: “/ CREDIWEB / CompromisoPago_Reporte”, se copian el valor del parámetro de la solicitud en la respuesta generada por la

aplicación. Esto ocurre en los siguientes campos: “Txt_Solicitud”, “Txt_TipoCaso”, “Txt_Nombre” y Txt_Cedula”.

Este tipo de comportamiento reflejo permite que los datos se copien de una solicitud y se repitan en la respuesta inmediata de la aplicación. Como tal esta no es una vulnerabilidad, pero si un factor de riesgo asociado a vulnerabilidades del lado del cliente tales como la inyección de secuencias de comandos entre sitios, la redirección abierta, la suplantación de contenido y la inyección de encabezado de respuesta.

La siguiente tabla muestra las rutas identificadas con este comportamiento de tipo reflejo identificadas dentro del aplicativo:

Tabla 9: Entrada devuelta en respuesta.

Asunto:	Entrada devuelta en respuesta (reflejada)
Gravedad:	**Información**
Confianza:	**Cierto**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/Content/ DataTablePaginacion /datatables.min.js**
Ruta:	**/ CREDIWEB / CompromisoPago_Reporte**
Ruta:	**/Contenido/complementos/bootstrap/ js /bootstrap.bundle.min.js**
Ruta:	**/Contenido/complementos/ jquery /jquery.min.js**
Ruta:	**/ CREDIWEB / Cargar Catálogos Principales**
Ruta:	**/ CREDIWEB / CierrePlanesInversion_Reporte**
Ruta:	**/ CREDIWEB / ComandosRepotesExcel**
Ruta:	**/Content/ DateRangePicker /moment.min.js**
Ruta:	**/Content/ DateRangePicker /daterangepicker.js**

Ruta:	**/Scripts/bootstrap.js**
Ruta:	**/Contenido/ DualListBoxSoporte /jquery.bootstrap-duallistbox.js**
Ruta:	**/Scripts/jquery-3.4.1.min.js**
Ruta:	**/ CREDIWEB /Agenda**
Ruta:	**/robots.txt**

(Pérez y Molina. 2023. Entrada devuelta en respuesta).

4.5 A04:2021 Diseño Inseguro

El diseño inseguro o “Insecure Design” es una metodología que evalúa constantemente amenazas, buscando la garantía de que el código fuente que se diseñó para una solución haya sido probado de manera adecuada con el fin de prevenir ataques conocidos.

Se debe determinar mediante la creación de historias de usuarios un flujo de datos adecuado e identificar los posibles puntos de falla en el sistema, además se debe determinar el cómo se validará la inserción de datos y su flujo a través del sistema.

4.5.1 Vulnerabilidades relacionadas con el diseño inseguro

Según la publicación del “Top 10” de “OWASP” las siguientes son las vulnerabilidades comunes identificadas que tiene relación con el diseño inseguro:

- CWE-73 Control externo de nombre de archivo o ruta.
- CWE-183 Lista permisiva de entradas permitidas.
- CWE-209 Generación de mensaje de error que contiene información confidencial.

- CWE-213 Exposición de información confidencial debido a políticas incompatibles.
- CWE-235 Manejo inadecuado de parámetros adicionales.
- CWE-256 Almacenamiento desprotegido de credenciales.
- CWE-257 Almacenamiento de contraseñas en un formato recuperable.
- CWE-266 Asignación de privilegios incorrecta.
- CWE-269 Gestión de privilegios inadecuada.
- CWE-280 Manejo inadecuado de permisos o privilegios insuficientes.
- CWE-311 Falta el cifrado de datos confidenciales.
- CWE-312 Almacenamiento de texto sin cifrar de información confidencial.
- CWE-313 Almacenamiento de texto claro en un archivo o en disco.
- CWE-316 Almacenamiento de texto sin cifrar de información confidencial en la memoria.
- CWE-419 Canal primario desprotegido.
- CWE-430 Implementación de controlador incorrecto.
- CWE-434 Carga sin restricciones de archivo con tipo peligroso.
- CWE-444 Interpretación inconsistente de solicitudes HTTP ('Contrabando de solicitudes HTTP').
- CWE-451 Interfaz de usuario (UI) Tergiversación de información crítica.
- CWE-472 Control externo de parámetro web supuestamente inmutable.
- CWE-501 Violación de límites de confianza.
- CWE-522 Credenciales insuficientemente protegidas.
- CWE-525 Uso de la memoria caché del navegador web que contiene información confidencial.
- CWE-539 Uso de cookies persistentes que contienen información confidencial.
- CWE-579 Malas prácticas de J2EE: Objeto no serializable almacenado en sesión.
- CWE-598 Uso del método de solicitud GET con cadenas de consulta confidenciales.
- CWE-602 Aplicación del lado del cliente de la seguridad del lado del servidor.
- CWE-642 Control externo de datos de estado crítico.

- CWE-646 Dependencia del nombre de archivo o extensión de archivo suministrado externamente.
- CWE-650 Confiar en los métodos de permisos HTTP en el lado del servidor.
- CWE-653 Compartimentación insuficiente.
- CWE-656 Confianza en la seguridad a través de la oscuridad.
- CWE-657 Violación de los principios de diseño seguro.
- CWE-799 Control inadecuado de la frecuencia de interacción.
- CWE-807 Dependencia de entradas no confiables en una decisión de seguridad.
- CWE-840 Errores de lógica de negocios.
- CWE-841 Aplicación incorrecta del flujo de trabajo conductual.
- CWE-927 Uso de intenciones implícitas para comunicaciones confidenciales.
- CWE-1021 Restricción incorrecta de marcos o capas de interfaz de usuario renderizados.
- CWE-1173 Uso inadecuado del marco de validación.

4.5.2 Vulnerabilidades relacionadas con el diseño inseguro que fueron identificadas en la solución denominada “CREDIWEB”

1. Se identificó que el servidor es vulnerable a los ataques de desincronización del lado del cliente ya que mediante el envío de solicitud de tipo “POST” a la ruta “/robots.txt” generando un retraso previo al envío del cuerpo de la solicitud, el servidor agotó el tiempo de espera del cuerpo de la solicitud, pero no cerró la conexión, luego al enviar el cuerpo de la solicitud, el servidor interpretó esta como una nueva solicitud.

   Este tipo de vulnerabilidades relacionadas a la desincronización del lado del cliente (CSD), se hacen presentes debido a que el servidor web no está procesando de forma adecuada la longitud de las peticiones realizadas mediante el método “POST”, lo cual puede permitir que un ciberdelincuente obligue a la víctima a desincronizar su conexión conduciendo a un ataque de tipo XSS (ataques de secuencias de comandos entre sitios).

A continuación, se muestra el detalle del archivo identificado por el escaneo, sobre el cual se identificó el evento:

Tabla 10: Desincronización del lado del cliente

Asunto:	Desincronización del lado del cliente
Gravedad:	**Alto**
Confianza:	**Tentativo**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/robots.txt**

(Pérez y Molina. 2023. Desincronización del lado del cliente).

2. Se identificó un formulario que se utiliza para la carga y envío de archivos, el cual no restringe el tipo de formato o el tamaño de archivo que puede ser cargado, lo que permite la utilización de esta ruta como punto de entrada de archivos maliciosos o utilización de técnicas como la esteganografía para la carga de código malicioso ejecutable dentro de la estructura de almacenamiento de archivos del aplicativo web.

La ruta identificada con este problema es "http://---.---.---.---/CREDIWEB/CobroJudicial_GuardarConsulta". A continuación, se detalla en la tabla siguiente:

Tabla 11: Funcionalidad de carga de archivos.

Asunto:	Funcionalidad de carga de archivos
Gravedad:	**Información**
Confianza:	**Cierto**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/ CREDIWEB / CobroJudicialConsulta**

(Pérez y Molina. 2023. Funcionalidad de carga de archivos).

3. Se identificó que la aplicación web no posee directivas de almacenamiento de cache que le indique al navegador local que no almacene copias locales de los datos confidenciales, debido a esto los navegadores web pueden almacenar una copia en caché local del contenido recibido de los servidores web, almacenando en caché el contenido al que se accede a través de HTTPS, lo que puede generar una brecha de seguridad si otro usuario mediante un ataque de hombre en el medio (MitM) tuviera acceso a la caché del navegador web del equipo comprometido.

Esto se identificó en las siguientes rutas de la aplicación web de la entidad financiera bajo análisis:

Tabla 12: Respuesta HTTPS almacenable en caché.

Asunto:	Respuesta HTTPS almacenable en caché
Gravedad:	**Información**
Confianza:	**Cierto**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/_siguiente/estático/media/icon-plus.92830e8b.svg**
Ruta:	**/_siguiente/estático/media/icono-megafono.0ccacf94.svg**
Ruta:	**/_next/static/media/icon-mask.e38ac5ad.svg**
Ruta:	**/_next/static/media/icon-shield.0331e344.svg**
Ruta:	**/_next/static/media/Metrópolis-Bold.f6c09cc3.woff2**
Ruta:	**/_siguiente/estático/media/a745ddc86d0c3d32-spwoff2**

Ruta:	**/_siguiente/estático/media/2aaf0723e720e8b9-spwoff2**
Ruta:	**/_next/static/media/5f2dbf67246008c1-spwoff2**
Ruta:	**/_siguiente/estático/media/bd82e0c1746bd560-spwoff2**

(Pérez y Molina. 2023. Respuesta HTTPS almacenable en caché).

4.6 A05:2021 Configuración de Seguridad Incorrecta

Una aplicación puede considerarse vulnerable en torno a la configuración de seguridad incorrecta o “Security Misconfiguration”, cuando esta presenta falta de “hardening” relacionado a la seguridad en cualquier parte del “stack” tecnológico, permisos incorrectamente configurados, funciones innecesarias habilitadas, como por ejemplo puertos o servicios, exceso de usuarios con privilegios, mala gestión de errores, funciones de seguridad deshabitadas o mal configuradas, los “Frameworks” de las aplicaciones no poseen o utilizan valores seguros o simplemente el software está desactualizado.

4.6.1 Vulnerabilidades relacionadas con la Configuración de Seguridad Incorrecta

Según la publicación del “Top 10” de “OWASP” las siguientes son las vulnerabilidades comunes identificadas que tiene relación con la configuración de seguridad incorrecta:

- CWE-2 7PK - Medio ambiente.
- CWE-11 Configuración incorrecta de ASP.NET: creación de un binario de depuración.
- CWE-13 Configuración incorrecta de ASP.NET: contraseña en el archivo de configuración.

- CWE-15 Control externo del sistema o ajuste de configuración.
- CWE-16 Configuración.
- CWE-260 Contraseña en el archivo de configuración.
- CWE-315 Almacenamiento de texto sin cifrar de información confidencial en una cookie.
- CWE-520 Configuración incorrecta de .NET: uso de suplantación de identidad.
- CWE-526 Exposición de Información Sensible a través de Variables Ambientales.
- CWE-537 Mensaje de error de Java Runtime que contiene información confidencial.
- CWE-541 Inclusión de información confidencial en un archivo de inclusión.
- CWE-547 Uso de constantes codificadas de forma rígida y relevantes para la seguridad.
- CWE-611 Restricción incorrecta de referencia de entidad externa XML.
- CWE-614 Cookie sensible en sesión HTTPS sin atributo 'seguro'.
- CWE-756 Falta la página de error personalizado.
- CWE-776 Restricción incorrecta de referencias de entidad recursiva en DTD ('Expansión de entidad XML').
- CWE-942 Política de dominio cruzado permisiva con dominios que no son de confianza.
- CWE-1004 Cookie sensible sin indicador 'HttpOnly'.
- CWE-1032 OWASP Top Ten 2017 Categoría A6 - Configuración incorrecta de seguridad.
- CWE-1174 Configuración incorrecta de ASP.NET: validación de modelo incorrecta.

4.6.2 Vulnerabilidades relacionadas con la Configuración de Seguridad Incorrecta que fueron identificadas en la solución denominada “CREDIWEB”

1. Se detectó que la depuración remota de ASP.NET está habilitada por defecto en el servidor, no obstante, para explorar esta vulnerabilidad es necesario contar con un roll que tenga permisos para ejecutar la depuración. Por lo que una escalada de privilegios y un ataque de fuerza bruta podrían materializar este factor de riesgo identificado.

   La siguiente tabla muestra la ruta y el archivo que genera el factor de riesgo mencionado:

Tabla 13: Depuración de ASP.NET habilitada

Asunto:	Depuración de ASP.NET habilitada
Gravedad:	**Información**
Confianza:	**Firme**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/Predeterminado.aspx**

(Pérez y Molina. 2023. Depuración de ASP.NET habilitada).

2. Se identificó que la raíz del aplicativo web es susceptible a ataques de secuencias de comandos entre sitios (XSS), esto debido a que no se identificaron encabezados en el sitio web que prevengan este tipo de ejecución sobre los “scripts” colocados en la página web.

   A continuación, se muestra la ruta que puede ser afectada por ataques de secuencias de comandos entre sitios (XSS):

Tabla 14: Filtro de secuencias de comandos entre sitios del navegador deshabilitado.

Asunto:	Filtro de secuencias de comandos entre sitios del navegador deshabilitado
Gravedad:	**Información**
Confianza:	**Cierto**

Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/**

(Pérez y Molina. 2023. Filtro de secuencias de comandos entre sitios del navegador deshabilitado).

3. Se identificaron dos *cookies*, las cuales fueron emitidas por la aplicación desde la raíz según el escaneo realizado por las distintas herramientas con las que se evaluó el sitio web. No se identificaron “tokens” dentro de estas, sin embargo, tampoco se detectó un indicador de tipo “HttpOnly”, el cual es un atributo definido para evitar que las aplicaciones ejecutadas del lado del cliente puedan acceder a las *cookies*, esto se pone como bandera para prevenir vulnerabilidades por ataques de secuencias de comandos entre sitios (XSS).

En la siguiente tabla se muestra el ámbito de las cookies identificadas, además de la carencia de la bandera “HttpOnly”:

Tabla 15: Cookie en el ámbito del dominio principal.

Asunto:	Cookie en el ámbito del dominio principal
Asunto:	**Cookie sin indicador HttpOnly establecido**
Gravedad:	**Información**
Confianza:	**Cierto**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/**
Cookies del dominio principal	**1P_JAR, AEC**
Cookies sin indicador HttpOnly	**1P_JAR**

(Pérez y Molina. 2023. Cookie en el ámbito del dominio principal).

4.7 A06:2021 Componentes Vulnerables y Desactualizados

La utilización de componentes vulnerables y desactualizados o “Vulnerable and Outdated Components” es frecuente, debido a que los grupos de desarrollo en su mayoría desconocen las versiones de todos los componentes que se utilizan o se utilizaron de forma directa o indirecta en el desarrollo de una solución.

Considerando que, si el software utiliza componentes desactualizados, estos pueden contribuir a la vulnerabilidad del aplicativo, debido a que estos componentes, Frameworks, o librerías carecen de soporte.

Por lo que se vuelve necesario realizar de forma recurrente búsquedas de vulnerabilidades relacionadas al tema para mitigar riesgos relacionados a la explotación por componentes vulnerables y desactualizados.

4.7.1 Vulnerabilidades relacionadas con los componentes vulnerables y desactualizados

Según la publicación del “Top 10” de “OWASP” las siguientes son las vulnerabilidades comunes identificadas que tiene relación con los componentes vulnerables y desactualizados:

- CWE-937 OWASP Top 10 2013: Uso de componentes con vulnerabilidades conocidas.
- CWE-1035 2017 Top 10 A9: Uso de componentes con vulnerabilidades conocidas.
- CWE-1104 Uso de componentes de terceros sin mantenimiento.

4.7.2 Vulnerabilidades relacionadas con los componentes vulnerables y desactualizados que fueron identificadas en la solución denominada “CREDIWEB”

1. Se identificaron 12 vulnerabilidades relacionadas a cinco librerías obsoletas de “JQuery” utilizadas dentro del aplicativo de “CREDIWEB”.

   A continuación, se detalla la versión de la librería utilizada y los CVEs identificados con estas:

   a. jquery versión 3.3.1.min:
      - CVE-2019-11358: jQuery anterior a 3.4.0, como se usa en Drupal, CMS de fondo y otros productos, maneja mal jQuery.extend(true, {}, ...) debido a la contaminación de Object.prototype.
      - CVE-2020-11022: Regex en su jQuery.htmlPrefilter a veces puede introducir XSS.
      - CVE-2020-11023: pasar HTML que contiene elementos <option> de fuentes no confiables, incluso después de desinfectar, a uno de los métodos de manipulación DOM de jQuery (es decir, .html(), .append() y otros) puede ejecutar código no confiable.

   b. jquery versión 3.4.1.min:
      - CVE-2020-11022: Regex en su jQuery.htmlPrefilter a veces puede introducir XSS.
      - CVE-2020-11023: pasar HTML que contiene elementos <option> de fuentes no confiables, incluso después de desinfectar, a uno de los métodos de manipulación DOM de jQuery (es decir, .html(), .append() y otros) puede ejecutar código no confiable.

   c. jquery-ui versión 1.13.1:
      - CVE-2022-31160: XSS al actualizar las casillas de verificación si los datos controlados por el usuario en las etiquetas.
      - CVE-2022-31160: XSS al actualizar una casilla de verificación con una etiqueta de texto inicial similar a HTML.

   d. jquery.datatables versión 1.10.22:
      - posible XSS
        - https://github.com/DataTables/Dist-DataTables/commit/59a8d3f8a3c1138ab08704e783bc52bfe88d7c9b

        - https://cdn.datatables.net/1.11.3/
    - prototipo de contaminación
        - https://github.com/DataTables/DataTablesSrc/commit/a51cbe99fd3d02aa5582f97d4af1615d11a1ea03
        - https://cdn.datatables.net/1.10.23/

e. Moment.js versión 2.24.0:
    - CVE-2022-24785: esta vulnerabilidad afecta a los usuarios de npm (servidor) de moment.js, especialmente si el usuario proporcionó una cadena de configuración regional, por ejemplo, fr se usa directamente para cambiar la configuración regional de moment.
    - CVE-2022-31129: Denegación de servicio de expresión regular (ReDoS), paquete de moment que afecta, versiones >=2.18.0 <2.29.4
.

La siguiente tabla muestra las rutas donde se identificaron las librerías desactualizadas que se están ejecutando en “CREDIWEB”:

Tabla 16: Dependencia de JavaScript vulnerable.

Asunto:	Dependencia de JavaScript vulnerable
Gravedad:	**Bajo**
Confianza:	**Tentativo**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/CREDIWEB/MantenimientoUsuariosMasivos**
Ruta:	**/CREDIWEB/CompromisoPago_Reporte**
Ruta:	**/Scripts/jquery-3.4.1.min.js**
Ruta:	/
Ruta:	**/Content/DataTablePaginacion/datatables.min.js**
Ruta:	**/Content/DateRangePicker/momento.min.js**
Ruta:	**/Contenido/plugins/jquery/jquery.min.js**

(Pérez y Molina. 2023. Dependencia de JavaScript vulnerable).

4.8 A07:2021 Fallas de Identificación y Autenticación

Las fallas relacionadas a la identidad y la autenticación o “Identification and Authentication Failures” están relacionadas a la inadecuada gestión de sesiones de usuario dentro del sistema o aplicativo que pueden desencadenar ataques de autenticación, por lo que se recomienda validar si la aplicación es susceptible a ataques automatizados como por ejemplo la fuerza bruta, utilizando diccionarios de usuarios y contraseñas.

Otro punto para considerar es la estructura de la conformación definida o establecida para las contraseñas, ya que las contraseñas de construcción débil o de longitudes inferiores a 12 dígitos son más propensas a romperse por ataque automatizados.

También se debe tomar en cuenta el método que se utiliza para la recuperación de contraseñas y la incorporación de múltiples factores de autenticación o el deshabilitar los usuarios por intentos fallidos de acceso al sistema.

4.8.1 Vulnerabilidades relacionadas con las fallas de identificación y autenticación

Según la publicación del “Top 10” de “OWASP” las siguientes son las vulnerabilidades comunes identificadas que tiene relación con las fallas de identificación y autenticación:

- CWE-255 Errores de administración de credenciales.
- CWE-259 Uso de contraseña codificada.

- CWE-287 Autenticación incorrecta.
- Omisión de autenticación CWE-288 usando una ruta o canal alternativo.
- CWE-290 Omisión de autenticación por suplantación de identidad.
- CWE-294 Omisión de autenticación por Capture-replay.
- CWE-295 Validación de certificado incorrecta.
- CWE-297 Validación incorrecta del certificado con discrepancia de host.
- Canal CWE-300 accesible por punto no final.
- Omisión de autenticación CWE-302 por datos supuestamente inmutables.
- CWE-304 Falta un paso crítico en la autenticación.
- CWE-306 Autenticación faltante para función crítica.
- CWE-307 Restricción incorrecta de intentos de autenticación excesivos.
- CWE-346 Error de validación de origen.
- Fijación de sesión CWE-384.
- CWE-521 Requisitos de contraseña débil.
- CWE-613 Caducidad de sesión insuficiente.
- CWE-620 Cambio de contraseña no verificado.
- CWE-640 Mecanismo de recuperación de contraseña débil para contraseña olvidada.
- CWE-798 Uso de Credenciales Codificadas.
- CWE-940 Verificación incorrecta de la fuente de un canal de comunicación.
- CWE-1216 Errores del mecanismo de bloqueo.

4.8.2 Vulnerabilidades relacionadas con las fallas de identificación y autenticación que fueron identificadas en la solución denominada “CREDIWEB”

Según los escaneos realizados con “Burp Suite”, “Zen Attack Proxy (ZAP)” y “Nikto”, no se identificaron vulnerabilidades relacionadas con fallas de identificación y autenticación.

Esto debido a que el acceso al sistema está controlado por el directorio activo, impidiendo vulnerabilidades relacionadas con ataques de diccionario, por otra parte, se presentan políticas de cambio de contraseña de forma mensual.

Otra característica que se consideró es la utilización de contraseñas con longitudes mayores a catorce dígitos, con una estructura alfanumérica no secuencial (no se admiten contraseñas con secuencias de números o letras seguidas).

4.9 A08:2021 Fallas en el Software y en la Integridad de los Datos

Los fallos de integridad del software y de los datos o “Software and Data Integrity Failures”, están relacionados con código e infraestructura desprotegidos contra alteraciones lo que expone la pérdida de integridad en el sistema.

Esto puede provocar acceso no autorizado, la inclusión de código malicioso o comprometer la infraestructura en general del sistema.

También existe el riesgo de descargar actualizaciones de forma automática las cuales no son verificadas lo que aumenta el factor de riesgo.

4.9.1 Vulnerabilidades relacionadas con las fallas en el software y en la integridad de los datos

Según la publicación del “Top 10” de “OWASP” las siguientes son las vulnerabilidades comunes identificadas que tiene relación con las fallas en el software y en la integridad de los datos:

- CWE-345 Verificación insuficiente de la autenticidad de los datos.
- CWE-353 Falta soporte para verificación de integridad.
- CWE-426 Ruta de búsqueda no confiable.
- CWE-494 Descarga de código sin comprobación de integridad.
- CWE-502 Deserialización de datos no confiables.
- CWE-565 Dependencia de cookies sin verificación de validación e integridad.

- CWE-784 Dependencia de cookies sin verificación de validación e integridad en una decisión de seguridad.
- CWE-829 Inclusión de funcionalidad de esfera de control no confiable.
- CWE-830 Inclusión de funcionalidad web de una fuente no confiable.
- CWE-915 Modificación controlada incorrectamente de atributos de objetos determinados dinámicamente.

4.9.2 Vulnerabilidades relacionadas con las fallas en el software y en la integridad de los datos que fueron identificadas en la solución denominada “CREDIWEB”

1. Se identificó dentro de la aplicación web una serie de “scripts” que acceden a dominios externos, estos dominios son ejecutados dentro del contexto de seguridad de la aplicación, no obstante, hacen llamados a sitios fuera del ámbito de seguridad.

   Los “scripts” identificados con este tipo de comportamiento son los siguientes:

   - https://cdnjs.cloudflare.com/ajax/libs/xls/0.7.4-a/xls.core.min.js
   - https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.7.7/xlsx.core.min.js
   - https://code.jquery.com/jquery-3.6.0.js
   - https://code.jquery.com/ui/1.13.1/jquery-ui.js
   - https://activos-prod.sumo.prod.webservices.mozgcp.net/static/354.3a7ddcb120703df0.js
   - https://activos-prod.sumo.prod.webservices.mozgcp.net/static/672.2aa5f418a3ec2171.js
   - https://activos-prod.sumo.prod.webservices.mozgcp.net/static/901.5c1bf4c53aef45f5.js
   - https://activos-prod.sumo.prod.webservices.mozgcp.net/static/common.6968c88d57044d85.js

- https://activos-prod.sumo.prod.webservices.mozgcp.net/static/common.fx.download.7c161292b0ad1beb.js
- https://activos-prod.sumo.prod.webservices.mozgcp.net/static/document.b5035c05ca0ac300.js
- https://assets-prod.sumo.prod.webservices.mozgcp.net/static/gtm-snippet.c572a96f8b784ca0.js
- https://assetsprod.sumo.prod.webservices.mozgcp.net/static/jsi18n/es/djangojs-min.js

Estos “scripts” se han detectado en la raíz del aplicativo web y en un formulario de mantenimiento de usuarios, como lo demuestra la siguiente tabla:

Tabla 17: Secuencia de comandos entre dominios.

Asunto:	Secuencia de comandos entre dominios
Gravedad:	**Información**
Confianza:	**Cierto**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/ CREDIWEB / MantenimientoUsuariosMasivos**
Ruta:	**/**

(Pérez y Molina. 2023. Secuencia de comandos entre dominios).

4.10 A09:2021 Fallas en el Registro y Monitoreo

El apartado de fallas en el registro y monitoreo tiene como objetivo apoyar en la detección y escalamiento de incidentes de seguridad.

Una mala gestión de monitoreo y un inadecuado registro de eventos puede incrementar el riesgo de vulnerabilidades o brechas de seguridad no detectadas de forma oportuna.

Estas falencias aumentan significativamente el riesgo de fuga o pérdida de información, por lo que se recomienda que se registren eventos como los inicios de sesión, fallas en el inicio de sesión y transacciones de alto valor, además de advertencias y errores generados por las aplicaciones.

Otro punto que se debe considerar es donde se almacenarán estos registros, ya que no es recomendable tenerlos de forma local.

Es importante tener un adecuado proceso de escalamiento para los eventos de ciberseguridad y seguridad de la información, este debe estar oficializado y contar un cronograma de pruebas de penetración ya que los escaneos realizados por medio de herramientas como “OWASP ZAP” no generan alertas de forma dinámica.

4.10.1 Vulnerabilidades relacionas con las fallas en el registro y monitoreo

Según la publicación del “Top 10” de “OWASP” las siguientes son las vulnerabilidades comunes identificadas que tiene relación con las fallas en el registro y monitoreo:

- CWE-117 Neutralización de salida incorrecta para troncos.
- CWE-223 Omisión de información relevante para la seguridad.
- CWE-532 Inserción de información confidencial en el archivo de registro.
- CWE-778 Registro insuficiente.

4.10.2 Vulnerabilidades relacionadas con las fallas en el registro y monitoreo que fueron identificadas en la solución denominada “CREDIWEB”

Debido al alcance del trabajo de investigación y los objetivos definidos para este, las fallas en el registro y monitoreo no fueron consideradas.

Esto debido a que la entidad financiera limitó el ámbito de acción únicamente al descubrimiento de vulnerabilidades y riesgos relacionados con la aplicación denominada “CREDIWEB”.

Esto excluye el análisis de la infraestructura, por tal motivo fue que se brindó un entorno controlado para la ejecución de las pruebas realizadas sobre la aplicación web.

4.11 A10:2021 Falsificación de Solicitudes del Lado del Servidor (SSRF)

La falsificación de solicitudes del lado del servidor (SSRF) o “Falsificación de Solicitud del Lado del Servidor (SSRF)”, ocurre cuando las aplicaciones web utilizan u obtienen recursos remotos sin una validación de seguridad previa.

Esto crea una vulnerabilidad que permite a los atacantes enviar solicitudes de falsificación, pasando la seguridad de “firewalls”, “VPNs” o listas de control de acceso (CLS).

Este escenario se ha vuelto más común en las aplicaciones web modernas y es un riesgo que se considera que va en aumento debido a la computación en la nube y la complejidad de las arquitecturas empresariales.

4.11.1 Vulnerabilidades relacionadas con la falsificación de solicitudes del lado del servidor (SSRF)

Según la publicación del “Top 10” de “OWASP” las siguientes son las vulnerabilidades comunes identificadas que tiene relación con la falsificación de solicitudes del lado del servidor (SSRF):

- CWE-918 Falsificación de solicitud del lado del servidor (SSRF).

4.11.2 Vulnerabilidades relacionadas con l la falsificación de solicitudes del lado del servidor (SSRF), que fueron identificadas en la solución denominada “CREDIWEB”

Según los diferentes análisis realizados sobre la aplicación web denominada “CREDIWEB”, no se logró identificar ningún evento relacionado a la explotación de vulnerabilidades relacionadas a la falsificación de solicitudes del lado del servidor (SSRF).

4.12 Otros Encontrados y no Contemplados en el “Top 10” de “OWASP” 2021

1. Se identificó que la aplicación web denominada como “CREDIWEB” puede ser vulnerable a la denegación de servicios basada en DOM, esto ocurre desde los datos que se leen desde el “input.value” y luego estos son transmitidos a “RegExp”.

Esto debido a que la vulnerabilidad se aprovecha de un “script” que se ejecuta del lado del cliente, permitiendo controlar parte del DOM, lo cual permite un procesamiento de datos no segura.

En la siguiente tabla se muestran las rutas con “scripts” susceptibles a esta vulnerabilidad:

Tabla 18: Denegación de servicio basado en DOM.

Asunto:	Denegación de servicio (basado en DOM)
Gravedad:	**Información**
Confianza:	**Firme**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/ CREDIWEB / MantenimientoUsuariosMasivos**
Ruta:	**/ CREDIWEB / MantenimientoUsuariosMasivos**

Ruta:	**/ CREDIWEB / AsignarCaso**

(Pérez y Molina. 2023. Denegación de servicio basado en DOM).

2. Se identificó que la aplicación puede ser susceptible al registro o captura de clics, debido a que no se han definido encabezados HTTP que controlen de forma adecuada este tipo de eventos o ataques.

   Esto debido a que no se identificaron cabeceras HTTP como “X-Frame-Options” o “Content-Security-Policy”, lo que puede permitir la explotación de los “IFrames” cargados en el aplicativo web, esto incrementa el riesgo a sufrir este tipo de amenaza.

   A continuación, se muestra el detalle de la ruta:

Tabla 19: posible secuestro de clics.

Asunto:	enmarcable (posible secuestro de clics)
Gravedad:	**Información**
Confianza:	**Firme**
Anfitrión:	**http://---.---.---.--- (Confidencial)**
Ruta:	**/**

(Pérez y Molina. 2023. posible secuestro de clics).

A continuación, se describe un resumen de los hallazgos identificados por el grupo investigador:

Para el apartado relacionado a la Pérdida de Control de Acceso, se identificaron 3 vulnerabilidades, una gravedad de nivel medio y dos de tipo informativas.

Para el apartado relacionado a las Fallas Criptográficas, se identificó una vulnerabilidad de nivel bajo.

Para el apartado relacionado a la inyección de código, se identificaron cuatro vulnerabilidades, entre las cuales una era de gravedad de nivel alto, una de gravedad baja y dos de tipo informativas.

Para el apartado relacionado al Diseño Inseguro, se identificaron tres vulnerabilidades, de las cuales una era de gravedad de nivel alto y dos de tipo informativas.

Para el apartado relacionado a la Configuración de Seguridad Incorrecta, se identificaron tres vulnerabilidades de tipo informativas.

Para el apartado relacionado a Componentes Vulnerables y Desactualizados, se identificó una vulnerabilidad de nivel bajo.

Para el apartado relacionado a Fallas de Identificación y Autenticación, no se identificaron vulnerabilidades.

Para el apartado relacionado a Fallas en el Software y en la Integridad de los Datos se identificó una vulnerabilidad de tipo informativa.

Para el apartado relacionado a Fallas en el Registro y Monitoreo, no se identificaron fallas, esto debido a la limitación del alcance.

Para el apartado relacionado a la Falsificación de Solicitudes del Lado del Servidor (SSRF), no se identificaron vulnerabilidades.

Para el aparato denominado como “Otros Encontrados y no Contemplados" en el “Top 10” de “OWASP” 2021” se identificaron dos vulnerabilidades de tipo informativas.

Capítulo 5. Propuesta de Solución

La siguiente propuesta de solución se presenta con base a los resultados obtenidos en el capítulo anterior, donde se ejecutaron las herramientas de diagnóstico mencionadas para identificar algunas vulnerabilidades del aplicativo.

La propuesta contempla una serie de recomendaciones generales y otras específicas para el sistema CREDIWEB. Las recomendaciones generales pueden ser aplicadas a diversos aplicativos webs, sin embargo, las propuestas específicas están enfocadas directamente a las vulnerabilidades encontradas en la aplicación. A continuación, se mencionan las recomendaciones emitidas, estas están ordenadas según la categorización del top 10 de OWASP:

5.1 A01:2021 Pérdida de Control de Acceso

5.1.2 Prevenciones generales relacionadas a la Pérdida de Control de Acceso:

El control de acceso solo es efectivo en el código de confianza del lado del servidor o en la API sin servidor, donde el atacante no puede modificar la comprobación de control de acceso o los metadatos, por lo que es importante implementar mecanismos de control de acceso en toda la aplicación, incluyendo la minimización del uso de Cross-Origin Resource Sharing (CORS). Además, los controles de acceso del modelo deben cumplir la propiedad del registro en lugar de aceptar que el usuario pueda crear, leer, actualizar o eliminar cualquier registro. Además, se debe definir un ámbito de ejecución específico para las aplicaciones únicas.

- Lista de directorios: Se debe deshabilitar la lista de directorios del servidor web y asegúrese de que los metadatos de los archivos (por ejemplo, .git) y los archivos de copia de seguridad no estén presentes dentro de las raíces

web.

- Generación de Registros: Se deben generar registros de errores de control de acceso, generando alertas a los administradores cuando sea apropiado (por ejemplo, errores recurrentes).

- Velocidad de acceso al API: Se debe considerar el reducir el límite de velocidad de acceso a la API y de esta forma minimizar el daño de las herramientas de ataque automatizado.

- Validación de identificadores de sesión: Se debe procurar invalidar los identificadores de sesión utilizados en el servidor después de cerrar la sesión. Los tokens JWT sin estado deberían ser de corta duración para que se minimice la ventana de oportunidad para un atacante. Para los JWT de mayor duración, se recomienda encarecidamente seguir los estándares de OAuth para revocar el acceso.

- Unidad de control de acceso y pruebas: Se debe definir un procedimiento para que los desarrolladores y el personal de control de calidad incluyan la unidad de control de acceso funcional y las pruebas de integración.

5.1.2 Prevenciones específicas relacionadas al Acceso y Control

- Principio del menor privilegio: La aplicación debe estar enfocada en el principio del menor privilegio, otorgando a los usuarios y componentes solo los permisos mínimos necesarios para llevar a cabo sus tareas. No se recomienda otorgar permisos excesivos a los usuarios que puedan abrir puertas a posibles vulnerabilidades.

- Autenticación sólida: Se deben implementar métodos robustos de autenticación como por ejemplo un Active Directory en la plataforma web, u otras como hashes de contraseñas seguras, autenticación de Múltiple factor (MFA) o autenticación basada en tokens.

- Autorización basada en roles: Se requiere definir roles y permisos específicos para diferentes tipos de usuarios. Se recomienda asegurar que cada usuario tenga acceso solo a las partes de la aplicación que son relevantes para su rol.

- Validación de entrada: Se debe validar y limpiar la entrada de datos del usuario para evitar ataques de inyección (como SQL o XSS). Usa bibliotecas de validación o marcos de trabajo para asegurarte de que los datos sean seguros antes de procesarlos.

- Control de sesiones: Se deben gestionar las sesiones de manera segura y con un tiempo adecuado de validez. Se recomienda utilizar tokens de sesión y asegurarse de que las sesiones caduquen después de un período de inactividad.

- Prevención de CSRF: Se deben implementar protecciones contra ataques de falsificación de solicitudes HTTP entre sitios (CSRF) utilizando tokens CSRF y validando las referencias cruzadas en las solicitudes.

- Segregación de datos: Se recomienda aislar y proteger los datos en diferentes repositorios utilizando técnicas de separación de datos, como bases de datos separadas para diferentes tipos de información o utilizar Clúster para dicho manejo. (AVAMAR, NETWORKER)

- Auditorías y trazabilidad: Según las consultas realizadas sobre alguna otra gestión de evaluación del sistema, no se cuenta con registros anteriores de monitoreo general sobre la plataforma, por lo cual se recomienda crear una actividad de monitoreo programado sobre el sistema en general en búsqueda de actividades sospechosas.

- Mejorar los Logs y registros de actividad del sistema: En el análisis ejecutado, el sistema no logró identificar los ingresos no permitidos o reportar dicha actividad a los administradores del sistema, por lo que se

recomienda revisar la configuración de los registros del aplicativo para prevenir que esto se vuelva a presentar.

- Actualizaciones y parches: El análisis reveló que se encuentran diferentes componentes con versiones desactualizadas dentro del código, por ejemplo, las bibliotecas y marcos de trabajo, así mismo se recomienda realizar un análisis de versiones en el equipo que contiene el IIS del aplicativo y validar los parches de seguridad para evitar vulnerabilidades conocidas, por lo que se recomienda actualizar estas librerías y definir un procedimiento de revisión de código fuente.

- Pruebas de seguridad: Se recomienda realiza pruebas de seguridad, como pruebas de penetración y escaneos de vulnerabilidades, de forma regular, para identificar y abordar posibles puntos débiles en la aplicación.

- Inversión de entrenamiento en seguridad: Este ítem es indispensable aplicarlo dentro de la institución ya que un adecuado desarrollo del personal en temas de seguridad genera un mejor enfoque sobre las mejores prácticas de seguridad en programación y la importancia de implementar políticas de acceso y control adecuadas, por lo que se recomienda que los desarrolladores cuenten con capacitación relacionada al desarrollo de software seguro (DevSecOps).

- Revisión de código: Se recomienda definir un procedimiento para la revisión de código por pares, con el fin de identificar posibles problemas de seguridad y asegurarte de que las políticas de acceso y control se implementan correctamente.

- Gestión de Cambios: Debe existir un procedimiento para controlar y revisar los cambios implementados en el código fuente a nivel de desarrollo. Los cambios deben ser revisados y aprobados por un supervisor antes de su implementación.

- Respuesta a Incidentes: Se debe definir un plan de respuesta a incidentes de seguridad de la información y ciberseguridad, esto para contar con una guía detallada de los pasos a seguir ante una eventual violación de acceso.

- Eliminación de datos corporativos dentro del Código: Se recomienda eliminar cualquier dirección de correo electrónico o información que no sea necesaria. Si esta práctica es requerida por el negocio se recomienda reemplazar las direcciones personales con direcciones de buzón de correo anónimas (como helpdesk@example.com).

- Correo Spam: Para reducir la cantidad de spam enviado a direcciones de buzones anónimos, considere ocultar la dirección de correo electrónico y, en su lugar, proporcione un formulario que se genere del lado del servidor de correo electrónico, además proteja este con un CAPTCHA si es necesario.

- Manejo adecuado de Entidades: No se recomienda transmitir información confidencial dentro de la cadena de consulta de URL, dado esto, se debe modelar el traslado de información por medios de clases entre las diferentes capas del sistema.

5.2. A02:2021 Fallas Criptográficas

5.2.1 Prevenciones generales relacionadas a Fallas Criptográficas

Clasificación de la Información: Se deben clasificar los datos procesados, almacenados o transmitidos por la aplicación, por lo que se requiere Identificar cuales datos son confidenciales de acuerdo con las leyes de privacidad, los requisitos reglamentarios o las necesidades comerciales. Por esto no se debe almacenar datos confidenciales innecesariamente. Para esto lo mejor es desechar estos lo antes posible o usar la tokenización compatible con PCI DSS o incluso el truncamiento.

Uso del cifrado para datos: Se debe asegurar el cifrado de todos los datos confidenciales en reposo, por lo tanto, se debe asegurar que los algoritmos, protocolos y claves estándar estén actualizados y sean algoritmos de cifrados sólidos. Otro aspecto para considerar es el cifrar todos los datos en tránsito con protocolos seguros como TLS con cifrados de secreto directo (FS), priorización de cifrado por parte del servidor y parámetros seguros, Aplicando el cifrado a directivas como HTTP Strict Transport Security (HSTS).

- Desactivar el caché: Se debe desactivar el almacenamiento en caché para las respuestas que contengan datos confidenciales.

- Uso de protocolos: Se deben evitar los protocolos heredados como FTP y SMTP para transportar datos confidenciales.

- Almacenamiento de contraseñas: Se recomienda utilizar funciones de hash fuertes y confiables para el almacenamiento de contraseñas como, por ejemplo: SHA-256, SHA512, Argon2.

- Autenticación y cifrado: Se debe utilizar autenticación de cifrado en lugar de solo el cifrado, además, se debe garantizar que las claves criptográficas se generan al azar y se almacenen en la memoria como matrices de bytes. También es importante considerar que se debe asegurar la aleatoriedad del vector de inicialización para que la función criptográfica no permita el sembrado de una semilla que sea predecible o con baja entropía, por último, se recomienda evitar las funciones criptográficas obsoletas y los esquemas de relleno, como MD5, SHA1, PKCS número 1 v1.5.

5.2.2 Prevenciones específicas relacionadas a Fallas Criptográficas

- Elección adecuada de algoritmos criptográficos: La aplicación web utiliza un logaritmo de encriptación de SHA-128, por lo cual hoy no es seguro, se recomienda utilizar algoritmos criptográficos confiables y ampliamente aceptados que sean apropiados para el uso específico. Por lo tanto, se recomienda sustituir estos algoritmos obsoletos o débiles que puedan estar propensos a ataques, por cifrados más seguros como SHA-256, SHA512, Argon2.

- Gestión adecuada de claves: Se debe almacenar las claves de manera segura utilizando soluciones como sistemas de administración de claves (KMS) o hardware de seguridad. El aplicativo almacenas cadenas de conexión a bases de datos y llaves de encriptación fundamentales para la seguridad de este, la recomendación es no almacenar claves en texto plano o dentro del código fuente.

- Longitud y complejidad de claves: Se deben utilizar claves largas y complejas para hacer más difícil el descifrado por fuerza bruta. Las claves cortas son más susceptibles a ataques de fuerza bruta o búsqueda exhaustiva.

- Salting para almacenamiento de contraseñas: Se recomienda utilizar gestores de contraseñas que utilicen un valor aleatorio conocido como "salt" antes de aplicar una función hash. Esto hace que las contraseñas almacenadas sean menos susceptibles a ataques de tablas arco iris.

- Validación de cifrado: Se debe asegurar que la herramienta cuenta con al menos 1 único proceso de encriptación con el fin de validar la integridad del mensaje cifrado al descifrarlo. La aplicación según el análisis de vulnerabilidades cuenta con más de un método de encriptación.

- Pruebas de seguridad: Se deben realiza pruebas de penetración y análisis de vulnerabilidades para identificar de forma recurrente para identificar posibles debilidades en la implementación criptográfica.

- Actualizaciones regulares: Se debe mantener un plan de actualización documentado de las bibliotecas criptográficas y marcos de trabajo para asegurarte de que estén libres de vulnerabilidades conocidas.

- Revisiones de código: Realiza revisiones periódicas del código fuente en busca de errores de implementación criptográfica.

- Validación de datos de entrada: Se debe contar con métodos de validación y limpieza de datos antes de cifrar o descifrar datos, para evitar posibles ataques de inyección u otros problemas.

- Documentación clara: Debe existir documentación del cómo se están utilizando los algoritmos criptográficos en tu aplicación. Esto facilitará futuras revisiones y auditorías.

5.3 A03:2021 Inyección

5.3.1 Prevenciones generales relacionadas a la Inyección

- Uso de capas: Se recomienda utilizar una capa intermedia segura, que genera un aislamiento de la capa de presentación y la capa del modelo al aplicativo, evitando el uso del intérprete por completo, proporcionando una interfaz parametrizada o migrar a Object Relational Mapping Tools (ORM). Considere que incluso teniendo parametrizados, los procedimientos almacenados, se puede introducir la inyección SQL si PL/SQL o T-SQL, ya que estos concatenan consultas y datos o ejecutan datos hostiles con EXECUTE IMMEDIATE o exec().

- Validación de entrada positiva: Se debe utilizar la validación de entrada positiva del lado del servidor, con el objetivo de poder gestionar cualquier consulta dinámica residual que contenga caracteres especiales de escape.

- Uso de nomenclatura: Se recomienda definir una nomenclatura diferente para los componentes de los formularios con relación al tipado de los datos en las clases del sistema, esto con el fin de proteger la integridad de las bases de datos.

- Uso de LIMIT: Considere utilizar LIMIT y otros controles adicionales dentro de las consultas SQL, para evitar la divulgación masiva de registros en caso de una inyección.

5.3.2 Prevenciones específicas relacionadas a Inyección

- Validación y saneamiento de entradas: Se debe validar y limpiar exhaustivamente todas las entradas de usuario antes de utilizarlas en cualquier operación o consulta. Utilizar listas blancas (aceptando solo caracteres válidos) en lugar de listas negras (bloqueando caracteres no válidos) para evitar omisiones.

- Parámetros de consulta preparados: Se debe considerar utilizar únicamente parámetros preparados o consultas parametrizadas al interactuar con bases de datos u otras fuentes de datos. Esto evita la concatenación de datos de usuario en consultas SQL o de otro tipo, se debe utilizar métodos más seguros como consultas parametrizadas o generadores de consultas ORM.

- Validación de datos de entrada: Se debe válida y filtra todas las entradas del usuario, incluidos campos como URLs, cookies, cabeceras HTTP, etc., para evitar posibles inyecciones.

- Validación en el lado del servidor: Se debe evitar realizar validaciones de lado del cliente, únicamente se deben realizar validaciones o verificaciones en el lado del servidor para evitar posibles manipulaciones de código.

- Aplicación del principio de mínimo privilegio: Se debe otorga a los componentes de la aplicación los permisos mínimos necesarios para

realizar sus tareas. Esto reduce el impacto de posibles inyecciones.

- Protección contra Cross-Site Scripting (XSS): Utilice una sanitización adecuada y el uso de funciones seguras de renderización en tu lenguaje de programación para evitar la inyección de scripts en la salida HTML.
- Actualizaciones y parches: Se deben mantener los componentes y bibliotecas actualizados para asegurarte de que estén libres de vulnerabilidades conocidas, además se debe definir un procedimiento de revisión y actualización de estos.

- Pruebas de seguridad: Se deben realiza pruebas de penetración y análisis de vulnerabilidades recurrentes para identificar posibles puntos de inyección y otros problemas de seguridad.

- Formación en seguridad: Se debe capacitar al equipo de desarrollo en DevSecOps con el fin de obtener un software más seguro mediante la ejecución e implementación de mejores prácticas de seguridad y comprensión de las vulnerabilidades de inyección.

- Monitorización y registro: Se recomienda implementar mecanismos de monitorización y registro para detectar actividades inusuales y potenciales intentos de inyección.

5.4 A04:2021 Diseño Inseguro

5.4.1 Prevenciones generales para el Diseño Inseguro

- Establecer y definir un ciclo de desarrollo seguro: Se debe definir un ciclo de desarrollo apoyado por profesionales en seguridad de aplicaciones para evaluar y diseñar la seguridad y controles relacionados con la privacidad.

- Uso de Patrones: Se debe establecer y utilizar un catálogo de patrones seguros o lineamientos de desarrollo como bases para ser utilizados como

guía en el proceso de desarrollo del software.

- Análisis de riesgos: Se debe contar con un análisis de riesgos relacionados al proceso de autenticación, control de acceso, lógica de negocio y otros, para implementar controles de seguridad en los procesos de desarrollo, integrando el lenguaje y los controles de seguridad en conjunto con las historias de usuario. Considerando integrar las verificaciones de viabilidad en cada capa de su aplicación (desde el frontend al backend).

- Ejecución de pruebas unitarias: Se debe definir un proceso para la ejecución de pruebas unitarias y de integración para validar que todos los flujos críticos son resistentes al modelo de amenazas, donde se recopile información sobre los casos de uso y casos de mal uso para cada capa de la aplicación y limitar el consumo de recursos por usuario o servicio.

5.4.2 Prevenciones específicas relacionadas al Diseño Inseguro

- Análisis de riesgos: Se debe realizar un análisis de riesgos para identificar posibles amenazas y vulnerabilidades potenciales en todas las etapas del diseño utilizando metodologías ágiles para evitar reprocesos. Esto te permitirá tomar decisiones informadas sobre la arquitectura y diseño.

- Separación de componentes: Se recomienda separar los componentes de la aplicación en capas y módulos distintos. Esto reduce la superficie de ataque y facilita el mantenimiento y la auditoría.

- Principio de defensa en profundidad: Se deben aplicar múltiples capas de seguridad en el diseño, considerando WAF (Cortafuegos de Aplicaciones Webs), sistemas de detección de intrusos IDS (Sistema de detección de Intrusos) y otras medidas de seguridad como IPS (Sistema de prevención de Intrusos) para mitigar diferentes tipos de amenazas.

- Modelos de amenazas: Se debería utiliza modelos de amenazas para identificar posibles escenarios de ataque y tomar decisiones de diseño que mitiguen esas amenazas.

- Gestión de ciclos de vida de seguridad: Se recomienda integra la seguridad en todas las etapas del ciclo de vida de desarrollo de software, desde el diseño hasta la implementación y el mantenimiento continuo.

5.5 A05:2021 Configuración de Seguridad Incorrecta

5.5.1 Prevenciones generales relacionadas a Configuración de Seguridad Incorrecta

- Entornos de trabajo: Los entornos de desarrollo, control de calidad y producción deben configurarse de forma idéntica, con diferentes credenciales utilizadas en cada uno. Este proceso debe automatizarse para minimizar el esfuerzo necesario para configurar un nuevo entorno seguro.

- Plataforma: Se debería establecer una plataforma mínima sin funciones, componentes, documentación ni ejemplos innecesarios. Elimine o no instale características y marcos de trabajo (Frameworks) no utilizados. (No instalar más de lo que se necesita)

- Configuración: Revisar y actualizar las configuraciones establecidas en los servidores web, con el fin de asegurar el correcto funcionamiento de los dispositivos que intervienen en el aplicativo web para mitigar el riesgo de intromisión al entorno.

- Directivas de seguridad: Establecer el envío de directivas de seguridad a los clientes finales, con el fin de mitigar ataques de denegación de protocolo y secuestro de Cookies en los puntos de conexión de los protocolos HTTP Y HTTPS.

- Automatización y gestión de configuraciones (RPA): Se recomienda utilizar herramientas de automatización y gestión de configuraciones para asegurarte de que las configuraciones sean consistentes y sigan las mejores prácticas de seguridad.

5.5.2 Prevenciones específicas relacionadas a la Configuración de Seguridad Incorrecta

- Configuraciones seguras: Se deben revisar y ajustar las configuraciones en los diferentes servicios que integran la plataforma web, con el propósito de detectar configuraciones por defecto. Las configuraciones predeterminadas a menudo son conocidas por atacantes y pueden ser explotadas.

- Desactivación de servicios no utilizados: Se recomienda desactivar cualquier otro servicio, puerto o característica que no se esté utilizando (FTP, SMTP, TELNET, entre otros). Esto reduce la superficie de ataque y minimiza el riesgo de configuraciones inseguras.

- Control de acceso: Se debe asegurar de que solo las personas autorizadas tengan acceso a las configuraciones y paneles de administración. Utiliza autenticación fuerte y autorización adecuada.

- Pruebas de configuración: Se recomienda realiza pruebas de intrusión contra la configuración de seguridad para identificar posibles brechas de seguridad en la configuración del sistema.

- Protección contra enumeración: Se debe evitar la enumeración de recursos y usuarios mediante el envío de mensajes de error personalizados para evitar revelar información sensible del sistema o la infraestructura.

- Segregación de ambientes: Se debe separar ambientes de desarrollo, pruebas y producción. Se debe asegurar que las configuraciones sean

adecuadas para cada ambiente y que no se compartan datos confidenciales.

- Controles de seguridad en el código: Se deben implementar controles de seguridad en el código para garantizar que las configuraciones sean aplicadas correctamente. Verifica que las configuraciones se adhieran a las políticas de seguridad.

- Auditorías y revisiones regulares: Se deberían realiza auditorías y revisiones periódicas de las configuraciones de tus sistemas para identificar posibles problemas de seguridad.

- Monitorización continua: Se recomienda implementa sistemas de monitorización para detectar cambios no autorizados en las configuraciones y alertar sobre posibles problemas de seguridad.

5.6 A06:2021 Componentes Vulnerables y Desactualizados

5.6.1 Prevenciones generales relacionadas a Componentes Vulnerables y Desactualizados

- Revisión de librerías y componentes obsoletos: se debería eliminar las dependencias que no son utilizadas, al igual que las funcionalidades, componentes, archivos y documentación innecesarios.

- Inventario de componentes: Es recomendable realizar un inventario continuo de las versiones de los componentes en el cliente y en el servidor (por ejemplo, Frameworks, bibliotecas) y sus dependencias utilizando herramientas como: versiones, OWASP Dependency Check, retire.js, etc. Además, se debe definir un control de seguimiento para las bibliotecas y los distintos componentes que no se actualicen de forma automática, debido a que estos no generan parches de seguridad.

- Monitoreo y supervisión: Se debe supervisar continuamente fuentes como Common Vulnerability and Exposures (CVE) y National Vulnerability Database (NVD) para detectar vulnerabilidades en los componentes. Se recomienda suscríbase para recibir alertas por correo electrónico sobre vulnerabilidades de seguridad relacionadas con los componentes que utiliza. También es recomendable elaborar un plan un plan continuo de monitoreo para poder clasificar y aplicar actualizaciones o cambios de configuración durante la vida útil de la aplicación o portafolio de aplicaciones.

- Uso de componentes: Se deben utilizar únicamente componentes como por ejemplo plugins, bibliotecas o Frameworks de fuentes oficiales, esto a través de enlaces seguros. Por lo que se deben utilizar solo paquetes firmados y de fuentes confiables para reducir la posibilidad de incluir un componente malicioso modificado.

5.6.2 Prevenciones específicas relacionadas a Componentes Vulnerables y Desactualizados

- Inventario de componentes: Se debe realizar un inventario exhaustivo de todos los componentes y bibliotecas de terceros utilizados en la aplicación WEB. Esto ayudará a identificar y rastrear los componentes vulnerables.

- Auditorías de seguridad de componentes: Es recomendable realizar auditorías de seguridad de las bibliotecas y componentes que se utilizan para identificar posibles vulnerabilidades y problemas de seguridad.

- Utilización de herramientas de escaneo: Se recomienda utilizar de forma periódica herramientas de escaneo de vulnerabilidades para identificar componentes desactualizados y con vulnerabilidades conocidas en la aplicación.

- Gestión de dependencias: Se deben utilizar administradores de dependencias que permitan automatizar la gestión y actualización de componentes, como por ejemplo el gestor de paquetes NUGET. Esto facilitará mantener tu aplicación actualizada.

- Seguimiento de problemas de seguridad: Debe existir un repositorio con los registros de los problemas de seguridad conocidos y las soluciones para los componentes que utilizas.

- Pruebas de calidad: Se debe realizar pruebas exhaustivas para asegurar que no se haya introducido ninguna vulnerabilidad o problema de funcionamiento.

5.7 A07:2021 Fallas de Identificación y Autenticación

5.7.1 Prevenciones generales relacionadas con Fallas de Identificación y Autenticación

- Implementar la autenticación Multi-factor: Esto se debe implementar para evitar ataques automatizados de reutilización de credenciales conocidas, fuerza bruta y reúso de credenciales robadas.

- credenciales por defecto: Se debe evitar utilizar credenciales por defecto, particularmente para usuarios administradores en el software además de implementar restricciones de uso contra contraseñas débiles, definiendo de longitud, complejidad y rotación de las contraseñas para verificar que una nueva contraseña o la utilizada en el cambio de contraseña no esté incluida en la lista de las 10,000 peores contraseñas.

- Recuperación de credenciales: Se debe asegurar que, el registro, la recuperación de las credenciales y el uso de APIs, no permiten los ataques de enumeración de usuarios, mediante la utilización de los mismos mensajes genéricos en todas las salidas. Por otra parte, se puede

implementar la función de limitar o incrementar el tiempo de espera entre intentos fallidos de inicio de sesión, teniendo en cuenta las medidas para evitar crear un escenario de denegación de servicio.

- Registro de intentos fallidos: Se deben registrar todos los intentos de ingresos fallidos y notificar a los administradores cuando se detectan ataques de rellenos automatizados de credenciales, fuerza bruta u otros.

- Gestor de sesión en el servidor: Se recomienda utilizar un gestor de sesión en el servidor, integrado, seguro y que genere un nuevo ID de sesión aleatorio con alta entropía después de iniciar sesión.

- Identificadores de sesión: Se debe evitar incluir los identificadores de sesión en la URL, deben almacenarse de forma segura y deben ser invalidados después del cierre de sesión, luego de un tiempo de inactividad o por un tiempo de espera absoluto.

5.7.2 Prevenciones específicas relacionadas a Fallas de Identificación y Autenticación

- Directivas de contraseña fuertes: Se requiere establecer políticas de seguridad de la información propias de la institución, con el fin de usar contraseñas más seguras.

- Autenticación de dos factores (2FA): Se debe implementar al menos 2 factores de autenticación siempre que sea posible. Esto agrega una capa adicional de seguridad al requerir una segunda forma de verificación, como un código generado en una aplicación móvil.

- Bloqueo de cuentas y políticas de intentos fallidos: Se requiere implementar medidas de bloqueo de cuentas después de varios intentos fallidos de autenticación. También establece políticas para manejar intentos de

autenticación y generar registros de estos intentos repetidos o sospechosos.

- Mecanismos de recuperación segura: De deben definir opciones de recuperación de cuenta, utilizando canales institucionales oficiales, asegurándose que sean seguras y verifiquen la identidad del usuario de manera adecuada. Evita preguntas de seguridad predecibles.

- Salting y hashing de contraseñas: En medida de lo posible se recomienda almacenar las contraseñas de los usuarios utilizando técnicas de hashing con salting. Esto protege las contraseñas en caso de que la base de datos sea comprometida.

- Protección contra ataques de fuerza bruta: Se debe implementar protecciones para prevenir ataques de fuerza bruta y diccionario, como retrasos en el intento de autenticación después de intentos fallidos.

- Sesiones seguras: Se recomienda utilizar cookies y tokens de sesión seguros para mantener la autenticación del usuario durante su sesión. Asegúrate de que las sesiones caduquen después de un período de inactividad.

- Gestión adecuada de tokens y cookies: Se debe asegurar que los tokens y cookies utilizados para la autenticación se generen y almacenen de manera segura, y que no sean vulnerables a ataques como secuestro de sesión.

- Autenticación basada en estándares: Se recomienda utilizar protocolos de autenticación ampliamente aceptados, como OAuth o OpenID Connect, en lugar de intentar desarrollar validaciones internas.

- Control de acceso basado en roles: Se debería implementar un sistema de control de acceso basado en roles para asegurarte de que los usuarios solo tengan acceso a las partes de la aplicación que son relevantes para su función.

- Registro y monitorización de autenticación: Debe implementarse un registro de auditoría para rastrear los intentos de autenticación y detectar actividades inusuales o sospechosas.

- Pruebas de penetración y autenticación: Deben Realizarse pruebas de penetración y evaluaciones de seguridad en la autenticación para identificar posibles vulnerabilidades y problemas.

5.8 A08:2021 Fallas en el Software y en la Integridad de los Datos

5.8.1 Prevenciones generales relacionadas con Software y en la Integridad de los Datos

- Bibliotecas y dependencias: Se debe asegurar que las bibliotecas y dependencias son utilizadas desde repositorios confiables. Si se desea tener un perfil de riesgo alto, entonces considere alojarlas en un repositorio interno cuyo contenido ha sido previamente analizado.

- Herramientas de análisis de componentes: Se recomienda utilizar herramientas de análisis de componentes de terceros, cómo OWASP Dependency Check u OWASP CycloneDX, con el fin de verificar la ausencia de vulnerabilidades conocidas.

- Proceso de revisión de cambios: deben existir procesos de revisión de cambios de código y configuraciones para minimizar las posibilidades de que código o configuraciones maliciosas sean introducidos en su pipeline.

- CI/CD: Se debe asegurar que el pipeline CI/CD utilizado posee adecuados controles de acceso, segregación y configuraciones que permitan asegurar la integridad del código a través del proceso de build y despliegue.

5.8.2 Prevenciones específicas relacionadas al Software y en la Integridad de los Datos

- Firmas digitales y hashes: Se recomienda utilizar firmas digitales y funciones de hash para verificar la integridad de los componentes de software y los datos, al verificar las firmas y comparar los valores hash, con esto se puede asegurar que los archivos no hayan sido modificados.

- Actualizaciones seguras: Se debe implementar un mecanismo seguro para la distribución y aplicación de actualizaciones de componentes internos de los equipos como por ejemplo los SDK, Runtime, Componentes .Net, Drivers ODBC, CLR Types, entre otros. Asegurándose que las actualizaciones provengan de fuentes confiables (Microsoft) y estén autenticadas.

- Control de acceso a sistemas y archivos: Se deben establecer políticas de control de acceso a nivel de sistema y archivo para garantizar que solo las personas autorizadas puedan modificar archivos y configuraciones críticas.

- Registro y auditoría de cambios: Se deberían implementar un sistema de registro y auditoría que guarde los cambios realizados en la estructura de datos, archivos y configuraciones críticas. Esto le permitirá rastrear quién realizó las modificaciones y cuándo.

- Implementar el monitoreo de integridad de archivos FIM (Monitoreo de Integridad de Archivos): Se recomienda implementar herramientas de monitoreo de integridad que alerten sobre cualquier cambio no autorizado en los archivos y componentes críticos.

- Restricción de acceso físico y lógico: Se debe limitar el acceso físico y lógico a los sistemas y servidores que almacenan datos críticos o componentes de software. Debe utilizar medidas de seguridad como autenticación y cifrado.

- Cifrado de datos: Se debe utilizar el cifrado para proteger los datos en reposo y en tránsito. Esto dificulta que los atacantes accedan a la información incluso si logran sortear otras medidas de seguridad.

- Segregación de funciones: Se deben dividir las responsabilidades y los permisos entre los miembros del equipo. Esto reduce la probabilidad de manipulación maliciosa de datos y componentes.

- Copias de seguridad y recuperación AVAMAR (software de copias de seguridad de datos basado en la nube): Se debería implementar un sistema de copias de seguridad regulares para que puedas restaurar datos y componentes en caso de una pérdida o corrupción.

5.9 A09:2021 Fallas en el Registro y Monitoreo

5.9.1 Prevenciones generales relacionadas con Fallas en el Registro y Monitoreo

- Errores de inicio de sesión: Se debe asegurar que todos los errores de inicio de sesión, como los de control de acceso y de validación de entradas de datos del lado del servidor se pueden registrar con suficiente contexto como para identificar cuentas sospechosas o maliciosas y mantenerlo durante el tiempo suficiente para permitir un posterior análisis forense.

- Registros: Se recomienda asegurar que los registros se generan en un formato fácil de procesar por las herramientas de gestión de registros, además de validar que los datos de registros estén correctamente codificados para prevenir inyecciones o ataques en el sistema de monitoreo o registros.

- Transacciones de alto valor: Se debe validar que las transacciones de alto valor poseen una traza de auditoría con controles de integridad para evitar

la modificación o el borrado, tales como permitir únicamente la inserción en las tablas de base de datos o similares.

- Equipos de DevSecOps: Los equipos de DevSecOps deben establecer alertas y monitoreo efectivo tal que se detecte actividades sospechosas y responder rápidamente, además se debería considerar el establecer o adoptar un plan de respuesta y recuperación de incidentes, tal como **NIST 800-61r2** o posterior.

5.9.2 Prevenciones específicas relacionadas a Fallas en el Registro y Monitoreo

- Definir requisitos de registro y monitorización: Se deben depurar los eventos y acciones de la aplicación que deben ser registrados y monitoreados (Auditorías). Esto puede incluir diversos eventos o cambios en configuraciones críticas y actividades inusuales.

- Implementación de registros de auditoría: Se deberían implementa registros de auditoría para registrar eventos importantes y actividades realizadas por usuarios y componentes del sistema. Registra información relevante como la fecha, hora, usuario y acción realizada.

- Formatos de registro seguros: Se recomienda utiliza formatos de registro seguros para evitar la inclusión de datos confidenciales o sensibles en los registros. No registres contraseñas u otra información personal.

- Almacenamiento seguro de registros: Se debe asegurar la correcta custodia y respaldo de los registros de manera segura en una ubicación protegida y aislada, para prevenir la manipulación maliciosa o el acceso no autorizado.

- Acceso controlado a registros: Se debe limitar el acceso a los registros únicamente al personal autorizado. Es decir, se deben implementar políticas

de control de acceso para asegurar que solo las personas necesarias puedan ver y analizar los registros.

- Monitorización en tiempo real: Se deben implementar sistemas de monitorización en tiempo real que alerten sobre eventos sospechosos o inusuales. Esto permitirá responder rápidamente a posibles amenazas.

- Integración con SIEM: Se recomienda integrar los registros y sistemas de monitorización con una solución de gestión de información y eventos de seguridad (SIEM) para centralizar y analizar los datos de seguridad.

- Definir umbrales y alertas: Se recomienda establecer umbrales y reglas de alerta para detectar comportamientos anómalos. Las alertas deben ser lo suficientemente sensibles para detectar amenazas, pero no generar demasiados falsos positivos.

- Pruebas de monitorización: Debería existir un plan periódico de pruebas de monitorización y respuesta a incidentes, con el fin de asegurar de que los sistemas estén funcionando según lo esperado y que el equipo esté preparado para responder a amenazas.

- Respaldo de registros: Se debería implementar una política de respaldo de registros para garantizar que los registros estén disponibles incluso en caso de fallos en los sistemas principales.

5.10 A10:2021 Falsificación de Solicitudes del Lado del Servidor (SSRF)

5.10.1 Prevenciones generales relacionadas a la Falsificación de Solicitud del Lado del Servidor (SSRF)

Los desarrolladores pueden prevenir SSRF implementando algunos o todos los siguientes controles de defensa en profundidad:

Desde la capa de aplicación:

- Se debe validar que todos los datos de entrada proporcionados por el cliente cumplan con el esquema de URL, el puerto y destino a través de una lista positiva de ítems permitidos.

- No se deberían enviar respuestas en formato "crudo" a los clientes, por lo tanto es recomendable deshabilitar las redirecciones HTTP.

- Se debe tener en cuenta la coherencia de la URL para evitar ataques como el enlace de DNS y las condiciones de carrera de "tiempo de verificación, tiempo de uso" (TOCTOU por sus siglas en inglés).

- No se debe mitigar el SSRF mediante el uso de una lista de denegación o una expresión regular. Por lo general los atacantes poseen listas de payloads, herramientas y habilidades para eludir las listas de denegación.

Medidas adicionales a considerar relacionadas a la Falsificación de Solicitud del Lado del Servidor (SSRF):

- Se debe considerar no implementar servicios relevantes para la seguridad en los sistemas frontales (por ejemplo, OpenID) además de controlar el tráfico local en estos sistemas (por ejemplo, localhost).

- Para frontends con grupos de usuarios dedicados y manejables, se debe considerar el uso del cifrado de red (por ejemplo, VPN) en sistemas independientes para considerar necesidades de protección muy altas.

5.10.2 Prevenciones específicas relacionadas a la Falsificación de Solicitud del Lado del Servidor (SSRF)

- Filtrar y validación de entradas: Es recomendable validar y filtrar todas las entradas del usuario, especialmente aquellas que se utilizan para construir URL o realizar solicitudes, por lo que se deberían utilizar listas blancas y evitar aceptar entradas no confiables.

- Utilización de rutas absolutas: En caso de ser posible, se recomienda utilizar rutas absolutas en lugar de rutas relativas en las solicitudes. Esto reduce la posibilidad de manipulación de rutas por parte de los atacantes.

- Validación de destino: Se recomienda verificar que las URL y las direcciones IP a las que se realizan solicitudes sean legítimas y autorizadas, para esto se deben utiliza mecanismos de control de acceso si es necesario.

- Configuración de red segura: Se debe configurar los sistemas y servidores que contienen los proyectos, para que solo permitan conexiones a destinos confiables y necesarios.

- Limitar recursos internos: Se debe limitar el acceso a los recursos internos solo a las partes de la aplicación que realmente los necesitan. No se deben permitir solicitudes indiscriminadas a servicios internos.

- Utilización de tokens CSRF: Se deberían implementar tokens CSRF (Cross-Site Request Forgery) para prevenir ataques que engañan al usuario en la realización de solicitudes no autorizadas.

- Supervisión de solicitudes salientes: Se recomienda monitorear las solicitudes salientes y registrar las solicitudes realizadas por el servidor hacia recursos internos. Esto puede ayudar a detectar actividades sospechosas.

Capítulo 6. Conclusión

En conclusión, la implementación de las recomendaciones brindadas en el presente trabajo de investigación, son cruciales para garantizar la integridad, confidencialidad y disponibilidad del aplicativo web definido como “CREDIWEB”, esto en el entorno digital actual en el que se ejecuta la herramienta.

Estas medidas bridadas, las cuales consideran desde el mantenimiento actualizado y la defensa proactiva, hasta la concienciación y preparación del personal que conforman estos departamentos, logran forman un escudo integral contra posibles ataques o incidentes que se puedan presentar en un futuro al aplicativo. Por lo tanto, al seguir estas recomendaciones, se establece no solo una base sólida para la protección contra ataques maliciosos, sino una respuesta efectiva a incidentes de seguridad y de esta forma se aplica de manera implícita el modelo de los tres pilares de la ciberseguridad, el cual está compuesto por personas, tecnología y procesos.

En última instancia, la inversión en ciberseguridad no solo genera confianza de los usuarios y directivos, sino que también contribuye a la sostenibilidad y éxito a largo plazo de la plataforma web y así como a la escalabilidad y robustes del mismo.

Referencias Bibliográficas

Akamai. (2023). ¿Qué es un ataque DDoS? https://www.akamai.com/es/glossary/what-is-ddos.

Ambit. (2023). Diferencias entre amenaza, vulnerabilidad y riesgo. https://www.ambit-bst.com/blog/diferencias-entre-amenaza-vulnerabilidad-y-riesgo.

Buchanan. (2023). Gestión de configuración. https://www.atlassian.com/es/microservices/microservices-architecture/configuration-management.

Collins: (2023). Source code. https://www.collinsdictionary.com/es/diccionario/ingles/source-code).

Cyber zaintza. (2023). Los tres pilares en la ciberseguridad: las personas, los procesos y la tecnología. https://www.ciberseguridad.eus/ciberpedia/buenas-practicas/los-tres-pilares-en-la-ciberseguridad-las-personas-los-procesos-y-la.

Distillery. (2023). ¿Qué son las pruebas de seguridad de las aplicaciones y cómo funcionan? https://distillery.com/es/blog/que-son-las-pruebas-de-seguridad-de-las-aplicaciones-y-como-funcionan/.

Enclave de Ciencia. (2023). Enclave de Ciencia. https://enclavedeciencia.rae.es/ciberdefensa.

Enclave de ciencia. (2023). Framework. https://enclavedeciencia.rae.es/Framework.

Escuela Europea de la excelencia. (2023). ISO 31000. Términos y definiciones. https://www.escuelaeuropeaexcelencia.com/2015/11/iso-31000-terminos-definiciones/.

FTC en Español. (2023). Comprender el marco de ciberseguridad del NIST. https://www.ftc.gov.

IBM. (2023). Cuentas de usuarios. https://www.ibm.com/docs/es/b2b-integrator/6.0.1?topic=security-user-accounts.

IBM. (2023). CVSS (Common Vulnerability Scoring System). https://www.ibm.com/docs/es/qradar-on-cloud?topic=vulnerabilities-common-vulnerability-scoring-system-cvss.

IBM. (2023). ¿Qué es el desarrollo de software? https://www.ibm.com/es-es/topics/software-development.

kaspersky. (2023). ¿Qué es la inyección de SQL? Definición y explicación. https://latam.kaspersky.com/resource-center/definitions/sql-injection.

Microsoft. (2023). ¿Qué es el control de acceso? https://www.microsoft.com/es-ww/security/business/security-101/what-is-access-control.

Ostec. (2023). Los pilares de la Seguridad de la Información, según la norma ISO 27001. https://ostec.blog/es/aprendizaje-descubrimiento/los-pilares-de-la-seguridad-de-la-informacion-segun-la-norma-iso-27001/.

Owasp. (2023). Format string attack. https://owasp.org/www-community/attacks/Format_string_attack.

Real Academia Española. (2023). Diccionarios. https://dle.rae.es/.

Wikipedia. (2023). Common Vulnerabilities and Exposures. https://es.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures.

Wikipedia. (2023). Código de error.https://es.wikipedia.org/wiki/C%C3%B3digo_de_error.

wikipedia. (2023). Common Weakness Enumeration. https://es.wikipedia.org/wiki/Common_Weakness_Enumeration.

wikipedia. (2023). Inyección de código. https://es.wikipedia.org/wiki/Inyecci%C3%B3n_de_c%C3%B3digo.

Zaptest. (2023). Pruebas de Caja Blanca: ¡Qué es, Cómo funciona, Retos, Métricas, Herramientas y Más! https://www.zaptest.com/es/pruebas-de-caja-blanca-que-es-como-funciona-retos-metricas-herramientas-y-mas.

Zaptest. (2023). Pruebas de caja gris – Profundice en qué son, tipos, procesos, enfoques, herramientas y mucho más. https://www.zaptest.com/es/pruebas-de-caja-gris-profundice-en-que-son-tipos-procesos-enfoques-herramientas-y-mucho-mas.

Zaptest. (2023). Pruebas de caja negra: qué son, tipos, procesos, enfoques, herramientas y mucho más. https://www.zaptest.com/es/pruebas-de-caja-negra-que-son-tipos-procesos-enfoques-herramientas-y-mucho-mas.

Insitech. (2022). Gestión y administración de la disponibilidad. https://go.insitech.com.mx/gestion-y-administracion-de-la-disponibilidad/.

KeepCoding. (2022). ¿Qué es el hacking web? https://keepcoding.io/blog/que-es-el-hacking-web/.

Orange. (2022). Cómo convertirse en un cazador de 'bugs' profesional. https://blog.orange.es/innovacion/bug-hunting-aprenderlo-profesion/

Pathak. (2022). SAST vs DAST: ¿Qué es mejor para las pruebas de seguridad de aplicaciones? https://geekflare.com/es/sast-vs-dast-application-security-testing/.

Báez. (2021). Qué es un ataque de XSS o Cross-Site Scripting. https://www.welivesecurity.com/la-es/2021/09/28/que-es-ataque-xss-cross-site-scripting/.

Wikipedia. (2021). Validación de datos. https://es.wikipedia.org/wiki/Validaci%C3%B3n_de_datos.

Gran Diccionario de la Lengua Española. (2016). Gran Diccionario de la Lengua Española. https://es.thefreedictionary.com/intrusi%c3%b3n.

Pérez y Gardey. (2015). API - Qué es, definición y concepto. https://definicion.de/api/.

Pérez. (2014). Qué son y cómo funcionan los Buffer Overflow. https://www.welivesecurity.com/la-es/2014/11/05/como-funcionan-buffer-overflow/.

Pérez y Gardey. (2014). Requerimiento - Qué es, definición, en la informática y en el derecho. Definiciones. Última actualización el 11 de octubre de 2021. https://definicion.de/requerimiento/.

ISO27000.es. (2005). https://www.iso27000.es/glosario.html.

Oracle. (2004). Protocolo de transferencia de hipertexto. https://www.oracle.com/it-infrastructure/.

Ríos, (1995) y M. Felicísimo. (1997). Curso sobre Modelos Digitales del Terreno. https://www6.uniovi.es/~feli/CursoMDT/Tema1/Page1.html.