ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"IMPLEMENTACIÓN DE UN ESQUEMA DE SEGURIDAD INICIAL PARA LAS APLICACIONES WEB DEL GRUPO COMERCIAL IIASA ECUADOR, USANDO COMO REFERENCIA LOS RIESGOS DE SEGURIDAD DE APLICACIONES WEB DEL APARTADO OWASP TOP 10 2013"

TRABAJO DE TITULACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

MAGISTER EN SEGURIDAD INFORMÁTICA APLICADA

ALEX MANUEL LOAIZA CARPIO

GUAYAQUIL – ECUADOR

2017

AGRADECIMIENTOS.

A Dios.

Por haberme dado las fuerzas necesarias para emprender y culminar esta etapa de mi vida y por brindarme día a día nuevos aprendizajes, experiencias y sobre todo felicidad.

A mis padres Aura y Manuel.

Que gracias a su valioso esfuerzo y ejemplo han podido guiarme y educarme correctamente, fomentando en mi valores y principios imborrables.

A mi compañera de vida Olga.

Por guiarme y acompañarme en este arduo camino y compartir conmigo alegrías y tristezas.

Alexandra y Doménica.

Por compartir conmigo la alegría de ser padre, presenciar el esfuerzo dedicado al presente documento y por los momentos de familia que podemos vivir y disfrutar juntos.

Miguel y Dayanna.
Mis hermanos, por haberme brindado el apoyo y las fuerzas para culminar con éxito el presente documento.

Ing. Johnny Vera Solórzano.
Por los consejos y la colaboración brindada durante la elaboración de este proyecto en las instalaciones de la compañía.

Ing. Robert Andrade.
Por la orientación, ayuda y consejos que me brindó para la realización de la tesis.

Ángel, Héctor, Lourdes y Joseph
Amigos que estuvieron desde el momento en que inicie la maestría hasta la culminación y que con su apoyo no hubiese logrado esta meta propuesta.

Gracias a todas las personas que ayudaron directa o indirectamente en la realización de este proyecto.

Alex

DEDICATORIA.

Gracias a mi universidad, por haberme permitido formarme profesionalmente, a todas las personas que fueron participes de este proceso: docentes, compañeros de clases y amigos los mismo que con sus enseñanzas y experiencias ayudaron de manera directa o indirecta la culminación del presente trabajo de Tesis.

Gracias a mi familia que juntos fueron el motor principal durante este proceso, gracias a Dios, que fue mi principal apoyo y motivador para continuar sin desmayar en el camino.

Una dedicatoria especial a una nena que aunque no esté conmigo esta siempre en mis pensamientos, este logro es para ti Nathaly Sophia que desde el cielo cuidas nuestros pasos. Te amo.

TRIBUNAL DE SUSTENTACIÓN.

DIRECTOR DE MSIG/MSIA

ING. LENÍN FREIRE

DIRECTOR DE PROYECTO DE GRADUACIÓN

ING. ROBERT ANDRADE

MIEMBRO DEL TRIBUNAL

ING. RONNY SANTANA

DECLARACIÓN EXPRESA.

"La responsabilidad y la autoría del contenido de esta Tesis de Grado, me corresponde exclusivamente; y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

.....

Ing. Alex Loaiza Carpio

RESUMEN.

Hoy en día las aplicaciones web, en una organización, son consideradas de gran valía porque permiten a sus integrantes acceder a la información de las organizaciones mediante el uso de navegadores web desde redes internas o desde el internet. IIASA es un grupo comercial que desarrolla sus aplicaciones web de manera interna y en ciertas ocasiones proveedores de software realizan proyectos de gran envergadura. Todo el desarrollo web realizado en la institución emplea la plataforma Java Enterprise Edition (JEE), las mismas que se han desarrollado sin tener un plan de seguridad que permita reforzar las seguridades de las aplicaciones web y de la información que allí se muestra.

El presente proyecto tiene como finalidad implementar un esquema de seguridad inicial orientado a los riesgos de seguridad propuestos por el apartado OWASP Top 10 2013; que permita la detección de vulnerabilidades, el análisis de riesgos según la metodología de OWASP y la implementación de mejoras de seguridad en las aplicaciones web con nivel de severidad medio o alto.

A partir de los resultados obtenidos donde se identificaron riesgos de Inyección, Secuencia de Comandos en Sitios Cruzados, Referencia Directa Insegura de Objetos y Configuraciones de Seguridad Incorrectas se realizó la implementación de controles de seguridad por cada vulnerabilidad encontrada usando la herramienta ESAPI Java, con la finalidad de asegurar las aplicaciones y mitigar los riesgos encontrados en las aplicaciones web consideradas en el presente estudio, las que no fueron consideradas y demás desarrollo interno y externo propuesto a futuro.

ÍNDICE GENERAL.

ABREVIATURAS Y SIMBOLOGÍA.

ACL	Lista de control de accesos.
AS400	Equipo IBM de gama media y alta.
ASP	Active Server Pages.
CAT	Fabricante de equipos de construcción.
CDT	Centro de Desarrollo Técnico.
COBIT	Control Objectives for Information and related Technology.
CRM	Customer Relationship Management.
CRSF	Cross-site Request forgery.
CSS	Cascading Style Sheets.
CWE	Common Weakness Enumeration.
DB2	Base de datos relacional introducido por IBM.
DBS	Dealer Business System.
DDoS	Ataque de denegación de servicio distribuido.
DES	Data Encryption Standard.

ERP	Enterprise Resource Planning.
ESAPI	Enterprise Security API.
ETC	Etcétera.
FTP	Protocolo de transferencia de archivos.
HTML	Hyper Text Markup Language.
HTTP	Hyper Text Transfer Protocol.
HTTPS	Hyper Text Transfer Protocol Secure.
IDS	Sistema de Detección de Intrusos.
IIASA	Importadora Industrial Agrícola S.A.
IP	Protocolo de Internet.
ISO	Organización Internacional de Normalización.
ITIL	Biblioteca de Infraestructura de Tecnologías de Información.
JSF	Java Server Faces.
LDAP	Lightweight Directory Access Protocol.
MIME	Multi-Purpose Internet Mail Extensions.
MVC	Modelo Vista Controlador.
OS	Sistema Operativo.

OWASP	Open Web Application Security Project.
PHP	Lenguaje de programación para desarrollo web.
RSA	Rivest, Shamir y Adleman. Sistema criptográfico de clave pública.
SQL	Lenguaje de consulta estructurado.
SSL	Secure Sockets Layer.
SVN	Apache Subversion. Sistema de revisiones de versiones de software.
TI	Tecnología de la información.
URL	Localizador uniforme de recursos.
WASC	Web Application Security Consortium.
XML	Sistema para definir, validar y compartir formatos de documentos en la web.
XSS	Secuencia de comandos en sitios cruzados.

A1	Inyección.
A2	Pérdida de autenticación y gestión de sesiones.
A3	Secuencia de comandos en sitios cruzados.
A4	Referencia directa insegura a objetos.
A5	Configuración de seguridad incorrecta.
A6	Exposición de datos sensibles.
A7	Ausencia de control de acceso a funciones.
A8	Falsificación de peticiones en sitios cruzados.
A9	Utilización de componentes con vulnerabilidades conocidas.
A10	Redirecciones y reenvíos no validados.
$	Moneda Americana (dólar)
%	Porcentaje.

ÍNDICE FIGURAS.

ÍNDICE TABLAS.

INTRODUCCIÓN.

Desde instituciones del estado, multinacionales, empresas y hasta los pequeños negocios en el mundo hacen uso de la tecnología para brindar, a sus diferentes clientes, un valor agregado del producto o servicio que ofrecen, llegando así a diferentes mercados con el uso del internet y de las aplicaciones web.

El uso de tecnologías de información es considerado actualmente como una estrategia de éxito en los negocios de una organización, transformando de manera positiva su infraestructura y su entorno social, volviéndolas más competitivas y mejorando así sus operaciones internas y externas.

Dentro de estas tecnologías podemos destacar el uso de aplicaciones web para el despliegue y manejo de información invaluable de una organización (números de tarjeta de créditos, cartera de clientes, inventarios físicos, presupuestos, sueldos, etc.) estas aplicaciones en su mayoría cumplen las necesidades funcionales para las cuales fueron creadas, pero se deja de lado

el tema de la seguridad dando lugar a que dichas aplicaciones sean blanco fácil de ataques mal intencionados o no.

La mayor parte del presupuesto de TI en las organizaciones está destinado a la seguridad en la infraestructura de la red (seguridad perimetral, firewalls, sistema de detección de intrusos (IDS), sistemas de prevención de intrusos (IPS), etc) sin embargo, las aplicaciones (web, escritorio, móviles, etc) no son tratadas de igual manera y se deja la seguridad en manos del programador de turno sin tener, en muchos de los casos, noción de las diferentes vulnerabilidades que se pueden presentar. OWASP es una organización sin fines de lucro que emite una clasificación de vulnerabilidades, Top 10 2013, en donde se detalla las diferentes vulnerabilidades a las que están expuestas las aplicaciones web de la mayoría de organizaciones a nivel mundial proporcionando una guía para enfrentar y prevenir las deficiencias de seguridad a las cuales se encuentran expuestas las aplicaciones.

Muchos de los ataques informáticos que suceden actualmente se dan a nivel de aplicación, por ello es importante disponer de un esquema de seguridad

que facilite la detección de vulnerabilidades en las aplicaciones web, luego analizar dichas vulnerabilidades emitiendo un criterio de severidad del riesgo que se produciría en caso de que la vulnerabilidad sea explotada y por último implementar las diferentes recomendaciones o soluciones para tratar los diferentes riesgos presentados.

CAPÍTULO 1.

1. GENERALIDADES

1.1 Antecedentes.

Las nuevas tendencias tecnológicas hacen que muchas empresas adopten el uso de software operativo y de información para sus tareas cotidianas, incluso la misma competencia en el mercado hace que las compañías adquieran software o realicen el propio de manera independiente.

El concepto de seguridad en aplicaciones en muy escaso debido a que muchos de los programadores, analistas, líderes de proyecto, gerentes, etc. no lo ven como prioridad, ya que su meta es realizar el software en un determinado tiempo y con el menor costo posible.

En la actualidad los ataques informáticos hacia sitios web, se dan en la capa de aplicación y no en la red o al sistema, esto además de provocar cierto daño

a los sistemas comprometidos también deja secuelas en la imagen de la empresa afectando su negocio.

Por eso es esencial, en toda compañía, tener un control de las aplicaciones web que están bajo su supervisión, sistemas operativos, herramientas de software, plugins, Frameworks, software de terceros, etc., para disminuir las afectaciones a los servicios web que la compañía ofrece a sus clientes.

1.2 Descripción del problema.

La compañía IIASA Ecuador es un grupo comercial importador establecido en el país por más de 90 años, su primera oficina se estableció en la ciudad de Guayaquil-Ecuador, gracias a la visión del Sr. Benjamín Rosales Pareja, es una de las primeras cadenas distribuidora de la marca Caterpillar en Latinoamérica y Canadá.

Actualmente la compañía tiene 2 sucursales, Guayaquil y Quito, y múltiples oficinas a nivel nacional, contribuyendo así al progreso del país.

El departamento de sistemas de la empresa IIASA está divido en: área técnica y área de desarrollo. El área de desarrollo cuenta con varios analistas que se encargan del soporte de las aplicaciones heredadas, nuevas o externas.

Las operaciones de cómputo se encuentran, de forma centralizada, en la oficina matriz ubicada en la ciudad de Guayaquil; entre sus aplicativos importantes tenemos el ERP que maneja todas las operaciones transaccionales de la empresa (ventas, inventario, órdenes de trabajo,

nómina, etc), CRM que maneja todas las oportunidades de venta con clientes vigentes o potenciales y la intranet en donde se realizan todas las operaciones administrativas tales como: aprobaciones, generación de diarios contables, revisión de estados de cuenta, generación de presupuestos, roles de pago, recibos de cobro, etc.

Existe preocupación por parte de la Gerencia de Sistemas con respecto a las aplicaciones web desplegadas en la intranet de la institución, como también de aquellas aplicaciones en mantenimiento o nuevas, ya que desde que se formó el departamento no se ha llevado a cabo ningún análisis de vulnerabilidades de las aplicaciones web y el desarrollo de aplicaciones no ha tenido un lineamiento a la seguridad, por lo que se nos ha solicitado se tenga en cuenta, para esta evaluación, los siguientes puntos:

- Escaneo de vulnerabilidades de las aplicaciones web de intranet críticas del negocio.
- Documentar los resultados del análisis de vulnerabilidades encontradas en las aplicaciones web estudiadas.
- Sugerir e implantar soluciones que mitiguen los riesgos encontrados.
- Instruir a los desarrolladores del departamento buenas prácticas de programación.

1.3 Objetivo General.

Implementar un esquema de seguridad inicial para las aplicaciones web del grupo comercial IIASA Ecuador, mediante el análisis de los riesgos de seguridad de las aplicaciones web y la puesta en práctica de las

recomendaciones de seguridad según “OWASP Top 10-2013”, para corregir los riesgos detectados y asegurar las aplicaciones nuevas o existentes.

1.4 Solución propuesta.

Como solución se propone implementar un esquema inicial de seguridad que abarca lo siguiente: el escaneo de vulnerabilidades de las aplicaciones web desplegadas, el análisis de las vulnerabilidades encontradas e implementar medidas correctivas y preventivas para mitigar las falencias encontradas.

El esquema de seguridad inicial tendrá como marco de referencia el apartado de seguridad OWASP Top 10 2013, en donde se describe los riesgos más serios que afectan a las aplicaciones web y las medidas a tomar para mitigar dichos riesgos.

OWASP es una fundación sin fines de lucro cuyo propósito es contribuir con material de seguridad de información a las organizaciones y profesionales, de manera tal las mismas se encuentren informadas y capacitadas para aplicar controles de seguridad sobre sus aplicativos web.

1.5 Objetivos Específicos.

- Identificar las aplicaciones web que integran la intranet y que serán objeto de estudio para el análisis.

- Analizar y diseñar un plan de detección de vulnerabilidades de las aplicaciones web de la compañía, analizando uno a uno los sistemas que integran la intranet según su importancia en el negocio, clasificar sus vulnerabilidades encontradas de acuerdo a las recomendaciones dadas

por “OWASP Top 10-2013” e indicar sus índices de riesgos e impacto en el negocio.

- Desarrollar un esquema para clasificar y evaluar los riesgos encontrados, definir su impacto en el negocio e implementar controles de seguridad estándar ESAPI, protocolo de comunicación seguro (https), manejo de sesiones (cookies, tokens, etc.) y otras recomendaciones de seguridad según “OWASP Top 10-2013”

- Implementar el esquema de seguridad inicial propuesto a las aplicaciones web de la intranet de la compañía.

1.6 Metodología.

OWASP es una comunidad abierta, dedicada a la información sobre las seguridades en aplicaciones web proporcionando directrices para un software seguro y además determina mediante estándares, estudios, buenas prácticas, etc., las causas por las que hacen a un software inseguro.

El marco de referencia proporcionado por OWASP es usado incluso por estándares internacionales de seguridad informática, tales como: COBIT, ITIL e ISO-27001. OWASP Top 10 2013, además de identificar los riesgos, enseña como remediar dichos riesgos con ejemplos y buenas prácticas.

El esquema de seguridad propuesto consiste en:

- Detectar las vulnerabilidades en las aplicaciones web mediante el uso de herramientas Open Source como: Owasp-Zap y Web-Scarab etc. Cabe

indicar que se escanearán 50 aplicaciones web previamente elegidas por el departamento de sistemas de la compañía.

- Para el análisis de las vulnerabilidades usaremos el apartado de seguridad de aplicaciones web “OWASP Top 10-2013” que nos detalla, mediante un esquema, los riesgos comunes a todo tipo de organización: identificando la amenaza y categorizando el riesgo, proporcionando así el criterio para determinar que vulnerabilidad tiene un mayor impacto en el negocio.

- Finalizando con las medidas de corrección, prevención y mejoras a implantarse en la compañía, para lo cual dispondremos una matriz de riesgos con su respectiva remediación, uso del protocolo de comunicación segura (https), manejo adecuado de sesiones de usuario mediante el uso de tokens, uso de la herramienta Open Source ESAPI para la detección de ingreso de código malicioso y establecer procedimientos y buenas prácticas de desarrollo seguro.

CAPÍTULO 2.

2. MARCO TEÓRICO.

2.1. Principios de seguridad informática.

2.1.1. Seguridad informática.

Es aquella disciplina encargada de proteger los activos de información mediante la implementación de procedimientos, métodos y técnicas a fin de evitar que posibles amenazas exploten las vulnerabilidades encontradas, volviendo al sistema seguro y confiable; minimizando el riesgo.

Eventualmente todo sistema de seguridad fallará es decir, se definen los controles y procedimientos para mantener un nivel aceptable del riesgo percibido y no la eliminación del mismo.

2.1.2. Principios.

El activo más importante de toda organización moderna es la información por tanto, la seguridad informática tiene como objetivo cumplir con los siguientes principios:

- Integridad de la Información: Este principio se encarga de garantizar que los datos no hayan sido alterados en su contenido es decir, que la información recibida o recuperada sea exactamente la misma que fue enviada o almacenada, sin que se haya producido una alteración indebida.

  Si la información enviada o recibida sufre alteraciones, entonces pierde la calidad de íntegra y no será confiable.

- Confidencialidad: Este principio tiene como finalidad asegurar que sólo la persona correcta tenga la autorización para acceder a la información concedida es decir, la información generada se destina a un cierto grupo o miembro que son capaces de acceder a ella debido a que cuentan con la autorización para hacerlo.

  La pérdida de confidencialidad es llamada también como la pérdida del secreto.

- Disponibilidad: Este principio tiene como finalidad garantizar que la información enviada de manera íntegra y confidencial llegue en el momento oportuno es decir, que se utilice cuando sea necesario,

que esté al alcance de los usuarios autorizados y que se pueda acceder a ella en cualquier momento.

La pérdida de disponibilidad se da cuando el ambiente tecnológico y humano no es el adecuado, provocando caídas y fallos, repercutiendo así en la continuidad de las operaciones cotidianas.

- No repudio: Este principio garantiza que el uso y/o modificación de la información enviada o recibida sea negada, en un futuro, por las partes involucradas es decir, que ninguna de las partes niegue el uso de la información; este principio es usado en las transacciones comerciales de hoy en día en internet, ya que incrementa la confianza entre las partes comunicadas.

2.2. Seguridad de aplicaciones web.

2.2.1. Aplicación web.

Se define como aplicación web a toda herramienta que permite a los usuarios acceder a un servidor web a través de la red (internet o intranet), mediante el uso de navegadores web como: Internet Explorer, Firefox, Chrome, Safari, etc.

El beneficio de una aplicación web radica en lo práctico y ligero que resulta el uso del navegador web, dando como resultado una fácil interacción entre el usuario y los datos que se transmiten.

Generalmente estas aplicaciones tiene una estructura de tres capas, la primera que está constituida por el propio navegador que se encuentra

en el cliente es decir, en cada uno de nuestros ordenadores; la segunda comprende el uso de un motor capaz de usar tecnología web dinámica es decir, interpretación de un lenguaje de programación PHP, Java Servlets, ASP, etc.; y por último la capa en donde se almacena la información transferida o requerida es decir, una base de datos, ficheros, etc.

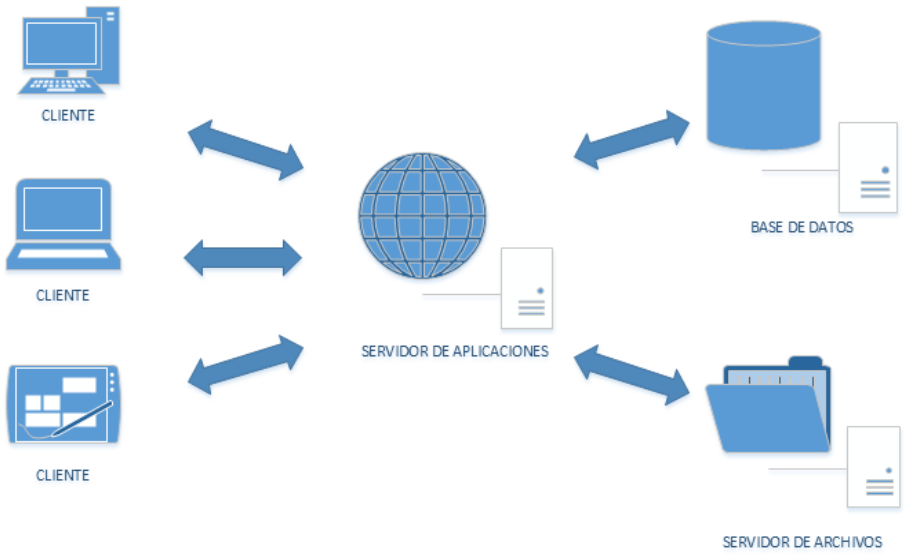


Figura 2.1: Arquitectura general de una aplicación web.

El uso de estas aplicaciones se ha extendido por muchos años y en la actualidad sigue creciendo, incluso se adaptan al dispositivo que realiza la petición o consulta.

2.2.2. Seguridad.

Podemos decir que la seguridad de aplicaciones web es una rama de la seguridad informática encargada primordialmente de la seguridad de sitios, aplicaciones y servicios web.

Las aplicaciones web son el blanco principal de ataques informáticos debido a: su exposición (internet o intranet) por no contar, en su mayoría, con un esquema adecuado de seguridad y porque en ellas se encuentra información lucrativa para los atacantes.

La seguridad, en aplicaciones web, supone un coste económico y de eficiencia, para ello debemos tener en cuenta 3 referencias: El riesgo cero no es práctico, existen diversas formas de mitigar el riesgo y no se debe invertir en seguridad si lo que se piensa proteger no es relevante para el individuo u organización.

Los problemas de seguridad de sitios web, portales, foros, redes sociales, tiendas virtuales, etc., se encuentran a nivel de aplicación y son en su mayoría el resultado de una codificación deficiente por parte del programador o del equipo de trabajo. La tarea de desarrollar código seguro depende en gran medida de los procedimientos y lineamientos establecidos para el desarrollo de estas aplicaciones, esto no solo debe recaer en el programador sino más bien debe ser tarea de todas las personas que intervienen en la creación, en el diseño, en la implementación, en las pruebas y la puesta en producción del software desarrollado.

2.3. Detección de vulnerabilidades en las aplicaciones web.

2.3.1. Amenazas.

Se define como amenaza informática a toda circunstancia, evento o persona que puede, potencialmente, causar daño a un sistema en forma de: robo, destrucción, divulgación, modificación de datos o negación de servicios.

Entre las amenazas comunes relacionadas a las aplicaciones informáticas, tenemos:

- Ingreso y flujo de datos: Se refiere a los datos ingresados por parte del usuario que pueden provocar comportamientos inesperados en la aplicación, manejando inadecuadamente los datos ingresados que pueden provocar cierto daño a la aplicación o su entorno.

- Relaciones de confianza: Se refiere a los componentes de un software y como este interactúa con los demás componentes propios o de terceros. Generalmente se tiene la idea de que un software o componente de terceros son elementos seguros, estos supuestos provocan fallas de vulnerabilidades serias.

- Supuestos y confianza mal colocada: Suponer algo que no es cierto es muy común entre los programadores, llevándolos a colocar confianza en donde no debe, esto se da porque se confía mucho en el origen de los datos y su contenido. Entre los supuestos incorrectos más comunes tenemos: Validación y formato de los

datos ingresados, capacidad de los atacantes y usuarios, seguridad de los programas de soporte, hostilidad potencial del ambiente de ejecución, etc.

- Ataques al ambiente: Esta amenaza está relacionada al ambiente en donde reside la aplicación y se ejecuta (sistema operativo, hardware, redes, bases de datos, usuarios). Asegurar que el ambiente en donde se ejecuta la aplicación no esté exenta de amenazas, es considerada como un supuesto que puede llevarnos a serios problemas de seguridad.

- Condiciones excepcionales: Esta condición está relacionada a los datos y al ambiente que pueden provocar una interrupción del programa a través de medidas externas. Pueden ocasionar una condición de falla en el programa con el consumo exagerado de recursos globales.

2.3.2. Vulnerabilidades.

Una vulnerabilidad es una debilidad o error (intencional o no) en un sistema informático que puede provocar daño a un activo o recurso informático.

La deficiencia en el diseño, implementación, operación o controles internos, pueden provocar violaciones a la seguridad de un sistema o aplicación. Para que exista una vulnerabilidad es necesario que se

presente una amenaza ya que por sí sola no podría causar daño alguno.

Las vulnerabilidades relacionadas a las aplicaciones web las podemos clasificar de la siguiente manera:

- De diseño: Como su nombre lo indica está relacionado a una de las primeras etapas del ciclo de vida del software, refiriéndose también a la etapa del levantamiento de requerimientos funcionales. Asumir que el ambiente en donde se ejecuta la aplicación está libre de cualquier ataque o es seguro, es una aseveración errónea. Es un error también enfocarse demasiado en los requerimientos funcionales y no en aquellos que dependan de la seguridad de la aplicación.

- De implementación: Está relacionada con la etapa de implementación en el ciclo de vida del software, en donde la aplicación desarrollada generalmente funciona según los requerimientos levantados en la primera etapa, pero puede llegar a ocurrir cierto problema de seguridad en la forma en que se lleva a cabo las operaciones que ejecuta la aplicación.

- Operacionales: Dichas vulnerabilidades surgen de procedimientos operativos y uso general del software en un entorno específico, no está relacionado al código propio de la aplicación sino a la interacción con el entorno.

2.3.3. Detección.

La detección de vulnerabilidades consiste en encontrar posibles fallas de seguridad que atenten con la confianza de un sitio o aplicación web. Estas fallas pueden ser explotadas por los ciber-delincuentes para exponer o hurtar la información que allí se despliegue (información personal, corporativa, etc).

El aprovechamiento de una vulnerabilidad es conocido como “ataque”, y este se clasifica en Activo y Pasivo. Los ataques activos fuerzan las entradas provocando que la aplicación deje de funcionar, mientras que el ataque pasivo es aún más peligroso, porque se introducen sin ser detectados y estos buscan privilegios de administrador en el sistema operativo para instalar algún tipo de programa dañino o para sustraer información sensible.

Existen técnicas para la detección de vulnerabilidades entra las cuales podemos nombrar las siguientes:

- Black-box (caja negra) es una técnica que se basa en descubrir vulnerabilidades desde el lado del atacante.

- White-box (caja blanca) es una técnica, en la cual se tiene información relevante de la organización, todo es planificado.

- Análisis estático de código, esta técnica no requiere de la ejecución de la aplicación, sino que se encarga de realizar un análisis al código fuente del programa.

- Análisis de código dinámico, esta técnica se comunica con la aplicación a través de su front-end para identificar posibles vulnerabilidades y debilidades en la arquitectura del sistema informático.

- Pruebas de penetración, esta técnica simula un ataque externo e interno para detectar las vulnerabilidades que podrían resultar de configuraciones deficientes, fallos de hardware o de software, fallos operativos, fallos en procedimientos establecidos, etc.

- Pruebas pasivas, diseñado para el análisis de tráfico en las telecomunicaciones. Permite detectar las fallas en las tramas enviadas por el canal de comunicación.

- Pruebas activas, esta técnica usa un programador de subproceso asignados al azar cuyo objetivo es verificar que las advertencias enviadas de un análisis predictivo son errores reales.

2.4. Riesgos de seguridad en aplicaciones web.

2.4.1. Riesgo en seguridad informática.

La norma ISO 27001 define como Riesgo: “A la posibilidad en que una amenaza concreta pueda explotar una vulnerabilidad, para causar una pérdida o daño en un activo de información”.

En general, el riesgo se mide como la multiplicación entre el impacto y la probabilidad de ocurrencia. El impacto se refiere a las consecuencias una vez concebido el riesgo, mientras que la probabilidad indica si el riesgo se da o no y con qué frecuencia.

Las aplicaciones web, en una organización, pueden ser atacadas por diferentes rutas; estas rutas deben ser identificadas de tal manera que la vulnerabilidad, iniciada por una amenaza, sea controlada o mitigada disminuyendo el impacto en el caso de que el riesgo se materialice.

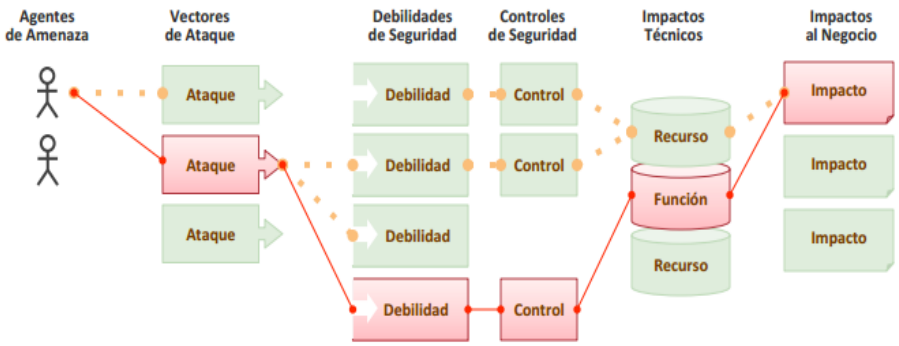


Figura 2.2: Esquema de amenazas, controles e impacto técnico y negocio según OWASP Top 10 2013 [1].

2.4.2. Riesgos de seguridad según OWASP Top 10 2013.

El Top 10 de la fundación OWASP es uno de los proyectos más conocidos internacionalmente ya que es referenciado por numerosos organismos, estándares y libros en la definición de requerimientos mínimos de seguridad exigidos en el desarrollo de aplicaciones web.

Existen varios apartados publicados desde el 2003, dando a conocer a la sociedad los diferentes riesgos de seguridad en los cuales muchas de las organizaciones se han visto expuestos. El último apartado fue publicado en el 2013 en donde se definen los diez riesgos de seguridad más críticos y extendidos en el ámbito de las aplicaciones web.

Entre los riesgos de seguridad publicados en el OWASP Top 10 2013 tenemos:

- **A1 Inyección**: Está relacionado con la validación de datos confiando así en la información enviada a un intérprete, ejecutando consultas o comandos con los datos no confiables. La explotación de esta vulnerabilidad permite el acceso, pérdida o alteración de la información almacenada.

- **A2 Pérdida de autenticación y gestión de sesiones**: Está relacionado con las deficiencias en el proceso de autenticación o en la gestión de sesiones comprometiendo contraseñas, claves, tokens de sesiones, etc., posibilitando la suplantación de usuarios

pudiendo realizar acciones indebidas con la identidad de la víctima suplantada.

- **A3 Secuencia de comandos en sitios cruzados (XSS)**: Está relacionado con la validación deficiente de los datos proporcionados por los usuarios. Permite al atacante inyectar secuencias maliciosas en el navegador de la víctima pudiendo obtener información como: la sesión de usuario, contenido no autorizado, captura de teclas pulsadas por la víctima o dirigirlos a sitios maliciosos, entre otros.

- **A4 Referencia directa insegura a objetos**: Relacionada a la exposición de un objeto en el sistema, que permite el acceso indebido a ficheros, directorios o bases de datos. El usuario es capaz de acceder a información para la cual no está autorizado, su impacto es la exposición de datos privados a usuarios no autorizados.

- **A5 Configuración de seguridad incorrecta**: Está relacionada a la mala configuración de seguridad en los equipos o servicios que conforman el entorno de una aplicación web (servidor de aplicación, correo, bases de datos, etc.). Las configuraciones de seguridad en su mayoría se establecen por defecto, por lo que deben gestionarse de manera segura, con el fin de evitar futuras intromisiones no autorizadas.

- **A6 Exposición de datos sensibles**: Se refiere a la protección incorrecta de datos críticos tales como números de tarjetas de crédito o credenciales de autenticación. Los datos sensibles deben tener métodos adicionales de protección como el cifrado de datos y protecciones especiales en el intercambio de información. La explotación de esta vulnerabilidad permite al atacante expropiarse de dicha información para luego cometer fraudes u otros delitos.

- **A7 Ausencia de control de acceso a funciones**: Escasez de controles en el servidor (aplicaciones, base de datos, etc.) que permiten el acceso no autorizado a diferentes funcionalidades ofrecidas por una o varias aplicaciones. Funcionalidades de administrador son las más requeridas.

- **A8 Falsificación de peticiones en sitios cruzados (CSRF)**: Riesgo relacionado al envío de peticiones sobre una aplicación vulnerable haciéndole creer que son legítimas, teniendo como premisa que muchas aplicaciones usan cookies para la autenticación.

- **A9 Utilización de componentes con vulnerabilidades conocidas**: Se refiere al uso de componentes tales como: Frameworks, librerías y otros que se conocen son vulnerables y aun así son implementados, permitiendo que un atacante pueda hacer uso de esta vulnerabilidad.

- **A10 Redirecciones y reenvíos no validados**: Los atacantes aprovechan el uso de redirecciones a otras páginas no confiables dirigiéndolos a sitios de phishing o malware, o utilizar reenvíos para acceder a páginas o sitios no autorizados.

2.4.3. Análisis de riesgos.

El apartado de seguridad OWASP Top 10 2013 identifica de manera genérica los riesgos de seguridad que se han detectado en diferentes aplicaciones, de todo tipo de organización, a nivel mundial. Además da a conocer información sobre la probabilidad e impacto en que dichos riesgos se materialicen sobre las aplicaciones web, este análisis de riesgos se lleva a cabo mediante la metodología de riesgos OWASP.

El descubrimiento de vulnerabilidades es importante pero más importante aún es poder estimar el riesgo encontrado y asociarlo al negocio.

Los riesgos los podemos definir o encontrar en las diferentes etapas del ciclo de vida del desarrollo de un software, es decir identificamos problemas de seguridad en el diseño o arquitectura de una aplicación, en las pruebas de código, en las pruebas de penetración o cuando la aplicación este en producción y se vea comprometida.

Por eso es importante estimar la gravedad de los riesgos asociados al negocio y tomar una decisión informada en base a la evaluación

realizada. Esto ayudará a no enfocarnos en riesgos menores sin tener en cuenta los riesgos más graves que son los menos conocidos.

Debemos tener en cuenta que no toda vulnerabilidad que es fundamental para organización puede no ser importante para otra.

La Metodología de Evaluación de Riesgos de OWASP consiste en 6 etapas:

Identificación del riesgo: Consiste en identificar agentes de amenazas, vulnerabilidades que pueden ser afectadas por los agentes de amenazas y el impacto sobre el negocio en caso de que se materialice la amenaza.

Estimar factores de probabilidad: Una vez encontrados los riesgos debemos estimar la probabilidad en que una vulnerabilidad en particular sea descubierta y explotada. Podemos estimar la probabilidad de manera cualitativa, pero para mayor certeza podemos emplear un análisis cuantitativo. Los factores a analizar son: agentes de amenazas y vulnerabilidades.

Estimar factores de impacto: Al igual que los factores de probabilidad estos factores deben ser estimados para corroborar el daño que puede provocar las vulnerabilidades encontradas sobre la infraestructura informática y el negocio.

Determinar la severidad del riesgo: Para determinar qué tan severo es el riesgo se debe emplear la matriz de severidad con respecto a los niveles obtenidos de los factores de probabilidad y de impacto.

Likelihood and Impact Levels	
0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

Figura 2.3: Escala para determinar niveles de probabilidad e impacto [10].

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Figura 2.4: Matriz para determinar la severidad del riesgo [10].

Priorizar planes de acción: Luego de clasificar los riesgos debemos desarrollar una lista con los riesgos de mayor prioridad y a estos dar una solución inmediata.

Personalizar el modelo de clasificación del riesgo: Es fundamental crear un modelo de clasificación de riesgos para las aplicaciones del negocio. Se debe considerar adicionar factores de riesgos, personalizar y ponderar dichos factores.

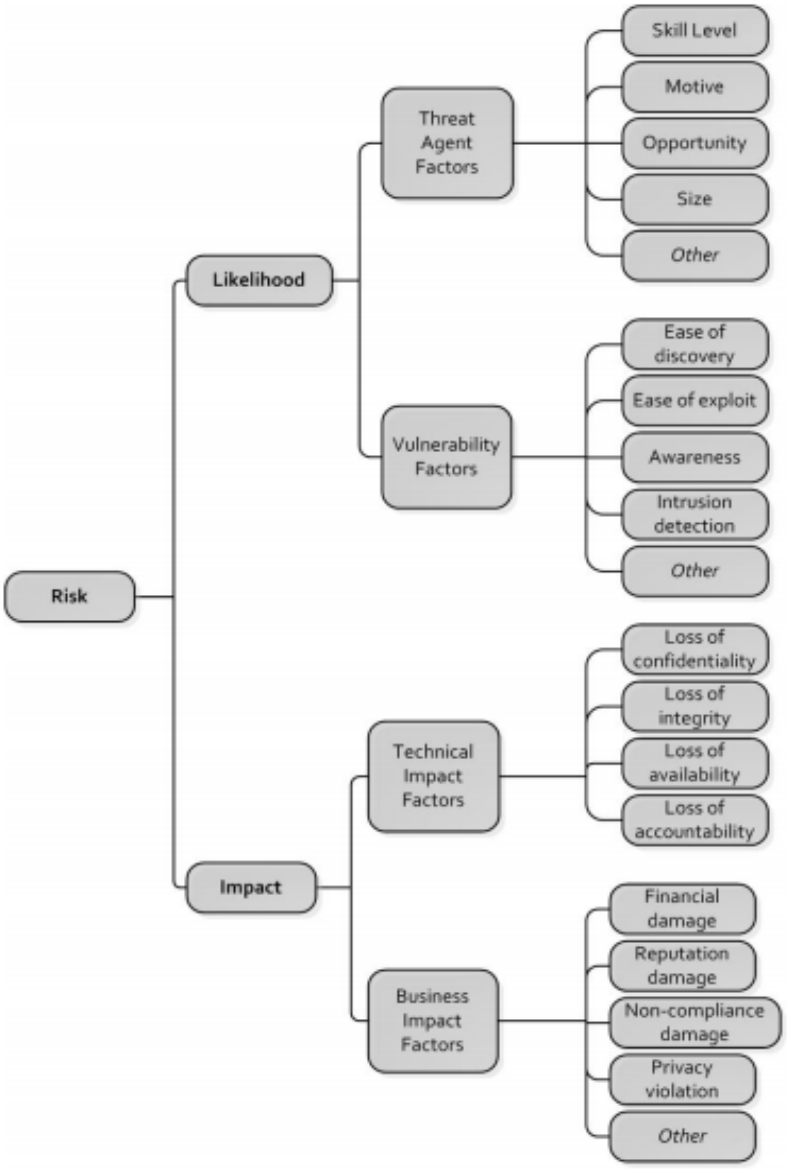


Figura 2.5: Descomposición del riesgo según la metodología de evaluación del riesgo de OWASP [9].

2.5. Auditoría Informática.

Se define como auditoría en general al examen objetivo, sistemático y profesional practicado con posterioridad a la ejecución de las operaciones o transacciones, con el objeto de emitir un informe o diagnóstico que derive comentarios, conclusiones y recomendaciones.

Aunque en la actualidad existen muchos tipos de auditoría no todos emiten una opinión sobre registros, sistemas, operaciones o actividad en particular.

La auditoría informática se define como la revisión práctica que se hace a los recursos informáticos con lo que cuenta una organización, con la finalidad de emitir un informe sobre la situación real en la que se desarrollan y se utilizan estos recursos.

Mediante este proceso se busca determinar aquellos riesgos que pudieran comprometer la confidencialidad, integridad y disponibilidad de una aplicación.

Las auditorías pueden ser internas o externas. Las internas son llevadas a cabo por personal capacitado de la organización los cuales revisan aspectos de interés para la administración, mientras que las auditorías externas son llevadas a cabo por empresas particulares que ofrecen servicios de auditoría dando un juicio imparcial de la evaluación realizada en la organización que solicitó sus servicios.

El propósito de la auditoría informática es verificar que todos los recursos informáticos (información, equipos, personal, dinero, software, etc.) estén

coordinadamente vigilados por la gerencia salvaguardando adecuadamente los activos, manteniendo la integridad de los datos y sistemas, proporcionando información confiable, alcanzando los objetivos organizacionales y utilizando los recursos eficientemente.

2.6. OWASP Top 10 vs WASC.

La problemática de seguridad que existe en las aplicaciones web se debe a siguientes factores:

- Aplicativos cada vez más complejos y dinámicos,
- Fáciles de comprometer,
- Accesibles desde internet,
- No están enfocados en la seguridad sino a la funcionalidad

Estos y otros factores provocan que muchas de las aplicaciones web tengan fallos de seguridad porque al desarrollar la aplicación no se dispone, en mucho de los casos, de una directriz que permita el desarrollo seguro; ocasionando pérdidas económicas y fugas de información de pequeñas o grandes empresas a nivel mundial.

Actualmente existen organizaciones que se involucran proporcionando documentación a la comunidad dando a conocer, mediante estudios realizados por expertos, las diferentes problemáticas de seguridad que se suscitan en las aplicaciones web.

Web Application Security Consortium (WASC) al igual que Open Web Application Security Project (OWASP) son organizaciones sin fines de lucro, las cuales están conformados por grupos de personas y asociaciones expertas en seguridad informática que contribuyen con guías y artículos de seguridad que son de libre acceso, participan en foros y charlas a nivel mundial. Además proporcionan software para el análisis de vulnerabilidades y documentación útil sobre amenazas de seguridad web.

Entre los principales proyectos que podemos comparar entre dichas organizaciones tenemos el OWASP Top 10 2013 y WASC: Threat Classification versión 2.0 (2010); ambos dan las pautas y directrices para tener software seguro disminuyendo considerablemente el riesgo que se puedan presentar en una aplicación web.

“OWASP Top 10 2013” clasifica los riesgos en un ranking de acuerdo a las vulnerabilidades mayormente encontradas en sitios web, mientras que “WASC Threat Classification” clasifica los riesgos de acuerdo a los diferentes ataques a los sitios web encontrados relacionándolos con una vulnerabilidad específica.

WASC Threat Classification determina cada ataque y vulnerabilidad en ciertas etapas del desarrollo de software (Diseño, Implementación y Despliegue) mientras que OWASP Top 10 2013 presenta la vulnerabilidad de manera general con sus respectivas medidas de mitigación.

OWASP Top 10 2013 puede ser usado y tomado como referencia inicial para implementar técnicas de seguridad en aplicaciones web, mientras que WASC Threat Classification es usado comúnmente en un sistema de seguridad un poco más maduro.

Por estos motivos consideramos que el apartado OWASP Top 10 2013 le sería de gran utilidad a la compañía IIASA para analizar, detectar y mitigar los riesgos encontrados; evaluando el riesgo y el impacto en la organización dando a conocer preceptos de seguridad e instaurar en la compañía el enfoque hacia el desarrollo y mantenibilidad segura de las aplicaciones web desplegadas en la compañía.

CAPÍTULO 3.

3. LEVANTAMIENTO DE INFORMACIÓN.

3.1. Levantamiento de requerimientos.

A continuación detallaremos los requerimientos solicitados por parte del departamento de sistemas de la compañía IIASA:

1) Escaneo de vulnerabilidades de las aplicaciones web de intranet consideradas importantes para el negocio.

   - Listado de aplicaciones web que serán analizadas, las mismas que deben estar detalladas por nombre de aplicación, opciones habilitadas, número de usuarios autorizados y departamento responsable.

- Empleo de una computadora de la institución, la cual se le instalará el software de detección de vulnerabilidades seleccionado.

- El escaneo de las aplicaciones debe hacerse en un horario en donde no interrumpa las labores cotidianas de la compañía y de manera controlada para que no afecte datos almacenados.

2) Documentar los resultados del análisis de vulnerabilidades encontradas en las aplicaciones web estudiadas.

   - Los resultados serán presentados a la gerencia de sistemas con todas las vulnerabilidades encontradas por cada aplicación web.

   - No se tomaran en cuenta los resultados “falsos positivos” para la documentación ya que cada vulnerabilidad encontrada será analizada y corroborada con el código fuente de cada aplicación analizada.

3) Sugerir e implantar soluciones que mitiguen los riesgos encontrados.

   - El apartado OWASP Top 10 2013 presenta las diferentes soluciones para mitigar los riesgos nombrados en su apartado, de manera que pueda ser implementado en cada aplicación analizada.

   - Mejorar el uso de la plataforma intranet con el empleo del protocolo https para la transferencia de información entre el cliente y el servidor.

   - Gestionar el uso adecuado de cookies y sesiones de usuarios.

- Uso de la herramienta ESAPI de OWASP para la gestión de seguridad en las aplicaciones web de la compañía.

4) Instruir a los desarrolladores del departamento buenas prácticas de programación.

- Dar a conocer las diferentes vulnerabilidades encontradas de cada aplicación analizada a los desarrolladores de sistemas.

- Establecer directrices de seguridad que incluya el desarrollo seguro y la detección de vulnerabilidades para nuevas aplicaciones web internas o externas y aquellas a las que se les da mantenimiento continuo.

3.2. Información del ambiente de sistemas.

3.2.1. Red de datos.

El centro de cómputo de la compañía se encuentra ubicado en Guayaquil, donde funciona su local matriz. La comunicación con las agencias y sucursales se hace mediante la contratación de un proveedor de internet, que brinda enlaces dedicados para la comunicación con las sucursales y agencias.

Alrededor de 600 equipos se conectan a la red interna de la compañía mediante el uso de PC's, laptops, tabletas, smartphones, etc., con el propósito de acceder a los servicios que la compañía brinda a sus colaboradores. La red está constituida principalmente por routers

perimetrales, switch core, un firewall que a la vez es un IDS (Sistema de detección de intrusos), base de datos, servidor de archivos, servidor de aplicaciones, un equipo As400 y demás equipos que permiten la conectividad entre departamentos.

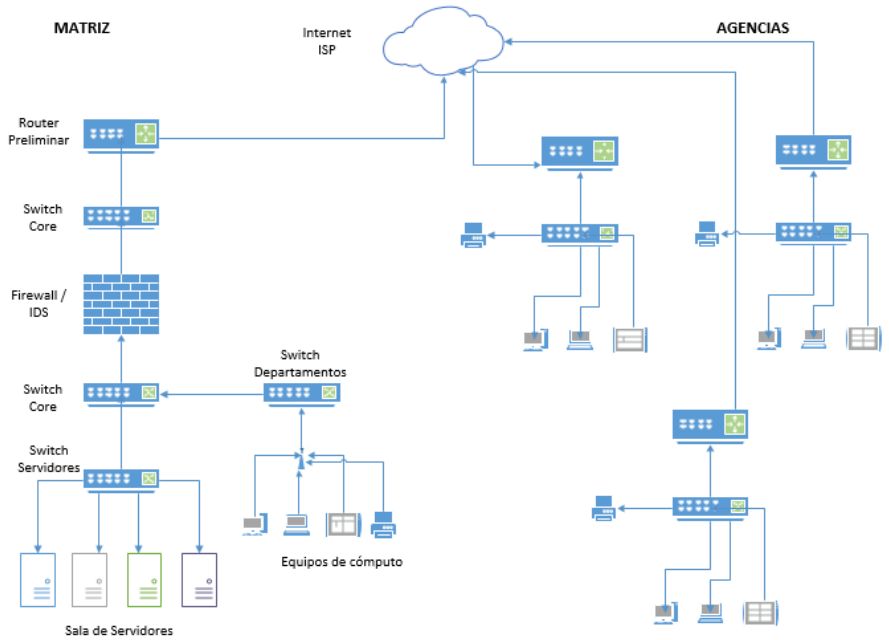


Figura 3.1: Red interna IIASA Ecuador.

En la instalación matriz de la compañía se emplea un cableado estructurado vertical y horizontal para las diferentes comunicaciones de red, telefonía y vídeo.

El cableado vertical se lo emplea para la comunicación entre las diferentes plantas de la instalación principal, para ello se usan 2 cuartos de comunicación que se conectan a un “backbone” principal; se emplea cableado UTP categoría 6ª.

En cambio el cableado horizontal se lo emplea para enlazar los cuartos de comunicación con las estaciones de trabajo, el cableado usado es par trenzado UTP categoría 6a.

El firewall de la empresa filtra el acceso a equipos sensibles de la institución (DMZ), controla el acceso libre a internet, establece reglas para acceso a intranet desde el internet y habilita ciertos servicios necesarios para la compañía (telnet, ftp, smtp, http, etc,).

3.2.2. Equipos.

El área técnica es la encargada de supervisar el correcto funcionamiento de los equipos de TI de la compañía así como también dar soporte al usuario en caso de que requiera.

El área de software es la encargada el correcto funcionamiento de los sistemas de la compañía, brindando el soporte necesario a los usuarios; también se encargan de realizar nuevos programas o modificaciones a programas existentes.

A continuación detallaremos los equipos de tecnología con los que cuenta la compañía:

Tabla 1: Equipos, aplicaciones y servicios.

Plataf.	Rol	S.O	Servicio	Programa	Obs.
Intranet	Aplicación	Windows Server 2008 Standard	Servidor de aplicaciones web	Jboss As 7.1.1 Final	Java 1.7.0_51

	Base de Datos	Windows Server 2008 Standard	Base de datos	Microsoft SQL Server 2008	
	Archivos	Windows Server 2008 Standard	Tareas programadas		Java 1.7.0_51
			Repositorio de fuentes	S.V.N 1.7.10	
CRM	Aplicación	Windows Server 2003 Standard	Servidor de aplicaciones web	Internet Information Services IIS 6.0	
	Base de Datos	Windows Server 2008 Standard	Base de Datos	Microsoft SQL Server 2008	
LDAP	Active Directory	Windows Server 2008 Standard	Gestión de directorios y usuarios en la red		
ERP	As400	Versión 7 Release 1	ERP / FTP / SMTP	DBS	Versión Java 1.6.0

3.2.3. Políticas para el acceso a información y uso de recursos.

La compañía estableció que la información recabada en el presente documento no debe ocasionar futuros problemas relacionados a la divulgación de información de aquellas aplicaciones que serán objeto de estudio.

Para acceder a información sensible como: usuario de intranet, usuarios de bases de datos, fuentes de aplicaciones, manuales de usuario y técnicos, se solicitará el acceso a ella mediante la elaboración de un memo que es regularizado por el departamento de auditoría y de sistemas.

3.3. Aplicaciones del negocio.

Podemos listar las diferentes aplicaciones que ofrece la compañía para uso de sus colaboradores:

- DBS (Dealer Business System): Es un sistema multiusuario proporcionado por Caterpillar para uso de sus franquicias o dealers en todo el mundo, fue desarrollado por la compañía Accenture. Sus principales funciones son: Procesar órdenes de trabajo, control de inventario, servicios por reparación de maquinarias, renta de equipos, venta de maquinarias, facturación, procesos de pedidos a fábrica, etc. El uso de este programa es constante. No es una aplicación web.

- CRM (Customer Relationship Managment): Es una aplicación web destinada para dar seguimiento a posibles clientes, manejo de cartera de clientes, oportunidades de negocio, marketing, etc. Desarrollado también por la compañía Accenture.

- Portal web: Sitio web público de la compañía en donde se describe información de la compañía, los productos y servicios que se ofrecen.

Dicho sitio no está alojado en las instalaciones de la compañía sino en un Hosting particular.

- Intranet: Es la plataforma en la cual se encuentran muchas de las aplicaciones web operativas y de decisiones de la compañía, está formada por alrededor de 165 aplicaciones web, las cuales están segmentadas por departamento. Al igual que el DBS, esta plataforma es usada recurrentemente por los usuarios de la compañía para sus labores cotidianas.

- CDT: Es una aplicación web en donde los técnicos/mecánicos de la compañía mejoran sus habilidades para la resolución de problemas en maquinaria pesada.

- CAT-Builder: Es una aplicación de escritorio que desglosa los diferentes mantenimientos que se deben realizar en una maquinaria supervisando así cualquier anomalía en dichos equipos.

Todas las aplicaciones pueden compartir información gracias a procesos de migración facilitando el acceso en las distintas plataformas usadas en la compañía.

3.3.1. Intranet.

Una intranet es una red corporativa o local de una organización que está estructurada por el conjunto de tecnologías que sustentan el internet, mientras que una extranet es la intranet de una organización pero accesible desde el internet, de manera que los recursos pueden

ser accedidos por empleados que no encuentran en las instalaciones de la compañía.

La intranet de la compañía está conformada por diferentes aplicaciones, muchas de ellas desarrolladas de manera interna por el grupo de desarrolladores de software de la compañía y otras creadas por proveedores externos. Esta plataforma permite a los colaboradores de la compañía acceder a información relevante y crítica que puede ser usada o manipulada para fines no confiables.

Actualmente la comunicación entre el cliente (navegador) y el servidor web se lo realiza mediante el protocolo HTTP, que es parte de la familia de protocolos de internet que permiten la transferencia de datos entre computadores.

Es usada diariamente por personal administrativo y personal técnico/mecánico del área de talleres, por lo que debe estar disponible las 24 horas del día. Los días sábados se deshabilita por alrededor de 90 minutos por política del departamento de sistemas para mantenimiento de equipos.

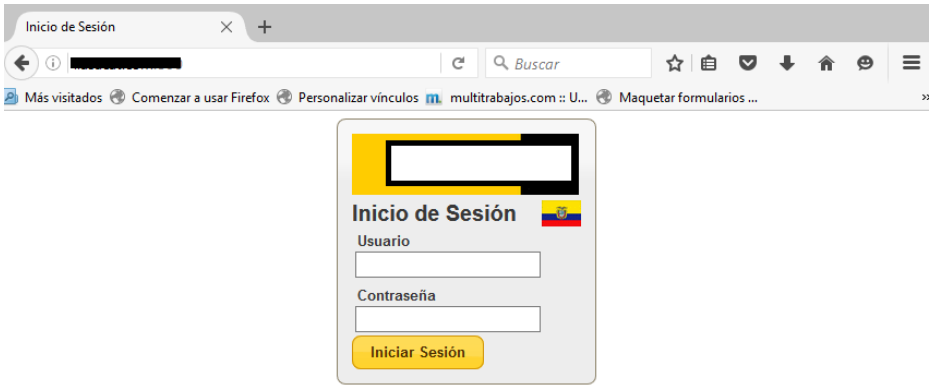


Figura 3.2: Acceso a la intranet de IIASA.

Figura 3.3: Portada inicial del interno.

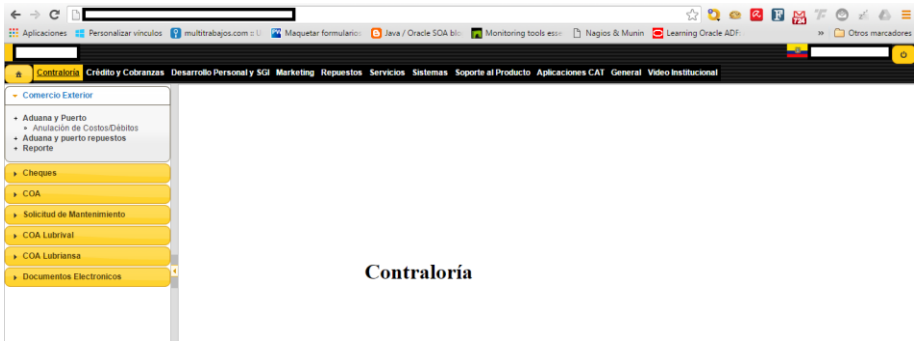


Figura 3.4: Opciones de la intranet.

3.3.2. Arquitectura de la intranet.

La intranet de la organización tiene una arquitectura cliente – servidor de 3 capas y 3 niveles, que es un modelo de aplicación distribuida en que las tareas se reparten entre los diferentes proveedores de servicios o recursos, llamados servidores, y los demandantes, llamados clientes.

Esta plataforma consulta información de diferentes repositorios que son usados por las distintas aplicaciones web y de escritorio es decir, accede a bases de datos tales como: SQL Server, DB2, Access y MySQL dando lugar a que dicha plataforma sea vulnerable a distintos ataques sino se lleva un control de seguridad adecuado.

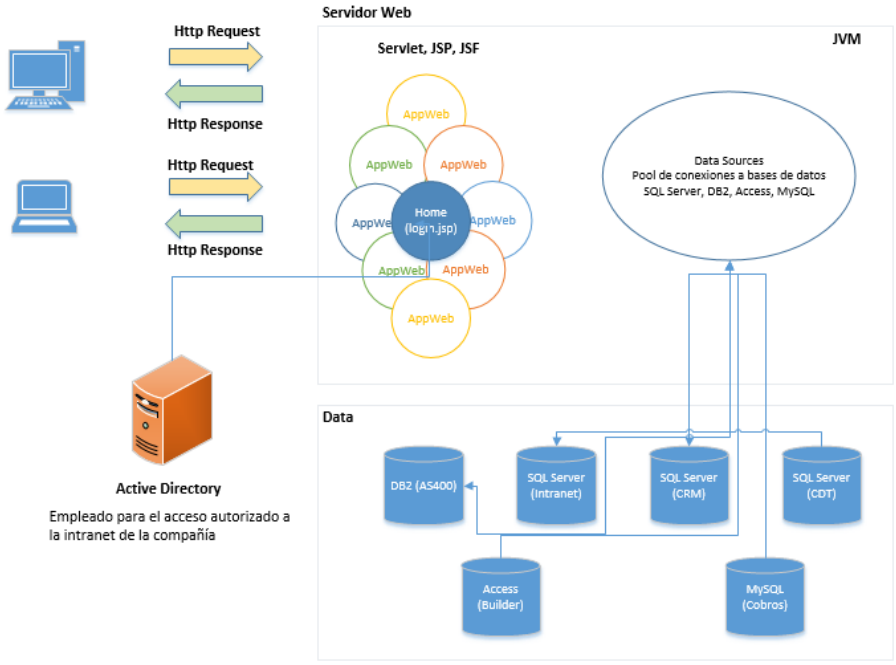


Figura 3.5: Arquitectura de la intranet de IIASA.

Cabe indicar que el acceso a las diferentes opciones que despliega la intranet está delimitada por permisos de usuario, que se autorizan previa revisión del departamento de auditoría y del departamento al cual pertenece el usuario. Solo se asignan las opciones necesarias para su puesto de trabajo y ningún otro usuario puede hacer uso de credenciales que no le pertenezca.

3.3.3. Tecnologías implementadas.

Todas las aplicaciones web de la intranet están desarrolladas enteramente en lenguaje Java y emplea diferentes tecnologías como: Struts 1, JSF 1.x, JSF 2.x, PrimeFaces, Servlets, Jquery, Web Services, etc. Estas aplicaciones se encuentran desplegadas en un servidor web Jboss 7.1.1 con acceso a base de datos SQL Server, DB2, Microsoft Access y MySQL; dichas aplicaciones conforman la intranet de la compañía, la misma que puede ser accedida interna o externamente.

Los navegadores web permitidos por la compañía para el uso de intranet son: Internet Explorer, Mozilla Firefox y Google Chrome.

En la compañía se recomienda el uso del navegador Internet Explorer, esto debido a que aplicaciones web antiguas y de mayor demanda tienen un mejor rendimiento en este navegador. Las nuevas aplicaciones desarrolladas son compatibles con los demás navegadores mencionados.

La compañía cuenta con equipos de escritorio y laptops con sistema operativo: Windows XP, Windows Vista, Windows 7 y Windows 8. En un futuro cercano el departamento de sistemas planea adaptar todas las aplicaciones web de la intranet a los navegadores más populares del mercado.

Anteriormente el servidor de correo empleado era Lotus Notes pero este año se migró a Gmail, dando facilidad a los empleados a acceder desde cualquier parte del mundo a su cuenta de correo empresarial, así como también el acceso a las demás aplicaciones que ofrece Google (Drive, Calendar, Google Docs., etc.).

3.4. Alcance del proyecto.

3.4.1. Alcance.

Para la implementación del presente proyecto se ha elegido a la intranet de la compañía como plataforma de estudio. La elección de dicha plataforma se debe al acceso a las fuentes de las diferentes aplicaciones web desarrolladas y al uso que los empleados les dan para realizar sus diferentes labores cotidianas.

Esta intranet es un conjunto de aplicaciones que cuentan con diferentes funcionalidades para determinados departamentos, el acceso a la intranet se lo realiza mediante una aplicación matriz donde los empleados ingresan sus credenciales y acceden a sus opciones previamente autorizadas.

Para el presente estudio se considerará solo 50 aplicaciones que se encuentran desplegadas en el servidor de aplicaciones. Serán elegidas de acuerdo a su criticidad en el negocio de la compañía y serán proporcionadas por el departamento de sistemas.

Después de conocer las aplicaciones web elegidas, se llevará a cabo la detección de vulnerabilidades según el apartado OWASP Top 10 2013, el análisis de los riesgos encontrados y la implementación de soluciones proporcionada por el mismo apartado.

A continuación se detallan las aplicaciones web que serán objeto de estudio, estarán identificadas por un código relacionado al departamento donde se lo usa y una pequeña descripción de cada aplicación.

Tabla 2: Aplicaciones web de la intranet del negocio.

#	Dpto.	Código	Desc. Problema
1	Contraloría	WEB-APP-C01	Sistema para la creación y anulación de cheques
2		WEB-APP-C02	Sistema que muestra cheques relacionados a comprobantes
3		WEB-APP-C03	Sistema de compras grupo Lubrival
4		WEB-APP-C04	Sistema de compras grupo IIASA
5		WEB-APP-C05	Sistema para generación de información tributaria
6		WEB-APP-C06	Sistema que permite la conciliación de documentos

7		WEB-APP-C07	Sistema para el control de facturas registradas por la compañía Talleres
8		WEB-APP-C08	Sistema para generar información de cuentas pendientes de proveedores
9		WEB-APP-C09	Sistema que controla el flujo de cheques aprobados
10		WEB-APP-C10	Sistema de liquidaciones de gastos de la compañía
11		WEB-APP-C11	Sistema para la consulta de inventario de máquinas y repuestos
12		WEB-APP-C12	Sistema de aprobación de cheques emitidos
13		WEB-APP-C13	Sistema para control de proveedores y autorizaciones SRI
14		WEB-APP-C14	Sistema de Guías de Remisión
15		WEB-APP-C15	Opciones varias para contraloría
16		WEB-APP-C16	Sistema para la consulta de inventario de máquinas y repuestos
17		WEB-APP-C17	Sistema para el control de cheques entregados a proveedores
18	Crédito y Cobranzas	WEB-APP-CC1	Sistema recibos de cobro a clientes
19		WEB-APP-CC2	Sistema para la gestión de cartera de clientes
20		WEB-APP-CC3	Sistema de cobranzas
21		WEB-APP-CC4	Sistema de control de recaudadores (registro de valores)
22		WEB-APP-CC5	Opciones varias para cobranzas

23		WEB-APP-CC6	Estado de cuentas de clientes
24	Maquinaria	WEB-APP-M01	Aprobaciones de comisiones de vendedores
25		WEB-APP-M02	Sistema para recolección de información para comisiones vendedores
26		WEB-APP-M03	Opciones varias del departamento
27		WEB-APP-M04	Opciones varias para comisiones de vendedores
28		WEB-APP-M05	Sistema de solicitud de crédito de clientes
29		WEB-APP-M06	Sistema de control de maquinarias
30		WEB-APP-M07	Sistema para generación de hoja de margen compañía rental store
31	Repuestos	WEB-APP-R01	Sistema de Costo de repuestos
32		WEB-APP-R02	Opciones varias de repuestos
33		WEB-APP-R03	Sistema para ajustes de precios y descuentos
34		WEB-APP-R04	Sistema para actualizar precios de repuestos y cotizaciones
35		WEB-APP-R05	Opciones varias pedidos
36		WEB-APP-R06	Sistema que monitorea los pedidos de repuestos a fábrica
37		WEB-APP-R07	Sistema de importaciones
38		WEB-APP-R08	Opciones de consultas de ventas de repuestos

39		WEB-APP-R09	Opciones para el control de emisión de N/C
40	Servicios	WEB-APP-S01	Sistema para el control de lubricantes en talleres
41		WEB-APP-S02	Sistema de marcaciones de técnicos en órdenes de trabajo
42		WEB-APP-S03	Sistema de presupuestos de talleres
43		WEB-APP-S04	Opciones varias de talleres
44		WEB-APP-S05	Sistema que gestiona las compras de talleres
45		WEB-APP-S06	Sistema de guías de remisión de talleres
46	Sistemas.	WEB-APP-I01	Control de documentos de proveedores
47		WEB-APP-I02	Sistema que verifica legalidad de clientes y fondos
48		WEB-APP-I03	Sistema de presupuestos o ventas de rental store
49		WEB-APP-I04	Sitio web de la intranet historia, misión, visión, directorio telefónico, etc
50		WEB-APP-I05	Sistema que administra el login y presenta las opciones asignadas a cada usuario.

Del grupo de aplicaciones seleccionadas, describimos con más detalle las aplicaciones que son consideradas relevantes para el trabajo diario de los empleados:

- **WEB-APP-C04**: Este aplicativo es considerado importante porque permite el ingreso de comprobantes que son emitidos por los

proveedores de la compañía, genera retenciones y la contabilización respectiva. Es usado ampliamente por la mayoría de asistentes de la compañía ya que cada departamento gestiona sus pagos.

- **WEB-APP-C12**: Este aplicativo permite la aprobación de pagos a proveedores, es usado por una cantidad menor de empleados pero la información que en ella se presenta es importante, porque permite decidir si el pago se hace efectivo o no mediante la emisión de cheques previa revisión de documentación que sustenta el pago respectivo.

- **WEB-APP-CC1**: Sistema que permite el registro de los cobros realizados a los clientes y se indica si el pago fue hecho en efectivo, cheque o transferencia bancaria. Es considerado importante porque maneja información de cartera de clientes y su situación financiera con respecto a la institución. Es usado solamente por vendedores y asistentes del área de cobranzas.

- **WEB-APP-M05**: Aplicativo que permite la aprobación de las “solicitudes de crédito” realizada por los clientes al momento de adquirir maquinaria pesada a la compañía. Este aplicativo es usado por los altos ejecutivos para decidir la aprobación o no del crédito. Es importante porque maneja información de crédito de los clientes de la compañía.

- **WEB-APP-R03**: Aplicativo que permite el ingreso de ajustes de precios de repuestos según descuentos especiales a clientes o debido a cambios de precios realizados después de emitida la cotización. Aplicativo usado por personal de repuestos y que modifica valores a cobrar en los repuestos solicitados por los clientes según sea el caso.

- **WEB-APP-R04**: Sistema que aprueba los ajustes de precios realizados en el aplicativo anterior. Es catalogado importante porque pueden presentarse pérdidas económicas al realizar cambios de precios para beneficio de clientes y/o empleados.

- **WEB-APP-S03**: Sistema que permite la generación de presupuestos por los servicios ofrecidos en los talleres de la compañía. Maneja información de precios y descuentos. Es usado por los supervisores de talleres y aprobado por el cliente en el caso de aceptar el presupuesto. Cabe indicar que la aprobación del cliente no es a través del sistema.

- **WEB-APP-I05**: Es el sistema más importante de la intranet porque permite el ingreso de los empleados a las demás funciones. Es considerada importante porque es el punto de partida para conocer las demás aplicaciones con las que cuenta la intranet de la compañía.

Después de conocer a breve rasgos las diferentes aplicaciones web con las que contaremos para el presente estudio, debemos continuar con la detección de vulnerabilidad, con el fin de detectar fallos de seguridad en cada aplicativo web, para lo cual usaremos un escáner de vulnerabilidades.

Esta herramienta emitirá un reporte con todas las vulnerabilidades encontradas en cada aplicación dando un criterio, en forma de alertas, del nivel de riesgo encontrado (alto, medio o bajo). Se tomarán en cuenta todas las vulnerabilidades halladas para el análisis de riesgos respectivo.

Para el análisis de riesgos usaremos la metodología, proporcionada también por OWASP, llamada “Evaluación de Riesgos de OWASP”.

Este análisis nos permitirá determinar las vulnerabilidades con niveles de severidad crítico, alto, medio o bajo para la compañía.

Debemos considerar un factor importante, el costo/beneficio, en el caso de implementar una solución. Se debe tomar en cuenta los requerimientos indicados por el área de sistemas.

3.4.2. Metodología.

La metodología a llevar a cabo para el presente estudio se detallará a continuación:

1) Definición y detección.

- Definir las aplicaciones web del negocio: Se mantendrá una reunión con el jefe de sistemas para que se nos indique las 50 aplicaciones a analizar, él dará a conocer el criterio por el cual se eligen dichas aplicaciones y se determinará también que departamentos se verán involucrados.

- Definir la herramienta de software para la detección o escaneo de vulnerabilidades en las aplicaciones web elegidas.

- Creación de un usuario temporal para acceder a la intranet de la compañía, el mismo que estará disponible hasta la finalización de la detección de vulnerabilidades (30 días)

- Detección de vulnerabilidades de las aplicaciones web elegidas cuyos riesgos estén enmarcados en el apartado OWASP Top 10 2013.

- La detección de vulnerabilidades será realizada en las instalaciones de la compañía y no se podrá hacerlo de manera remota.

- El escaneo de vulnerabilidades debe hacerse de manera controlada de manera que no afecte las actividades del negocio.

- El escaneo de vulnerabilidades se lo realizará los fines de semana y en ciertas ocasiones se lo hará los días laborables a partir de las 5pm.

2) Análisis de riesgos según la Metodología de Evaluación de Riesgos OWASP.

- Identificación del riesgo
- Estimación de la probabilidad
- Estimación del impacto
- Determinación de la severidad del riesgo
- Priorizar los planes de acción
- Personalizar el modelo de clasificación del riesgo

3) Medidas preventivas y correctivas a implementarse.

- Implementar las soluciones sugeridas por el apartado OWASP Top 10 2013, para los riesgos cuya severidad sea crítica, alta o media teniendo como factor preponderante el costo que implica la mitigación del riesgo.
- Implementar soluciones sugeridas por el departamento de sistemas

- Implantar solución ESAPI para detectar intrusión de código malicioso en el servidor de aplicaciones o aplicaciones web.
- Establecer procedimientos y buenas prácticas según los riesgos encontrados.

CAPÍTULO 4.

4. PLAN DE DETECCIÓN DE VULNERABILIDADES Y ANÁLISIS DE RIESGOS ENCONTRADOS.

4.1. Plan de detección de vulnerabilidades a las aplicaciones web.

4.1.1. Identificación de vulnerabilidades.

Para identificar las vulnerabilidades en las aplicaciones web debemos tener claro los conceptos de vulnerabilidad, amenazas y riesgo. El proceso de identificar debilidades en un aplicativo es algo lento y laborioso; se lo puede hacer de manera manual o automática.

La búsqueda de vulnerabilidades implica conocimientos en el uso de programa para detectar vulnerabilidades, en lenguajes de programación, comunicación de datos, bases de datos, administración

de servidores, etc., para una fácil comprensión de las debilidades que se pueden presentar en las aplicativos escaneados.

Esta búsqueda implica las siguientes fases: Reconocimiento, Identificación y Validación.

- Reconocimiento: En esta fase se define los objetivos y los sistemas asociados al entorno. Es decir, conocer el servidor web donde es desplegado el aplicativo, el lenguaje de programación empleado, determinar si usa algún tipo de cifrado de información o empleo del protocolo SSL.

- Identificación: Una vez obtenida la información global del entorno es posible identificar los agentes de amenazas y los vectores de ataques que pueden materializar el riesgo que se puede presentar al encontrar una vulnerabilidad.

- Validación: En esta fase se comprueba si la vulnerabilidad es explotable y si lo es, se realiza el respectivo análisis para representar el riesgo asociado a la vulnerabilidad.

4.1.2. Herramientas de software para la detección de vulnerabilidades en aplicaciones web.

Las herramientas de software para la detección de vulnerabilidades realizan pruebas de manera automática en función a una base de conocimiento, que les permite generar instrucciones para determinar las diferentes vulnerabilidades que pueden comprometer el aplicativo

web, no tienen acceso al código fuente más solo realizan pruebas funcionales.

Explicaremos a continuación algunas herramientas de software útiles para la detección de vulnerabilidades, las mismas que son Open Source:

- Grabber

  Es una herramienta agradable desarrollada en lenguaje de programación Python, que puede detectar vulnerabilidades de seguridad tales como: Cross Site Scripting, Inyección SQL, Pruebas Ajax, inclusión de archivos.

  Se recomienda el uso, para aplicaciones web pequeñas, pues toma demasiado tiempo escanear aplicaciones más grandes y complejas. No ofrece una interfaz gráfica y no exporta los resultados obtenidos.

- Vega

  Es un programa que permite la detección de vulnerabilidades en aplicaciones web, está disponible en sistemas operativos: Linux, Windows y OSx. Puede detectar vulnerabilidades de seguridad tales como: Inyección SQL, listado de directorios, Cross Site Scripting, inclusión de archivos.

Esta herramienta puede ser usada para interceptar las peticiones web mediante la función de proxy manipulando las URL y realizar pruebas específicas.

- Zed Attack Proxy

  Es una herramienta desarrollada por OWASP programada bajo el lenguaje Java, disponible para los sistemas operativos Windows, Linux y OSx. La herramienta es fácil de usar y puede detectar vulnerabilidades de seguridad tales como: Inyección SQL, Cross Site Scripting, inclusión de archivos, gestión de sesiones, configuraciones de servicios débiles, entre otros. Las vulnerabilidades que puede detectar están alineadas con el proyecto OWASP Top 10 2013.

  Al igual que la herramienta anterior se puede usar como proxy para interceptar peticiones y así manipular el envío de parámetros sobre las URL de las aplicaciones y detectar vulnerabilidades. Presenta informes en formato HTML y PDF que puede ser exportado para la presentación de un informe final.

- W3af

  Esta herramienta está desarrollada en Python y ofrece una interfaz gráfica intuitiva o el uso mediante comandos, permite la detección de alrededor de 200 tipos de vulnerabilidades entre las cuales tenemos: inyección SQL, Cross Site Scripting, entre otros. Se

puede instalar en Linux y Windows y está en constante actualización.

El uso de estas herramientas, se aconseja, que deba realizarse de manera controlada es decir, no afectar de ninguna manera la integridad de la información de los sistemas escaneados y garantizar la disponibilidad de los servicios atacados.

4.1.3. Procedimiento para la detección de vulnerabilidades.

Para la detección de vulnerabilidades usaremos la herramienta gratuita Zed Attack Proxy debido a las ventajas que ofrece en la detección de vulnerabilidades relacionadas a los riesgos más comunes del apartado OWASP Top 10 2013.

La compañía nos ha concedido el acceso a sus instalaciones por lo que el programa antes mencionado será instalado en un equipo del departamento de sistemas y desde allí se realizará el escaneo de vulnerabilidades a las aplicaciones de intranet.

Es necesaria la creación de un usuario temporal, al que se le concede el acceso a las diferentes funciones asociadas a las aplicaciones web que se desea analizar en el momento. Este usuario debe seguir el procedimiento instaurado en la compañía para la creación de usuarios de intranet es decir, debe ser aprobado por auditoría y por los jefes departamentales.

Cabe indicar que este usuario será revocado una vez finalizada la detección de vulnerabilidades de todas aplicaciones web elegidas.

Configuramos la herramienta Zed Attack Proxy para interceptar las URLs de las opciones de cada aplicativo, con la finalidad de encontrar la mayor cantidad de vulnerabilidades que puedan existir.

Figura 4.1: Zed Attack Proxy (ZAP).

Esta herramienta, mediante sus procesos de escarbado, busca vulnerabilidades en cada directorio del aplicativo y genera alertas que identifican el nivel de criticidad de riesgo encontrado. Las alertas de color amarillo indican un nivel de riesgo bajo, las alertas de color naranja indican que las vulnerabilidades tienen un nivel de riesgo medio y por último las de color rojo indican un nivel de riesgo alto.

Una vez terminada la exploración de las opciones, activamos la opción “Escaneo Activo” justo en el directorio de la aplicación antes explorado, proporcionando un resultado que puede variar por cada aplicación.

Esta función puede tomar varios minutos pues depende del tamaño de cada aplicación y de los diferentes ataques que se pueden llevar a cabo en cada URL.

Figura 4.2: Progreso del escaneo activo.

Una vez finalizado el escaneo activo podemos exportar los resultados en dos formatos: HTML y XML. El reporte HTML consta en su parte inicial con un resumen de todas las alertas encontradas por nivel de criticidad (alto, medio y bajo) y en la parte inferior se detalla cada vulnerabilidad con el número de instancias que fueron encontradas.

Además describe cada vulnerabilidad con su respectiva remediación y relaciona la vulnerabilidad encontrada con otros apartados importantes (WASC y CWE).

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	3
Medium	4
Low	5
Informational	0

Figura 4.3: Resumen de vulnerabilidades encontradas por ZAP.

Para el uso de la herramienta ZAP es importante determinar lo siguiente:

- Seleccionar la aplicación web objetivo,
- Configurar como proxy para interceptar las diferentes peticiones “Post” y “Get” al servidor de aplicaciones,
- Evitar el escaneo de palabras reservadas como “mailto” porque puede darse el caso de que se envíe correo basura cuando se realice el escaneo de vulnerabilidades,
- Usar el adecuado “modo de escaneo”, generalmente se usa el “por defecto” pero pueden emplearse otros,
- Uso del modo “spider” para determinar los diferentes directorios del aplicativo web analizado.

4.2. Plan de análisis de riesgos de seguridad de las aplicaciones web según OWASP Top 10 2013.

4.2.1. Vulnerabilidades encontradas en las aplicaciones web de la compañía.

Después de la detección de vulnerabilidades de cada aplicativo web, se pudo determinar que existen varias opciones comprometidas, las cuales pueden ser manipuladas para robar información o comprometer la plataforma del negocio.

Cada vulnerabilidad detectada fue agrupada según la clasificación del apartado OWASP Top 10 2013, para esto usamos los reportes generados por la herramienta de detección. Cada vulnerabilidad descrita en el reporte tiene un identificador numérico que se relaciona con la respectiva clasificación de vulnerabilidades mantenida por WASC versión 2, mediante esta relación podemos indicar en que grupo de OWASP Top 10 2013 se ubica la vulnerabilidad detectada [9].

Todas las aplicaciones seleccionadas presentan vulnerabilidades con diferentes niveles de criticidad de riesgo. En la siguiente tabla se describen dichas vulnerabilidades con su respectiva clasificación del riesgo según OWASP Top 10 2013:

Tabla 3: Resumen de vulnerabilidades encontradas en las aplicaciones web de la intranet de la compañía.

Vulnerabilidad	OWASP Top 10 2013
Falla por Inyección SQL	A1
Inyección Remota de Comandos OS	A1
Content-Type Header Missing	A2
Cross Site Scripting (Reflejada)	A3
Directory Traversal	A4
X-Frame-Options Header Not Set	A5
Application Error Disclosure	A5
Web Browser XSS Protection Not Enabled	A5
X-Content-Type-Options Header Missing	A5
Password Autocomplete in Browser	A5
Cross-Domain JavaScript Source File Inclusion	A5
Private IP Disclosure	A5
Cookie No HttpOnly Flag	A5

Las vulnerabilidades consideradas con nivel de criticidad alto deben ser tratadas con mayor urgencia porque pueden comprometer la información o el sistema mismo, las de nivel medio deben ser tratadas pero procurando haber mitigado las de nivel alto y las de nivel bajo queda a disposición del negocio si se tratan a mediano o largo plazo.

A continuación se mostrarán los resultados de las vulnerabilidades encontradas por departamento, con el número de incidencias asociadas a cada vulnerabilidad. Llámese incidencia al número de vías por las cuales se puede propiciar un ataque.

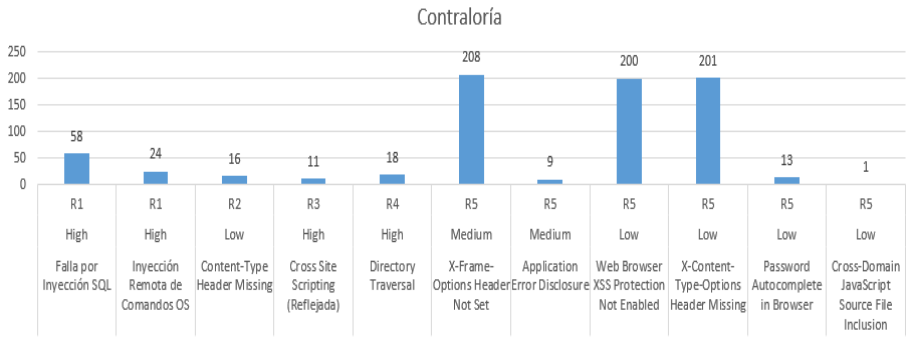


Figura 4.4: Vulnerabilidades encontradas Dpto. de contraloría.

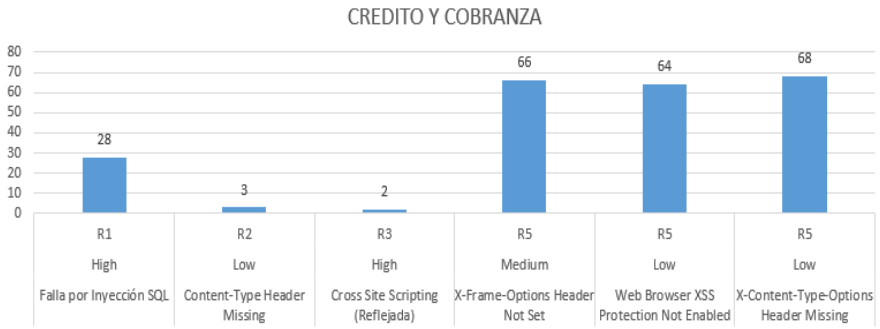


Figura 4.5: Vulnerabilidades encontradas Dpto. de crédito y cobranzas.

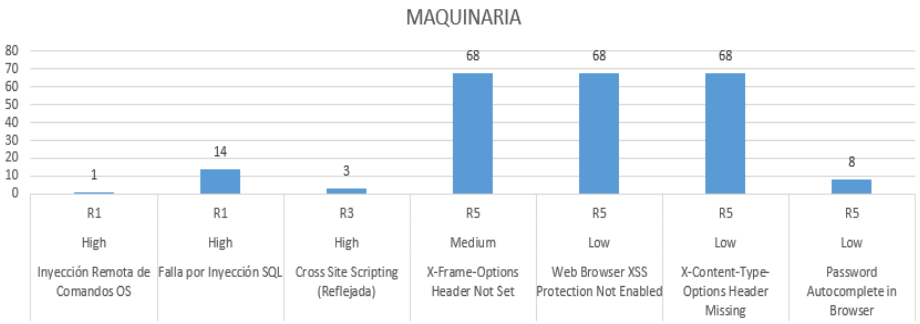


Figura 4.6: Vulnerabilidades encontradas Dpto. de maquinarias.

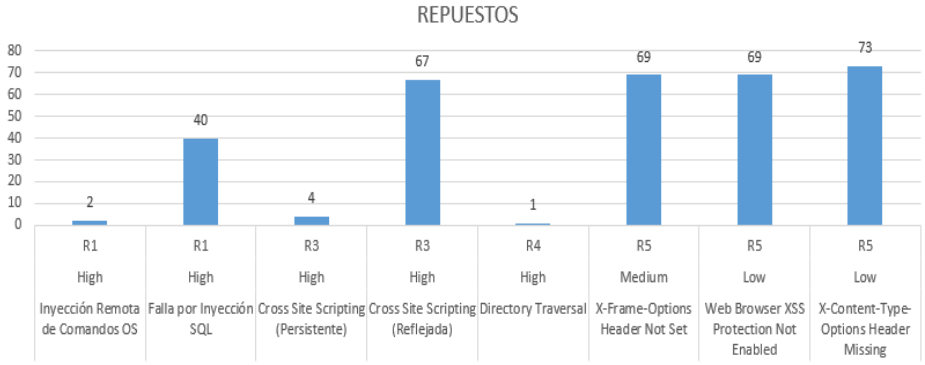


Figura 4.7: Vulnerabilidades encontradas Dpto. de repuestos.

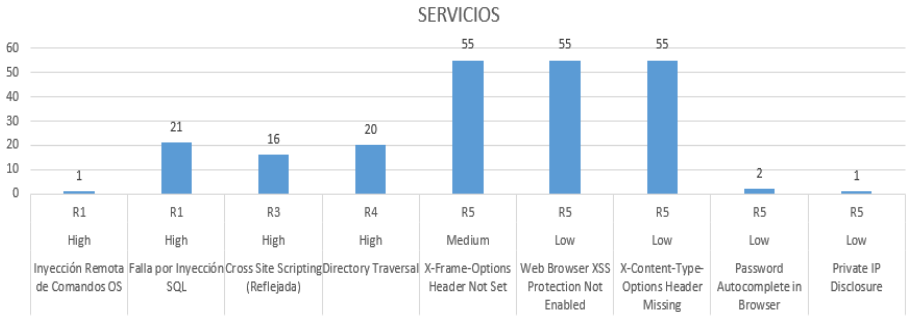


Figura 4.8: Vulnerabilidades encontradas Dpto. de servicios.

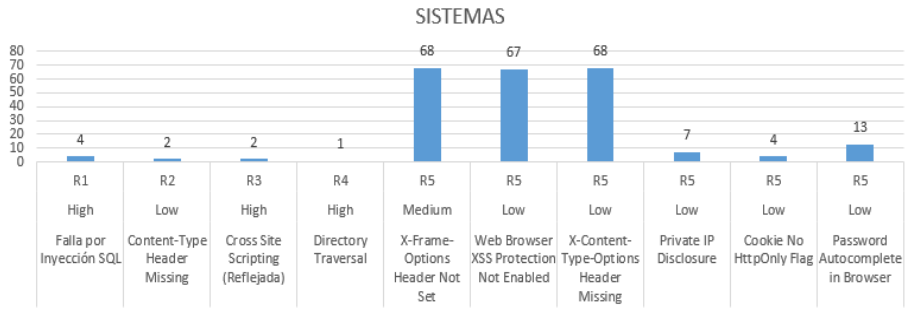


Figura 4.9: Vulnerabilidades encontradas Dpto. de sistemas.

4.2.2. Descripción de vulnerabilidades detectadas en el negocio.

En la sección anterior se presentó información de las vulnerabilidades detectadas en los aplicativos de cada departamento, en ellas se puede constatar que existen vulnerabilidades recurrentes.

A continuación describiremos cada una las vulnerabilidades encontradas.

4.2.2.1. Inyección SQL.

Este ataque consiste en enviar datos no confiables en entradas de un formulario, las mismas que al no ser validadas se envían al interprete (base de datos) como parte de una consulta "válida".

Este tipo de ataque puede comprometer seriamente la información que maneja una organización adulterando y/o divulgando la misma, causando daños económicos y/o mala reputación. Este tipo de ataque es muy común dentro de las aplicaciones web debido a la “confianza” en la que se le atribuye a un usuario en que no registrará datos maliciosos.

4.2.2.2. Inyección remota de comandos OS.

Técnica de ataque usada para la ejecución no autorizada de comandos del sistema operativo. Este ataque es posible cuando una aplicación acepta la entrada que no es de

confianza para construir comandos del sistema operativo de una manera insegura.

4.2.2.3. Content-Type Header Missing.

Se refiere a la pérdida de la cabecera HTTP Content Type, este parámetro indica, a los navegadores web, como interpretar la codificación del documento que se ha descargado del servidor web. Al no definir el Content Type en las páginas web pueden provocar un riesgo de ataque MIME-sniffing.

4.2.2.4. Cross Site Scripting (Reflejada).

Es el tipo de ataque XSS más habitual y consiste en editar los valores que se pasan mediante URL, cambiando el tipo de dato pasado por el usuario a la web, haciendo que el código insertado se ejecute en dicho sitio es decir, el código se ejecutará dentro de la “zona de seguridad” del alojamiento del sitio web.

4.2.2.5. Directory Traversal.

Es una técnica que permite al atacante acceder a los archivos, directorios y comandos que potencialmente residen fuera del directorio raíz de la aplicación web, exponiendo información del equipo donde se aloja el servidor web y su red.

4.2.2.6. X-Frame-Options Header Not Set.

Al no incluir en la respuesta HTTP el parámetro indicado estamos expuestos a posibles ataques de Clickjacking. El clickjacking es una técnica para engañar a los usuarios con el fin de que revelen información confidencial o tomar control de su computador cuando hacen clic en páginas web aparentemente inocentes.

4.2.2.7. Application Error Disclosure.

Indica que la página web contiene un mensaje de error/advertencia en la que se puede revelar información sensible, como por ejemplo: la ubicación del archivo que produjo la excepción no controlada. Esta información puede ser usada para ataques aún mayores.

4.2.2.8. Web Browser XSS Protection Not Enabled.

La protección contra ataques de comandos en sitios cruzados (XSS) no ha sido habilitada.

4.2.2.9. X-Content-Type-Options Header Missing.

Se refiere a la pérdida o no inclusión del parámetro X-Content-Type-Options en la cabecera HTTP para evitar ataques del tipo MIME-sniffing; al no establecer el parámetro como "nosniff" puede ocasionar que la respuesta que interpreta el navegador no se muestre con el tipo de contenido declarado en el servidor web.

4.2.2.10. Password Autocomplete in Browser.

El atributo autocompletar no está desactivado en un elemento HTML input que contiene la entrada de una contraseña. Las contraseñas pueden ser almacenadas en los navegadores y luego se pueden recuperar.

4.2.2.11. Cross-Domain JavaScript Source File Inclusion.

La página incluye uno o más archivos de comandos desde un dominio de terceros.

4.2.2.12. Private IP Disclosure.

Una IP privada como 10.x.x.x, 172.x.x.x, 192.168.x.x se ha encontrado en el cuerpo de la respuesta HTTP. Esta información podría ser útil para otros ataques contra los sistemas internos.

4.2.2.13. Cookie No HttpOnly Flag.

Una cookie se ha fijado sin el indicador HttpOnly, lo que significa que la cookie se puede acceder por JavaScript. Si un script malicioso se puede ejecutar en esta página, entonces la cookie será accesible y puede ser transmitida a otro sitio.

4.3. Diseño de la matriz de riesgos encontrados (análisis de probabilidad).

Después de conocer información de las vulnerabilidades detectadas en los aplicativos web de la compañía, el siguiente paso es determinar si la debilidad encontrada representa un riesgo para la aplicación o su entorno.

La metodología de análisis de riesgos de OWASP consiste en estimar la probabilidad global de ocurrencia y estimar los factores que engloban el impacto técnico y de negocio sobre una vulnerabilidad. Esta información es importante luego para la toma de decisiones en caso de materializarse el riesgo.

Para determinar la **probabilidad de ocurrencia**, la metodología de análisis de riesgos de OWASP, lo divide en 2 grupos: factores relacionados al agente causante de la amenaza y factores que afectan a la vulnerabilidad en sí. Los factores están asociados a un grado de probabilidad entre 0 y 9 que servirán posteriormente para para estimar la probabilidad global (Figura 2.4).

4.3.1. Factores relacionados con el agente causante de la amenaza.

Un agente de amenaza está relacionada a un grupo de atacantes que pueden hacer uso de sus conocimientos para provocar daño, esto mediante el envío de datos maliciosos dentro de la red interna o a través del internet.

Dentro de este grupo se puede considerar algunos agentes de amenazas como: usuarios internos y externos, personal de TI, usuarios con acceso privilegiado, la competencia, ciber-delincuentes, etc., que

pueden hacer uso de fallas en los sistemas para desencadenar riesgos que pueden provocar daño a la institución.

A continuación daremos a conocer los diferentes factores asociados al agente de amenaza con su probabilidad asociada.

- Nivel de conocimiento: Este factor determina cuán grande es el conocimiento técnico que puede tener el grupo de atacantes.

- Motivación: Este factor determina cuán motivante sería, para el grupo de atacantes, explotar la vulnerabilidad hallada y si esta representa beneficio monetario.

- Oportunidad: Este factor pretende determinar qué tan fácil puede ser encontrada y explotada la vulnerabilidad por el grupo de atacantes.

- Tamaño: Este factor mide el tamaño o numerosidad del grupo de atacantes.

Tabla 4: Descripción y pesos de los factores relacionados con el agente causante de la amenaza. [10]

FACTORES AGENTES DE AMENAZAS			
Habilidades Técnicas	**Motivación**	**Oportunidad**	**Tamaño**
Sin conocimiento técnicos **(1)**	Baja motivación/Sin recompensa **(1)**	Acceso total **(1)**	Desarrolladores / Administradores de sistemas **(2)**
Cierto conocimiento técnico **(3)**	Posible recompensa **(4)**	Acceso especial **(4)**	Usuarios de la intranet **(4)**

Usuario avanzado en computación **(5)**	Alta recompensa **(9)**	Acceso parcial **(6)**	Socios **(5)**
Usuario con conocimiento en redes y programación **(6)**		Sin acceso **(9)**	Usuarios autenticados **(6)**
Conocimiento de intrusiones de seguridad **(9)**			Usuarios anónimos de internet **(9)**

4.3.2. Factores que afectan la vulnerabilidad identificada.

Las vulnerabilidades son generalmente fallas que se dan en la aplicación. Estos factores pretenden estimar la probabilidad en que la vulnerabilidad detectada sea encontrada y explotada por los agentes de amenazas.

- Facilidad de descubrimiento: Este factor determina la factibilidad de poder descubrir la vulnerabilidad en la aplicación

- Facilidad de explotación: Este factor determina la factibilidad de poder explotar la vulnerabilidad encontrada.

- Conocimiento: Este factor determina que tan conocida es la vulnerabilidad, si es de conocimiento público o desconocido.

- Detección de intrusos: Este factor mide la frecuencia con la que se detecta un exploit.

Tabla 5: Descripción y pesos de los factores que afectan la vulnerabilidad. [10]

FACTORES DE VULNERABILIDAD			
Facilidad de Descubrimiento	**Facilidad de Explotación**	**Conocimiento**	**Detectores de Intrusión**
Dificultad alta **(1)**	Dificultad alta **(1)**	Desconocido **(1)**	Detección activa **(1)**
Dificultad media **(3)**	Dificultad media **(3)**	Medianamente conocido **(4)**	Autenticado y monitoreado **(3)**
Fácil **(7)**	Sencilla **(5)**	Conocido **(7)**	Autenticado y sin monitoreo **(7)**
Herramientas automatizadas disponibles **(9)**	Herramientas automatizadas disponibles **(9)**	Conocimiento público **(9)**	No autenticado **(9)**

4.3.3. Análisis de probabilidad.

Ahora bien, una vez definido los factores que usaremos para determinar la probabilidad de ocurrencia general de cada vulnerabilidad encontrada, debemos ser capaces de darle un peso según el factor analizado.

El peso otorgado a cada uno de los factores de amenazas y de vulnerabilidad fue consensuado con el departamento de sistemas, específicamente con el jefe de software y de infraestructura. Consideramos que su evaluación es la adecuada debido a que los factores están relacionados a temas tecnológicos, dando lugar a una apreciación más acertada de los pesos de cada factor evaluado. La metodología de riesgos de OWASP ya establece pesos a cada factor, en mucho de los casos se mantuvo el mismo valor pero en otros no.

Una vez conocido los pesos de los factores a analizar procedemos a generar nuestra matriz de riesgo, con el fin de establecer la probabilidad de ocurrencia de cada aplicación web escaneada.

Tanto los factores correspondientes al agente causante de la amenaza como los asociados a la vulnerabilidad se suman y se dividen para la cantidad de factores tomadas en cuenta en el análisis. El valor calculado es considerado para determinar el nivel de probabilidad general, dicho valor puede fluctuar entre bajo, medio y alto.

Para ejemplificar este análisis tomaremos en cuenta las vulnerabilidades analizadas de la aplicación **WEB-APP-R08** del área de repuestos en donde detallaremos los diferentes pesos otorgados a cada factor analizado.

En la siguiente figura se observan los pesos asignados a cada factor que integran el grupo relacionado a los agentes de amenazas, estos valores se suman dando un peso total a cada vulnerabilidad de la aplicación evaluada.

APLICACIONES WEB DEL DEPARTAMENTO DE REPUESTOS [VULNERABILIDAD/CÁLCULO SEVERIDAD DEL RIESGO]	Factores agentes de amenazas				
	Habilidades Técnicas	Motivación	Oportunidad	Tamaño	*TOTAL*
WEB-APP-R08					
Cross Site Scripting (Reflejada)	6	4	4	4	18
Falla por Inyección SQL	6	4	4	4	18
Inyección Remota de Comandos OS	6	1	4	4	15
X-Frame-Options Header Not Set	6	1	4	4	15
Web Browser XSS Protection Not Enabled	6	1	4	4	15
X-Content-Type-Options Header Missing	6	1	4	4	15

Figura 4.10: Análisis de factores agentes de amenazas.

A continuación, en la siguiente figura, se observan los pesos que fueron asignados a cada factor relacionado con la vulnerabilidad en sí, de igual manera estos valores se suman dando un total por cada vulnerabilidad.

APLICACIONES WEB DEL DEPARTAMENTO DE REPUESTOS [VULNERABILIDAD/CÁLCULO SEVERIDAD DEL RIESGO]	PROBABILIDAD				
	Factores de vulnerabilidad				
	Facilidad de Descubrimiento	Facilidad de Explotación	Conciencia o Conocimiento	Detectores de Intrusión	*TOTAL*
WEB-APP-R08					
Cross Site Scripting (Reflejada)	7	5	7	7	26
Falla por Inyección SQL	7	5	7	7	26
Inyección Remota de Comandos OS	7	5	7	7	26
X-Frame-Options Header Not Set	7	5	4	7	23
Web Browser XSS Protection Not Enabled	7	5	4	7	23
X-Content-Type-Options Header Missing	7	5	4	7	23

Figura 4.11: Análisis de factores que afectan la vulnerabilidad.

Para calcular el nivel de probabilidad general debemos tomar en cuenta los valores totales de cada grupo de factores (amenazas y vulnerabilidades), sumamos dichos valores y lo dividimos para la cantidad de factores de cada grupo analizado; este valor fluctúa entre 0 y 9 (probabilidad general) el mismo que se relaciona a un nivel de probabilidad (alto, medio y bajo).

Para entender el cálculo de la “Probabilidad General” tomaremos como ejemplo la vulnerabilidad “Cross Site Scripting” de la aplicación WEB-APP-R08 (figura 4.12).

VULNERABILIDAD	Factores agentes de amenazas				
	Habilidades Técnicas	Motivación	Oportunidad	Tamaño	TOTAL
WEB-APP-R08					
Cross Site Scripting (Reflejada)	6	1	4	4	15

VULNERABILIDAD	Factores de vulnerabilidad				
	Facilidad de Descubrimiento	Facilidad de Explotación	Conciencia o Conocimiento	Detectores de Intrusión	TOTAL
WEB-APP-R08					
Cross Site Scripting (Reflejada)	7	9	9	7	32

Total Factores Agentes Amenazas	Total Factores de vulnerabilidad	Cantidad de factores usados
15	32	8
Probabilidad General (Amenazas+Vulnerabilidad) / cantidad de factores		5.875
Nivel de probabilidad		MEDIUM

Figura 4.12: Cálculo de la probabilidad general. [10]

Después de realizar los cálculos pertinentes pudimos encontrar que todas las vulnerabilidades de la aplicación, tomada como ejemplo, tienen una probabilidad general media.

APLICACIONES WEB DEL DEPARTAMENTO DE REPUESTOS [VULNERABILIDAD/CÁLCULO SEVERIDAD DEL RIESGO]	TOTAL FACTORES AGENTES DE AMENAZAS	TOTAL FACTORES DE VULNERABILIDAD	TOTAL	PROBABILIDAD GENERAL	NIVEL DE PROBABILIDAD
WEB-APP-R08					
Cross Site Scripting (Reflejada)	18	26	44	5.5	MEDIUM
Falla por Inyección SQL	18	26	44	5.5	MEDIUM
Inyección Remota de Comandos OS	15	26	41	5.125	MEDIUM
X-Frame-Options Header Not Set	15	23	38	4.75	MEDIUM
Web Browser XSS Protection Not Enabled	15	23	38	4.75	MEDIUM
X-Content-Type-Options Header Missing	15	23	38	4.75	MEDIUM

Figura 4.13: Probabilidad general del aplicativo web WEB-APP-R08.

4.4. Diseño de la matriz de riesgos encontrados análisis de impacto.

Después del análisis de probabilidad, la metodología de análisis de riesgos de OWASP, manifiesta que se debe analizar el impacto (infraestructura y de negocio) que conllevaría la explotación de la vulnerabilidad encontrada.

Para determinar el impacto global de cada vulnerabilidad encontrada se debe estimar el impacto técnico que está relacionada a la aplicación, a los datos que se manejan y a las funciones que la aplicación proporciona.

También debemos considerar el impacto en el negocio y que es particular de cada institución, pues para una organización una vulnerabilidad puede ser seria pero no para otra por eso, es primordial mitigar los riesgos más importantes para las aplicaciones del negocio.

4.4.1. Factores para estimar el impacto técnico.

Los factores relacionados con el impacto técnico están alineados con los pilares de la seguridad informática (la confidencialidad, la integridad y la disponibilidad) lo cual permite determinar la dimensión del impacto en las aplicaciones si la vulnerabilidad es explotada.

A continuación se mencionan algunos factores para estimar el impacto técnico:

- Pérdida de la confidencialidad: Este factor establece una medida para saber si la cantidad y la sensibilidad de la información, que se muestra en la aplicación, podría ser revelada y cuán delicada es.

- Pérdida de la integridad: Este factor mide el nivel de daño que podría sufrir la información que se muestra en la aplicación. Se considera el tipo de información perdida o dañada.

- Pérdida de disponibilidad: Este factor mide la disponibilidad de la aplicación en caso de que la vulnerabilidad sea explotada, el servicio se mantiene o se detiene por completo.

- Pérdida de auditabilidad: Este factor mide la posibilidad en que un ataque sea rastreado hasta llegar con el causante del daño.

Tabla 6: Descripción y pesos de los factores para estimar el impacto técnico. [10]

Factores de impacto técnico			
Pérdida de la Confidencialidad	**Pérdida de la Integridad**	**Pérdida de la Disponibilidad**	**Pérdida de la Auditabilidad**
Mínima (data no crítica) **(1)**	Mínima (data no crítica) **(1)**	Mínima (Servicios no críticos) **(1)**	Totalmente auditable **(1)**
Mínima (data crítica) **(3)**	Mínima (data crítica) **(3)**	Mínima (Servicios críticos) **(3)**	Posiblemente auditable **(7)**
Considerable (data crítica) **(6)**	Considerable (data no crítica) **(5)**	Considerable (Servicios no críticos) **(5)**	No auditable **(9)**
Considerable (data no crítica) **(7)**	Considerable (data crítica) **(7)**	Considerable (Servicios críticos) **(7)**	
Corrupción de datos total **(9)**	Corrupción de datos total **(9)**	Pérdida total del servicio **(9)**	

4.4.2. Factores para estimar el impacto en el negocio.

Estimar el impacto en el negocio puede llegar a ser muy complejo porque depende de la percepción de cada organización, por esto es importante conocer la apreciación de lo que realmente es importante para el negocio.

Conocer el riesgo sobre el negocio determinará o justificará la inversión realizada para solucionar los inconvenientes de seguridad que se pueden presentar en las tecnologías de información.

A continuación se presenta los factores que nos permitirá estimar el impacto en el negocio:

- Daño financiero: Este factor mide el daño monetario que podría provocar la explotación de la vulnerabilidad analizada.
- Daño de imagen: Este factor pretende medir el daño a la reputación de la compañía, si la vulnerabilidad es explotada.
- Incumplimiento: Exposición introduce la no conformidad
- Violación de la privacidad: Mide la cantidad de individuos afectados por la explotación de la vulnerabilidad.

Tabla 7: Descripción y pesos de los factores para estimar el impacto en el negocio. [10]

Factores de impacto en el negocio			
Daño Económico	**Daño de Imagen**	**Incumplimiento**	**Violación a la Privacidad**
Menor al costo de arreglar la vulnerabilidad **(1)**	Daño mínimo **(1)**	Mínimo **(2)**	Una persona **(1)**
Leve efecto en el beneficio anual **(3)**	Pérdida de cuentas principales **(3)**	Medio **(5)**	Cientos de personas **(5)**
Efecto significativo en el beneficio anual **(7)**	Pérdida de credibilidad a gran escala **(6)**	Alto **(8)**	Miles de personas **(7)**
Bancarrota **(9)**	Daño total de la imagen **(9)**		Millones de personas **(9)**

4.4.3. Análisis de impacto.

Una vez determinado los factores que usaremos para determinar el análisis de impacto, debemos ser capaces de valorar cada una de las vulnerabilidades encontradas por cada aplicación escaneada.

Los pesos a los factores de impacto técnico fueron revisados por el departamento de sistemas, colaborando el jefe del área técnica y de software; se dedujo que la evaluación realizada por ellos permitirá tener un mejor resultado debido a que los factores están dirigidos a temas tecnológicos.

En cambio, para definir los factores de impacto en el negocio se tuvo que establecer reuniones con los jefes de cada departamento y de

sistemas, además contamos con la ayuda del jefe financiero. Consideramos oportuna la intervención de cada uno de los integrantes porque permitieron definir los pesos a los factores establecidos, ofreciendo un criterio adecuado a las implicaciones de negocio que conllevaría en el caso de presentarse las vulnerabilidades encontradas en los aplicativos.

Una vez conocido los pesos de los factores a analizar procedemos a generar nuestra matriz de riesgo para determinar el impacto global de la vulnerabilidad en caso de que se explote dicha falla.

Tanto los factores correspondientes al impacto técnico como aquellos asociados al impacto del negocio se suman y luego, se dividen para la cantidad de factores tomadas en cuenta en el análisis. El valor calculado determinará el nivel de probabilidad de impacto, dicho valor puede fluctuar entre bajo, medio y alto.

Para demostrar este análisis tomaremos en cuenta las vulnerabilidades de la aplicación **WEB-APP-R08** del área de repuestos antes estudiada en el análisis de probabilidad de ocurrencia.

En la siguiente figura, se observa los pesos asignados a cada vulnerabilidad detectada según el factor evaluado, estos valores se suman dando un peso total a cada vulnerabilidad.

APLICACIONES WEB DEL DEPARTAMENTO DE REPUESTOS [VULNERABILIDAD/CÁLCULO SEVERIDAD DEL RIESGO]	Factores de impacto técnico				
	Pérdida de la Confidencialidad	Pérdida de la Integridad	Pérdida de la Disponibilidad	Pérdida de la Auditibilidad	TOTAL
WEB-APP-R08					
Cross Site Scripting	7	5	1	7	20
Falla por Inyección SQL	7	5	1	7	20
Inyección Remota de Comandos OS	7	5	1	7	20
X-Frame-Options Header Not Set	3	1	1	7	12
Web Browser XSS Protection Not Enabled	3	1	1	7	12
X-Content-Type-Options Header Missing	3	1	1	7	12

Figura 4.14: Análisis factores de impacto técnico.

A continuación, la siguiente figura, demuestra los pesos asignados a cada vulnerabilidad detectada según el factor evaluado, al igual que el caso anterior los pesos de cada vulnerabilidad se suman dando un total para el grupo de factores relacionados con el impacto en el negocio.

APLICACIONES WEB DEL DEPARTAMENTO DE REPUESTOS [VULNERABILIDAD/CÁLCULO SEVERIDAD DEL RIESGO]	IMPACTO				
	Factores de impacto del negocio				
	Daño Económico	Daño de Imagen	Incumplimiento	Violación a la Privacidad	TOTAL
WEB-APP-R08					
Cross Site Scripting	1	1	5	3	10
Falla por Inyección SQL	1	1	5	3	10
Inyección Remota de Comandos OS	1	1	5	3	10
X-Frame-Options Header Not Set	1	1	2	3	7
Web Browser XSS Protection Not Enabled	1	1	2	3	7
X-Content-Type-Options Header Missing	1	1	2	3	7

Figura 4.15: Análisis de factores de impacto en el negocio.

Para calcular el nivel de impacto global, debemos tomar en cuenta las los valores totales de cada grupo (factores de impacto técnico y de negocio), sumamos los valores y lo dividimos para la cantidad de factores de cada grupo evaluado, el valor obtenido se asocia a un nivel de probabilidad.

Para entender el cálculo de la “Probabilidad General” tomaremos como ejemplo la vulnerabilidad “Cross Site Scripting” de la aplicación WEB-APP-R08, ver figura 4.16.

VULNERABILIDAD	Factor de impacto en el negocio				
	Pérdida de la Confidencialidad	Pérdida de la Integridad	Pérdida de la Disponibilidad	Pérdida de la Auditibilidad	*TOTAL*
WEB-APP-R08					
Cross Site Scripting (Reflejada)	7	5	1	7	20

VULNERABILIDAD	Factor de impacto en el negocio				
	Daño Económico	Daño de Imagen	Incumplimiento	Violación a la Privacidad	*TOTAL*
WEB-APP-R08					
Cross Site Scripting (Reflejada)	1	1	5	3	10

Total Factores impacto técnico	Total Factores impacto en el negocio	Cantidad de factores usados
20	10	8
Impacto Global (Impacto Técnico + Impacto Negocio) / cantidad de factores		3.75
Nivel de probabilidad		MEDIUM

Figura 4.16: Cálculo del impacto global. [10]

Ahora bien, después de realizado los cálculos pudimos encontrar que las vulnerabilidades asociadas a los riesgos de inyección (A1) y secuencia de comandos cruzados (A3) están valoradas con un nivel de impacto global medio y las vulnerabilidades asociadas a la configuración de seguridad incorrectas (A5) están valoradas con un nivel de impacto global bajo, según el apartado OWASP Top 10 2013.

APLICACIONES WEB DEL DEPARTAMENTO DE REPUESTOS [VULNERABILIDAD/CÁLCULO SEVERIDAD DEL RIESGO]	TOTAL FACTORES TÉCNICOS	TOTAL FACTORES DE IMPACTO DEL NEGOCIO	TOTAL	IMPACTO GLOBAL	NIVEL DE IMPACTO
WEB-APP-R08					
Cross Site Scripting	20	10	30	3.75	MEDIUM
Falla por Inyección SQL	20	10	30	3.75	MEDIUM
Inyección Remota de Comandos OS	20	10	30	3.75	MEDIUM
X-Frame-Options Header Not Set	12	7	19	2.375	LOW
Web Browser XSS Protection Not Enabled	12	7	19	2.375	LOW
X-Content-Type-Options Header Missing	12	7	19	2.375	LOW

Figura 4.17: Análisis de impacto de cada vulnerabilidad de la aplicación WEB-APP-R08.

4.4.4. Severidad del riesgo.

Una vez obtenido los niveles de probabilidad de ocurrencia general y de impacto global, es necesario identificar la severidad del riesgo asociado a cada una de las vulnerabilidades detectadas. Esta identificación nos permitirá discernir que riesgos se deben mitigar con mayor urgencia y cuales se deben rezagar por el momento.

Para determinar la severidad del riesgo usaremos la técnica indicada por la metodología de análisis de riesgo de OWASP (figura 2.4) la misma que nos indica, mediante una matriz, las coordenadas para hallar la severidad del riesgo asociado según sus niveles de probabilidad y de impacto global.

Como ejemplo usaremos la aplicación WEB-APP-R08 y determinaremos la severidad del riesgo asociada a cada vulnerabilidad.

Tabla 8: Determinación de severidad del riesgo aplicativo WEB-APP-R08.

WEB-APP-R08 Vulnerabilidades	Nivel de Probabilidad	Nivel de Impacto	Severidad del Riesgo
Cross Site Scripting (Reflejada)	Medio	Medio	Medio
Falla por Inyección SQL	Medio	Medio	Medio
Inyección Remota de Comandos OS	Medio	Medio	Medio
X-Frame-Options Header Not Set	Medio	Bajo	Bajo
Web Browser XSS Protection Not Enabled	Medio	Bajo	Bajo
X-Content-Type-Options Header Missing	Medio	Bajo	Bajo

En la tabla anterior se puede observar que las vulnerabilidades de Cross Site Scripting e Inyección SQL son consideradas, para esta aplicación, con criticidad media pero las vulnerabilidades relacionadas al mal uso de configuraciones son catalogadas con severidad baja.

Esta información es relevante a la hora de tomar de decisiones, porque los esfuerzos de mitigación estarán dirigidos inicialmente en remediar aquellos riesgos con nivel de severidad media, mientras que las bajas se pueden considerar o no en el plan de remediación a corto o mediano plazo.

A continuación se presentan tablas con información de las aplicaciones que fueron calificadas con un nivel de severidad alto o medio, divididas por departamento:

Tabla 9: Severidad del riesgo para las aplicaciones web del departamento de contraloría

Vulnerabilidades	Probabilidad General	Impacto Global	Severidad del riesgo
WEB-APP-C02			
Cross Site Scripting (Reflejada)	MEDIO (5.12)	MEDIO (4.00)	MEDIO
WEB-APP-C03			
Directory Traversal	MEDIO (5.12)	MEDIO (4.00)	MEDIO
Cross Site Scripting (Reflejada)	MEDIO (5.12)	MEDIO (4.00)	MEDIO
Application Error Disclosure	MEDIO (5.12)	MEDIO (4.00)	MEDIO
WEB-APP-C04			
Cross Site Scripting (Reflejada)	MEDIO (5.12)	MEDIO (4.00)	MEDIO
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (4.00)	MEDIO
Application Error Disclosure	MEDIO (5.12)	MEDIO (4.00)	MEDIO
WEB-APP-C05			
Inyección Remota OS	MEDIO (5.12)	MEDIO (3.75)	MEDIO
WEB-APP-C06			
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (3.75)	MEDIO
WEB-APP-C07			
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (4.00)	MEDIO
WEB-APP-C08			
Cross Site Scripting (Reflejada)	MEDIO (5.12)	MEDIO (4.00)	MEDIO
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (4.00)	MEDIO
WEB-APP-C09			
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (3.75)	MEDIO
WEB-APP-C10			
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (4.00)	MEDIO
WEB-APP-C11			
Directory Traversal	MEDIO (5.12)	MEDIO (3.75)	MEDIO
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (3.75)	MEDIO
Application Error Disclosure	MEDIO (5.12)	MEDIO (3.37)	MEDIO
WEB-APP-C12			
Cross Site Scripting (Reflejada)	MEDIO (5.50)	MEDIO (4.62)	MEDIO
Falla por Inyección SQL	MEDIO (5.50)	MEDIO (4.87)	MEDIO

Application Error Disclosure	MEDIO (5.12)	MEDIO (4.62)	MEDIO
WEB-APP-C13			
Cross Site Scripting (Reflejada)	MEDIO (5.12)	MEDIO (4.25)	MEDIO
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (4.25)	MEDIO
WEB-APP-C17			
Inyección Remota de Comandos OS	MEDIO (5.12)	MEDIO (3.75)	MEDIO

Tabla 10: Severidad del riesgo para las aplicaciones web del departamento de crédito y cobranzas.

Vulnerabilidades	Probabilidad General	Impacto Global	Severidad del riesgo
WEB-APP-CC1			
Cross Site Scripting (Reflejada)	MEDIO (5.12)	MEDIO (4.12)	MEDIO
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (4.12)	MEDIO
WEB-APP-CC2			
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (4.12)	MEDIO

Tabla 11: Severidad del riesgo para las aplicaciones web del departamento de maquinaria.

Vulnerabilidades	Probabilidad General	Impacto Global	Severidad del riesgo
WEB-APP-M02			
Falla por Inyección SQL	MEDIO (5.50)	MEDIO (4.75)	MEDIO
WEB-APP-M03			
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (3.50)	MEDIO
WEB-APP-M04			
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (3.50)	MEDIO
WEB-APP-M05			
Cross Site Scripting (Reflejada)	MEDIO (5.12)	MEDIO (3.50)	MEDIO
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (3.50)	MEDIO

Tabla 12: Severidad del riesgo para las aplicaciones web del departamento de repuestos.

Vulnerabilidades	Probabilidad General	Impacto Global	Severidad del riesgo
WEB-APP-R02			
Cross Site Scripting (Reflejada)	MEDIO (5.50)	MEDIO (4.50)	MEDIO
Falla por Inyección SQL	MEDIO (5.50)	MEDIO (4.50)	MEDIO
WEB-APP-R03			
Cross Site Scripting (Reflejada)	MEDIO (5.50)	MEDIO (4.12)	MEDIO
Falla por Inyección SQL	MEDIO (5.50)	MEDIO (4.12)	MEDIO
WEB-APP-R04			
Cross Site Scripting (Reflejada)	MEDIO (5.50)	MEDIO (4.37)	MEDIO
Falla por Inyección SQL	MEDIO (5.50)	MEDIO (4.37)	MEDIO
WEB-APP-R05			
Cross Site Scripting (Reflejada)	MEDIO (5.50)	MEDIO (3.75)	MEDIO
Falla por Inyección SQL	MEDIO (5.50)	MEDIO (3.75)	MEDIO
WEB-APP-R06			
Falla por Inyección SQL	MEDIO (5.50)	MEDIO (3.75)	MEDIO
WEB-APP-R08			
Cross Site Scripting (Reflejada)	MEDIO (5.50)	MEDIO (3.75)	MEDIO
Falla por Inyección SQL	MEDIO (5.50)	MEDIO (3.75)	MEDIO
Inyección Remota de Comandos OS	MEDIO (5.12)	MEDIO (3.75)	MEDIO

Tabla 13: Severidad del riesgo para las aplicaciones web del departamento de servicios.

Vulnerabilidades	Probabilidad General	Impacto Global	Severidad del riesgo
WEB-APP-S02			
Cross Site Scripting (Reflejada)	MEDIO (5.12)	MEDIO (3.75)	MEDIO
WEB-APP-S03			
Cross Site Scripting (Reflejada)	MEDIO (5.12)	MEDIO (4.62)	MEDIO

Falla por Inyección SQL	MEDIO (5.12)	MEDIO (4.62)	MEDIO

Tabla 14: Severidad del riesgo para las aplicaciones web del departamento de sistemas.

Vulnerabilidades	Probabilidad General	Impacto Global	Severidad del riesgo
WEB-APP-I03			
Falla por Inyección SQL	MEDIO (5.12)	MEDIO (3.75)	MEDIO
WEB-APP-I04			
Directory Traversal	MEDIO (5.50)	MEDIO (4.75)	MEDIO
Cross Site Scripting	MEDIO (5.50)	MEDIO (4.75)	MEDIO
Falla por Inyección SQL	MEDIO (5.50)	MEDIO (4.75)	MEDIO
WEB-APP-I05			
Cross Site Scripting (Reflejada)	MEDIO (5.50)	MEDIO (4.87)	MEDIO
X-Frame-Options Header Not Set	MEDIO (5.12)	MEDIO (4.25)	MEDIO
Web Browser XSS Protection Not Enabled	MEDIO (4.75)	MEDIO (4.25)	MEDIO
X-Content-Type-Options Header Missing	MEDIO (4.75)	MEDIO (4.25)	MEDIO

En los anexos nombrados a continuación se presentan las matrices de riesgos generadas de las aplicaciones seleccionadas por departamento:

- Anexo A - Matriz de riesgos de las aplicaciones del departamento de Contraloría.
- Anexo B - Matriz de riesgos de las aplicaciones del departamento de Crédito y Cobranzas.

- Anexo C - Matriz de riesgos de las aplicaciones del departamento de Maquinaria.
- Anexo D - Matriz de riesgos de las aplicaciones del departamento de Repuestos.
- Anexo E - Matriz de riesgos de las aplicaciones del departamento de Servicios.
- Anexo F - Matriz de riesgos de las aplicaciones del departamento de Sistemas.

4.5. Diseño de un esquema de seguridad de las aplicaciones web a implementar en la compañía.

4.5.1. Priorizar planes de acción.

En la sección anterior pudimos establecer la severidad del riesgo asociada a cada vulnerabilidad ahora el siguiente paso, es definir las vulnerabilidades que serán tomadas en cuenta para el plan de mitigación de riesgos a implantarse en la compañía.

Es importante conocer las acciones que llevaremos a cabo para enfrentar el riesgo encontrado, de manera que podamos darle un tratamiento efectivo (prevenir, reducir, compartir y/o aceptar el riesgo).

Se coordinó con la gerencia del departamento de sistemas que las vulnerabilidades con severidades altas y medias serán tomadas en cuenta para el plan de mitigación del presente trabajo, las mismas que

tendrán un nivel de prioridad según las aplicaciones que ellos consideren importantes y de mayor uso en el negocio. Aquellas vulnerabilidades con severidad baja se describirán como mejoras pendientes a realizar.

Las vulnerabilidades con severidades altas deben ser tomadas en cuenta en primera instancia pues estas reducen significativamente el impacto general de la vulnerabilidad sobre la aplicación. Debemos considerar que ciertos riesgos altos, medios o bajos representarán para la compañía algún tipo de costo monetario, la misma que debe ser aprobada por la gerencia, los cuales aceptarán o no dicho costo.

4.5.2. Diseño del esquema de seguridad.

Llevado a cabo el análisis de riesgos debemos generar un esquema que mitigue las necesidades de seguridad encontradas, con el fin de que este mismo esquema sea usado por las demás aplicaciones de la compañía; actuales o futuras.

El entorno de seguridad de las aplicaciones web la conforman diferentes áreas como son: 1) seguridad en el cliente, 2) seguridad en el servidor, 3) seguridad en la aplicación y 4) seguridad en la comunicación.

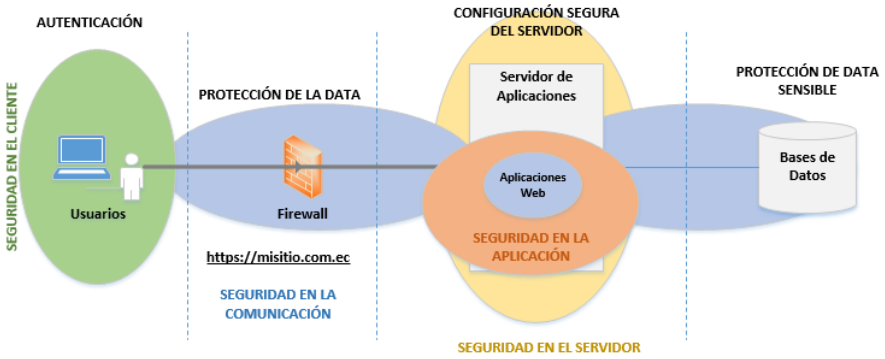


Figura 4.18: Esquema general de seguridad de las aplicaciones web.

Al emplear procedimientos seguros, definir responsables y establecer planes de seguridad eficientes, se puede afianzar que el entorno de seguridad será menos vulnerable a ataques internos o externos.

A continuación se describe cada una de las áreas que conforman el entorno seguro de las aplicaciones web, con información que debe llevarse a cabo en la compañía para encontrar dicha armonía:

- La seguridad en el cliente se refiere al entorno sobre el cual una aplicación web es solicitada esperando una respuesta a su petición. En este entorno se debe considerar la actualización permanente de sistemas operativos, antivirus, navegadores web, parches de seguridad, plugins, etc. y el empleo adecuado de lenguajes y/o programas que se ejecutan del lado del cliente (JavaScript, Applets Java, controles ActiveX, macros, etc.) brindando así un nivel de seguridad más efectivo ante posibles ataques a la aplicación web y al equipo del cliente.

Tabla 15: Actividades para la seguridad en el cliente.

Seguridad en el cliente		
Actividades	**Responsables**	**Revisión**
Actualización de sistema operativo	Técnicos del área de TI	Mensual
Actualización de antivirus	Técnicos del área de TI / Usuarios	Semanal
Escaneo de virus (programada)	Técnicos del área de TI	Semanal
Actualización de navegadores web	Técnicos del área de TI / Usuarios	Mensual
Respaldo de información	Usuarios	Diaria

- La seguridad en el servidor se refiere al entorno en donde reside la aplicación web, la base de datos, archivos, etc. Dicho entorno debe considerar seguridades físicas (acceso restringido al equipo físico, gestión de respaldos, DMZ, bitácora de novedades, etc.) y lógicas (gestión de accesos, auditoría y monitoreo de Logs, antivirus, ACLs, Hardening, evitar configuraciones por defecto, etc.), con esto lograremos mitigar las principales deficiencias de seguridad en dichos equipos.

Tabla 16: Actividades para la seguridad en el servidor.

Seguridad en el servidor		
Actividades	**Responsables**	**Revisión**
Respaldos de información de base de datos	Técnicos del área de TI / Analistas de sistemas	Diaria
Respaldos de información de servidor de archivos	Técnicos del área de TI / Analistas de sistemas	Diaria
Respaldos de información de servidor de aplicaciones	Técnicos del área de TI / Analistas de sistemas	Diaria
Respaldos de información de configuraciones	Técnicos del área de TI	Semanal

Revisión de Logs de servidores	Analistas de sistemas	Diaria
Revisión de configuraciones	Técnicos del área de TI	Mensual
Actualización de sistema operativo	Técnicos del área de TI	Mensual
Actualización de parches de seguridad	Técnicos del área de TI	Quincenal
Actualización de antivirus	Técnicos del área de TI	Semanal
Bitácora de novedades	Operadores / Jefe de TI	Diaria

- La seguridad en la aplicación es uno de los más aspectos importantes pero a la vez el más descuidado por parte de los administradores de sistemas, pues se piensa que la aplicación solo debe funcionar y no se mide o regula la seguridad de la misma en las diferentes etapas del ciclo de desarrollo de un software. Debemos considerar el control de acceso a las diferentes opciones de la aplicación (autenticación y autorización), validar los datos en las diferentes entradas de la aplicación es decir, no confiar en las entradas que realice cualquier usuario para evitar los diferentes riesgos que se mencionan en el OWASP Top 10 2013 y por último programar los sistemas de manera segura protegiendo la información sensible y gestionar los errores.

Tabla 17: Actividades para la seguridad en la aplicación.

Seguridad en la aplicación		
Actividades	**Responsables**	**Revisión**
Control de cambios en programas	Analistas de sistemas	Semanal
Documentación de cambios en programas	Analistas de sistemas / Jefe de Software	Diaria
Detección de vulnerabilidades	Jefe de Software	Semanal
Codificación segura	Analistas de sistemas / Jefe de Software	Semanal
Monitoreo de servidor de aplicaciones	Analistas de sistemas	Diaria
Monitoreo de servidor de base de datos	Analistas de sistemas	Diaria

- La seguridad en la comunicación se refiere a la seguridad establecida en el canal de comunicación esto con la finalidad de que los datos recibidos y enviados no sean legibles a un atacante y así pueda mantener la información segura frente a terceros. Generalmente se emplea el protocolo de comunicación segura SSL.

Tabla 18: Actividades para la seguridad en la canal.

Seguridad en el canal		
Actividades	**Responsables**	**Revisión**
Monitoreo del estado de la red	Jefe de TI	Diaria
Monitoreo detección de intrusos	Jefe de TI	Diaria
Monitoreo de VPN	Técnicos del área de TI	Diaria
Monitoreo de equipos de computación	Técnicos del área de TI	Diaria

Una vez mencionado los diferentes aspectos que abarca la seguridad de las aplicaciones web, podemos mencionar que el esquema de seguridad inicial a implantarse en la compañía IIASA Ecuador, está enfocada a ***la seguridad de la aplicación y la seguridad en el canal***, debido a los requerimientos solicitados por el departamento de sistemas, por ello este esquema abarcará dichos aspectos esencialmente.

La seguridad de la aplicación se llevará a cabo mediante la codificación correcta de las vulnerabilidades encontradas cuyo análisis de riesgo es considerado medio o alto y la implementación del protocolo SSL para la comunicación de la plataforma intranet que contiene las aplicaciones evaluadas con anterioridad.

CAPÍTULO 5.

5. IMPLEMENTACIÓN DE SOLUCIONES A LOS RIESGOS ANALIZADOS CON MAYOR IMPACTO EN EL NEGOCIO.

En este capítulo revisaremos los diferentes pasos que llevamos a cabo para implementar el esquema de seguridad propuesto con el propósito de mitigar la mayor cantidad de riesgos encontrados en las aplicaciones web analizadas previamente.

Se emplearon las medidas de seguridad otorgadas por el apartado OWASP para mitigar los riesgos con mayor severidad, determinamos mediante un análisis de costo/beneficio que la implementación de este esquema debe llevarse a cabo, adecuamos los diferentes controles de seguridad proporcionados por la herramienta ESAPI Java para las aplicaciones de la compañía y por último damos

a conocer las modificaciones realizadas en los fuentes de los aplicativos web para mitigar los riesgos encontrados.

5.1. Plan de implementación de las soluciones sugeridas por el apartado OWASP Top 10 2013 para aquellos riesgos con mayor severidad en el negocio.

Conocida la matriz de riesgos y el esquema de seguridad propuesto (seguridad en la aplicación y en la comunicación), el siguiente paso es establecer un plan que se ajuste a las necesidades de todas las aplicaciones web analizadas, a fin de disminuir la mayor cantidad de riesgos de seguridad encontrados.

El objetivo de este plan es que todo desarrollo interno o externo use el esquema de seguridad propuesto con la finalidad de evitar en lo posible vulnerabilidades asociadas con riesgos ya documentados.

Para llevar a cabo el plan de implementación de mejoras de seguridad en cada uno de los aplicativos web seguimos los siguientes pasos:

1) De todas las aplicaciones web analizadas se tomarán en cuenta, para la mitigación de riesgos, solo aquellos con severidades de riesgo media o alta (tablas 9 - 14),

2) De las aplicaciones web con severidad de riesgo medio o alto colocamos entre los primeros lugares las aplicaciones consideradas importantes para el negocio (empleados y/o clientes) es decir, fueron las primeras en ser mitigadas.

3) Tomamos en cuenta las soluciones sugeridas por el apartado OWASP Top 10 2013,

4) Evaluamos el costo/beneficio de implementar las soluciones de seguridad a las aplicaciones web elegidas, dando lugar a que la gerencia tome la decisión de implementar o no ciertos controles de seguridad propuestos en los aplicativos,

5) Familiarizarnos con los diferentes controles que provee la herramienta ESAPI Java y adaptarlo a las necesidades de seguridad con las que carecen los aplicativos evaluados de la compañía.

6) Y como último paso, realizamos las diferentes codificaciones y configuraciones de seguridad sugeridas por el apartado OWASP Top 10 2013, empleando la herramienta ESAPI java.

5.2. Categorizar las soluciones según necesidades indicadas y el impacto en el negocio.

La matriz de riesgo elaborada a partir de los datos recabados por el escáner de vulnerabilidades y las calificaciones otorgadas por personal de cada departamento, permitió definir los riesgos que se pueden presentar en una o varias aplicaciones de la intranet.

Entre las vulnerabilidades que podemos mencionar y cuyo riesgo puede ocasionar daños a las aplicaciones y a la plataforma en sí, son: “Inyección SQL”, “Inyección remota de comandos OS”, “Cross Site Scripting (reflejada)”, “Directory Traversal” y “Application Error Disclosure”.

Estas vulnerabilidades, por su severidad de riesgo en la compañía, son tratadas en primera instancia por el plan de implementación, cumpliendo con uno de los requisitos planteados por el departamento de sistemas.

Las soluciones implementadas para estas aplicaciones están categorizadas de acuerdo al apartado OWASP Top 10 2013, esto quiere decir que la planificación de soluciones estará orientada según el ranking del riesgo asociado a la vulnerabilidad, tomando como prioridad las aplicaciones consideradas importantes para la institución en el caso de que estas presenten riesgos para la plataforma.

Por disposición del departamento de sistemas las implementaciones requeridas en el presente estudio se llevarán a cabo en ambiente de prueba.

Dicho entorno cuenta con las mismas características que el ambiente de producción excepto en los datos.

En la siguiente tabla se describen las soluciones sugeridas por el apartado OWASP Top 10 2013, la misma que proporciona información útil para la mitigación del riesgo asociado a las vulnerabilidades antes mencionadas:

Tabla 19: Categorización de soluciones a riesgos de seguridad presentados en las aplicaciones web.

Vulnerabilidad	Ranking OWASP Top 10 2013	Severidad de Riesgo
Inyección SQL	A1 - Inyección	MEDIO
Remediación * No concatenar cadenas de consultas * Parametrizar consultas a la base de datos * Uso de procedimientos almacenados * No confiar en la entrada de datos por parte del usuario		
Inyección Remota de Comandos OS	A1 - Inyección	MEDIO
Remediación * Uso de listas blancas para la entrada de datos (verificar tamaño, tipo de dato, sintaxis) * Limitar los privilegios de acceso a archivos de configuración sobre el sistema operativo		
Cross Site Scripting (Reflejada)	A3 – XSS	MEDIO
Remediación * Codificar los caracteres introducidos por el usuario * Uso de listas blancas para la entrada de datos (verificar tamaño, tipo de dato, sintaxis)		
Directory Traversal	A4 - Referencia directa insegura de objetos	MEDIO
Remediación * Uso de listas blancas para la entrada de datos (verificar tamaño, tipo de dato, sintaxis) * Subida, descarga o acceso a archivos solo al repositorio permitido * Limitar los privilegios de acceso a archivos de configuración		
Application Error Disclosure	A5 - Configuración de seguridad incorrecta	MEDIO
Remediación * Direccionar errores a páginas personalizadas * Manejo de errores y mostrarlos solo en el Log del servidor * Evitar mostrar información detallada del error al cliente		

5.3. Evaluación de tiempo, costo y beneficio al implementar las soluciones a los riesgos de seguridad en las aplicaciones web analizadas.

Hoy en día los presupuestos de seguridad se encuentran dirigidos, en su mayoría, a la seguridad perimetral y en ciertos casos al control de amenazas de código malicioso, pero estas medidas no garantizan que las organizaciones no sufran de ataques de terceros, claro está que depende mucho del beneficio que el atacante recibirá a cambio de la información obtenida. Debemos considerar que las aplicaciones, por muy pequeña que esta sea, debe contar con lineamientos de seguridad en cada etapa del ciclo de vida del programa, para mantener no solo un software de calidad sino que además seguro y confiable.

Una vez analizada las vulnerabilidades de las aplicaciones web de la compañía IIASA, evaluamos los costos de implementación de las soluciones para corregir los fallos de seguridad encontrados, determinar si existen beneficios en realizar las correcciones y el tiempo que implicaría implementar dichas soluciones.

5.3.1. Tiempo de implementación de soluciones.

El análisis de riesgo arrojó como resultado que del 100% de aplicaciones analizadas, el 60% de estas tienen vulnerabilidades cuyo riesgo puede materializarse y afectar el entorno sobre el cual se despliega la intranet de la compañía. Estas vulnerabilidades van desde la filtración y pérdida de información, el acceso a archivos y/o directorios sensibles del servidor de aplicaciones hasta el manejo

incorrecto de errores que se dan en la aplicación web y que son visibles para el usuario.

El riesgo debe ser prevenido, reducido, compartido o aceptado dependerá de la factibilidad de la solución en cada aplicativo. Además de decidir la forma en cómo tratar el riesgo, también contaremos con el número de incidencias o instancias que una misma vulnerabilidad se repite en la aplicación, lo cual implica un mayor tiempo en la implementación de soluciones.

El tiempo de implementación de las soluciones para mitigar los riesgos con mayor impacto encontrados en las aplicaciones web analizadas de la compañía en ambiente de prueba es de 40 días. Se incluye 15 días adicionales para la investigación e implementación del canal de cifrado SSL sobre el ambiente de prueba y dejar todo listo para su paso a producción en caso de que la gerencia tome la decisión, lo mismo sucederá con las aplicaciones tomadas en cuenta.

5.3.2. Análisis de costo/beneficio.

El análisis de riesgo desveló que todos los aplicativos contaban con al menos una vulnerabilidad, las mismas que fueron categorizadas de acuerdo a su nivel de riesgo. Ahora bien, para determinar en dólares la inversión de la compañía en solucionar los fallos de seguridad de sus aplicaciones, cuantificaremos el costo de implementar las soluciones versus los beneficios que recibirá la compañía.

Al llevar a cabo este análisis la gerencia del departamento de sistemas y los ejecutivos de la compañía discernirán, de mejor manera, los peligros con los que cuenta la plataforma de intranet; tomando decisiones más acertadas y distribuyendo de mejor manera los recursos en pro de mejorar las aplicaciones analizadas y el resto de aplicaciones web de la compañía.

De entre las vulnerabilidades detectadas podemos considerar que las relacionadas con inyección SQL y Cross Site Scripting (XSS) son las más importantes y que pueden poner en peligro los servicios ofrecidos por la institución, cabe señalar que estas vulnerabilidades permiten al atacante adueñarse y/o alterar la información con fines de perjudicar el negocio u obtener beneficio económico a cambio.

Actualmente la compañía dependen de los servicios de TI que se encuentran alojados en sus instalaciones o en la nube, esta dependencia provoca que vulnerabilidades, como las detectadas, provoquen que los servicios dejen de funcionar dando lugar a pérdidas económicas, mala reputación y falta de credibilidad.

Diariamente la compañía tiene ventas por alrededor de $40.000, todo esto si sus servicios de TI, incluida la intranet, están en óptimas condiciones. En ciertas ocasiones por problemas técnicos la intranet tuvo que ser detenida y esto representa pérdidas de $2.000 la hora aproximadamente. No se han detectado problemas de seguridad, pero

de darse en un futuro la compañía debe disminuir la superficie de ataque de sus diferentes servicios.

La compañía gestiona sus respaldos diarios, semanales y mensuales de los equipos de tecnología de la organización.

A continuación exponemos los costos de implementación de las soluciones a las vulnerabilidades con mayor impacto en el negocio versus los beneficios que se pueden alcanzar al llevar a cabo la implementación de las soluciones requeridas.

Costos

- Costos para mejorar las aplicaciones web con vulnerabilidades de mayor impacto en el negocio.
- Gastos incurridos en la detección y análisis de vulnerabilidades de las aplicaciones web actuales y futuras.
- Gastos de mantenimiento anual de los sistemas de la institución.
- Gastos por cursos y/o seminarios relacionados a la seguridad de aplicaciones impartidas al equipo de desarrollo de la compañía.
- Gastos de consultoría en caso de requerirla.

Tabla 20: Costos de implementación de las soluciones a vulnerabilidades de las aplicaciones web con mayor impacto en el negocio.

COSTOS			
Cant.	**Descripción**	**Precio Unitario**	**Total**
115	Gastos incurridos en la detección y análisis de vulnerabilidades	$100	$11,500.00
115	Costos de implementación de soluciones	$60	$6,900.00
3	Gastos de mantenimiento anual de los sistemas	$6,000.00	$18,000.00
5	Gastos por capacitación al personal de sistemas en seguridad de aplicaciones	$1,500.00	$7,500.00
1	Gastos por consultoría en seguridad informática	$3,000.00	$3,000.00
1	Gastos por la compra de un certificado digital duradero a 2 años.	$500.00	$500.00
TOTAL			**$47,400.00**

Beneficios

- Mejoramiento de la seguridad de las aplicaciones web del negocio.
- Evitar pérdidas por filtración y alteración de información sensible
- Evitar pérdidas por la no disponibilidad de los servicios del negocio.
- Ahorro de adquisición y mantenimiento de hardware y software.

Tabla 21: Beneficios por la implementación de las soluciones a vulnerabilidades de las aplicaciones web con mayor impacto en el negocio.

BENEFICIOS			
Cant.	**Descripción**	**Precio Unitario**	**Total**
115	Ahorro en gastos incurridos por mejoramiento de las aplicaciones web de intranet	$40.00	$4,600.00
1	Evitar pérdidas y/o divulgación de información sensible (precios, cartera de clientes, presupuestos, etc) del 3% de ventas anuales por fallos de seguridad en aplicaciones de intranet	$432,000.00	$432,000.00
1	Evitar pérdidas por la no disponibilidad de los servicios de intranet del 1.5% de ventas anuales	$216,000.00	$216,000.00
1	Ahorro en adquisición y mantenimiento de hardware y software anuales	$10,000.00	$10,000.00
TOTAL			**$662,600.00**

La relación de rentabilidad (Total Beneficios / Total Costos) entre los costos que se incurren y los beneficios que se obtienen es de 13.97, este factor indica que por cada dólar gastado obtenemos $13.97 de ganancia.

Por esta razón el esquema de seguridad propuesto para minimizar los riesgos de seguridad de las aplicaciones web de la compañía debe ser implementado.

5.4. Esquema de seguridad para prevenir, detectar o corregir vulnerabilidades de seguridad.

Los riesgos encontrados en los aplicativos web de la institución demostraron la fragilidad de la plataforma, dando lugar a que dichos riesgos se puedan materializar comprometiendo todos los sistemas de la institución, interrumpiendo los servicios de TI que la compañía ofrece a sus clientes y usuarios. El uso de la extranet no se ha expandido, pero si es usado en muchas ocasiones por los altos ejecutivos cuando estos viajan al exterior y desean revisar información de la compañía.

El principal objetivo del esquema de seguridad es mitigar los riesgos de seguridad descritos por el apartado OWASP Top 10 2013, dando lugar a que dicho esquema pueda ser usado para las aplicaciones web restantes y para las futuras.

Figura 5.1: Arquitectura ESAPI. [15]

Este esquema presenta el uso de la herramienta ESAPI Java (Enterprise Security API) que es una colección gratuita de métodos de seguridad que todo desarrollador debe conocer para construir aplicaciones seguras. Es una herramienta de código abierto y que está disponible para diferentes lenguajes, su configuración resulta sencilla y sus métodos intuitivos.

Se busca que los desarrolladores eviten reinventar la rueda y más en temas de seguridad; con solo ciertos pasos es posible asegurar nuestro aplicativo sin mayor esfuerzo.

Para configurar la herramienta ESAPI en cada aplicativo web de la compañía debemos realizar los siguientes pasos:

- Descargar la librería esapi-2.1.0.jar
- Descargar las dependencias, las mismas que deben ser adicionadas a cada fuente de las aplicaciones analizadas.

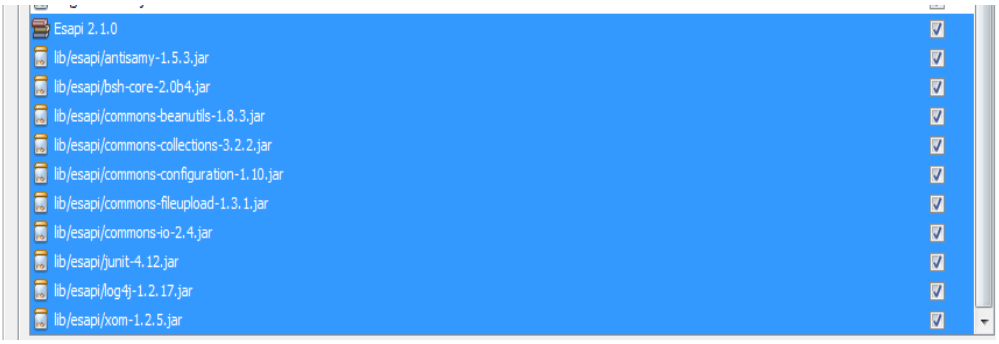


Figura 5.2: Adición de librerías a fuente de aplicativo web.

- Adición de archivos de configuración ESAPI en cualquier directorio del servidor de aplicaciones.

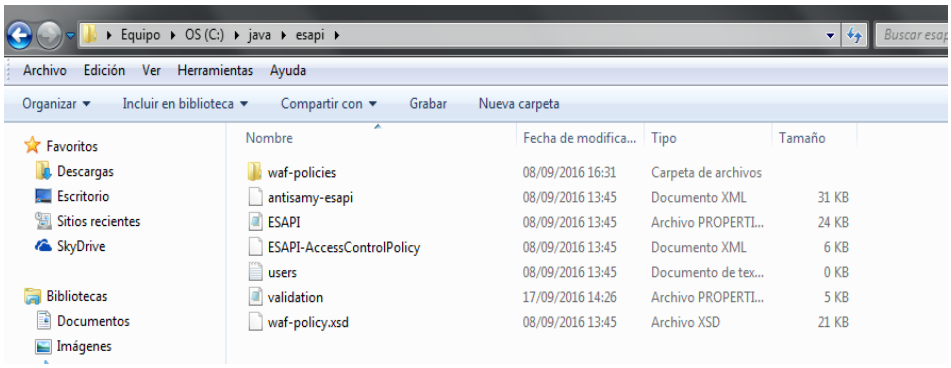


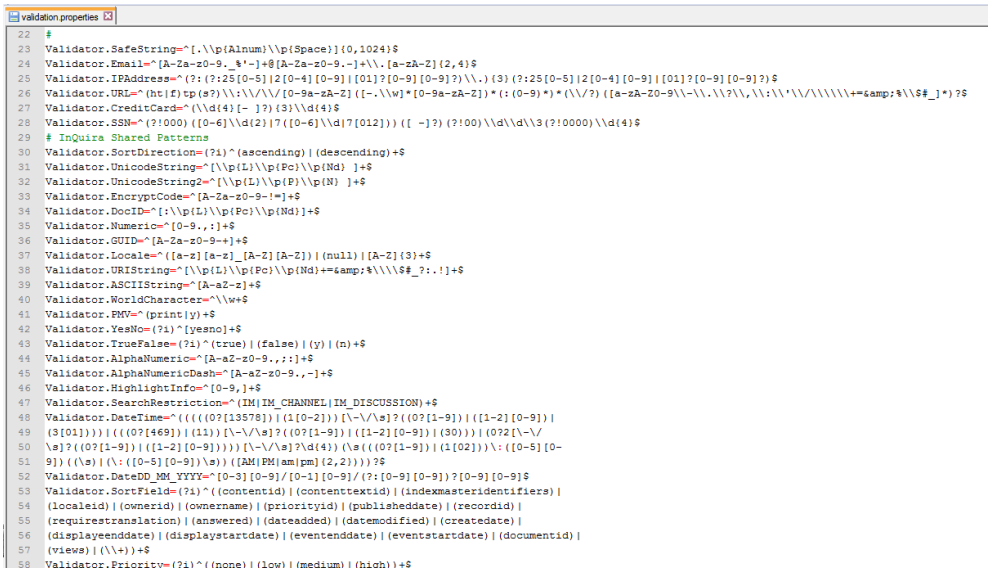
Figura 5.3: Archivos de configuración ESAPI.

Este paso solo se hace una vez ya que los archivos serán accedidos por todas las aplicaciones.

- En el script de arranque del servidor Jboss de la compañía se adiciona la propiedad *–Dorg.owasp.esapi.resources=C:\java\esapi*

- Esta propiedad permite a la herramienta acceder a los archivos de configuración de ESAPI Java.

- En el archivo de propiedades validation.properties se agregan las expresiones regulares (lista blanca) para validar la entrada de información en los campos de cada aplicativo web.



validation.properties

```
#
Validator.SafeString=^[.\\p{Alnum}\\p{Space}]{0,1024}$
Validator.Email=^[A-Za-z0-9._%'-]+@[A-Za-z0-9.-]+\\.[a-zA-Z]{2,4}$
Validator.IPAddress=^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$
Validator.URL=^(ht|f)tp(s?)\\://\\/[0-9a-zA-Z]([-.\\w]*[0-9a-zA-Z])*(:(0-9)*)*(\\/?)([a-zA-Z0-9\\-\\.\\?\\,\\:\\'\\/\\\\\\+=&amp;%\\$#_]*)?$
Validator.CreditCard=^(\\d{4}[- ]?){3}\\d{4}$
Validator.SSN=^(?!000)([0-6]\\d{2}|7([0-6]\\d|7[012]))([ -]?)(?!00)\\d\\d\\3(?!0000)\\d{4}$
# InQuira Shared Patterns
Validator.SortDirection=(?i)^(ascending)|(descending)+$
Validator.UnicodeString=^[\\p{L}\\p{Pc}\\p{Nd} ]+$
Validator.UnicodeString2=^[\\p{L}\\p{P}\\p{N} ]+$
Validator.EncryptCode=^[A-Za-z0-9-!=]+$
Validator.DocID=^[:\\p{L}\\p{Pc}\\p{Nd}]+$
Validator.Numeric=^[0-9.,:]+$
Validator.GUID=^[A-Za-z0-9-+]+$
Validator.Locale=^([a-z][a-z]_[A-Z][A-Z])|(null)|[A-Z]{3}+$
Validator.URIString=^[\\p{L}\\p{Pc}\\p{Nd}+=&amp;%\\\\$#_?:.!]+$
Validator.ASCIIString=^[A-aZ-z]+$
Validator.WorldCharacter=^\\w+$
Validator.PMV=^(print|y)+$
Validator.YesNo=(?i)^[yesno]+$
Validator.TrueFalse=(?i)^(true)|(false)|(y)|(n)+$
Validator.AlphaNumeric=^[A-aZ-z0-9.,;:]+$
Validator.AlphaNumericDash=^[A-aZ-z0-9.,-]+$
Validator.HighlightInfo=^[0-9,]+$
Validator.SearchRestriction=^(IM|IM_CHANNEL|IM_DISCUSSION)+$
Validator.DateTime=^(((((0?[13578])|(1[0-2]))[\-\/\s]?((0?[1-9])|([1-2][0-9])|
(3[01]))))|(((0?[469])|(11))[\-\/\s]?((0?[1-9])|([1-2][0-9])|(30))))|(0?2[\-\/
\s]?((0?[1-9])|([1-2][0-9]))))[\-\/\s]?\d{4})(\s(((0?[1-9])|(1[02]))\:([0-5][0-
9]))((\s)|(\:([0-5][0-9])\s))([AM|PM|am|pm]{2,2}))?$
Validator.DateDD_MM_YYYY=^[0-3][0-9]/[0-1][0-9]/(?:[0-9][0-9])?[0-9][0-9]$
Validator.SortField=(?i)^((contentid)|(contenttextid)|(indexmasteridentifiers)|
(localeid)|(ownerid)|(ownername)|(priorityid)|(publisheddate)|(recordid)|
(requirestranslation)|(answered)|(dateadded)|(datemodified)|(createdate)|
(displayeenddate)|(displaystartdate)|(eventenddate)|(eventstartdate)|(documentid)|
(views)|(\\+))+$
Validator.Priority=(?i)^((none)|(low)|(medium)|(high))+$
```

Figura 5.4: Archivo de configuración “validation.properties”.

Estas configuraciones son necesarias para el uso de la herramienta ESAPI ofreciendo una nueva alternativa de seguridad confiable y aplicable a cualquier aplicación web de la compañía.

A partir del esquema general de seguridad desarrollado (figura 4.18) las áreas de seguridad en el canal y seguridad de la aplicación se dividen en secciones específicas, a fin de llevar un control adecuado cuando se intente atacar cualquiera de los sitios analizados como aquellos que se podrían incluir en el futuro.

Cabe indicar que el estudio se inició con la detección de vulnerabilidades y después con el análisis de riesgos, esto porque en la compañía no se ha instaurado un procedimiento que permita analizar cada aplicativo web antes de ser desplegado en ambiente de producción, por tal motivo las demás aplicaciones web de la intranet se regirán a esta práctica también.

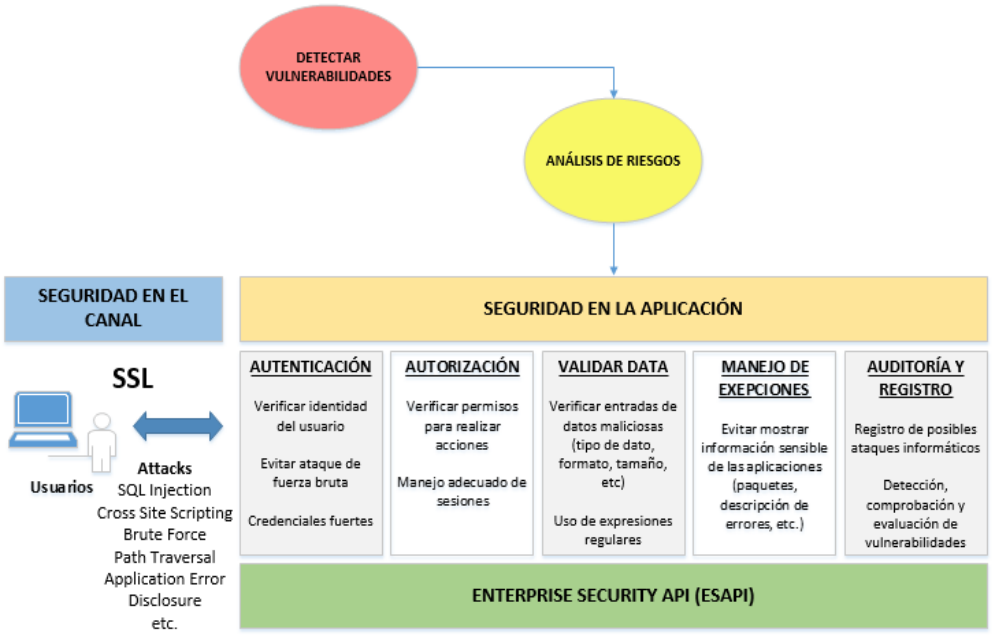


Figura 5.5: Esquema de seguridad para la detección, prevención y corrección de fallos de seguridad en las aplicaciones web del negocio.

Entre las secciones más importantes del esquema de seguridad, está el validar todo tipo de entrada de datos, esto permite disminuir sustancialmente la superficie de ataque porque limita y valida la información antes de ser enviada a la capa de negocios (base de datos).

Lo mismo sucede con las secciones que tienen que ver con la autenticación y autorización, ya que deben acceder usuarios permitidos y que las opciones habilitadas sean las correctas según el usuario logeado.

A partir de los riesgos más comunes presentados en las aplicaciones web de la compañía determinamos tipos de controles enfocados a cada sección del esquema de seguridad implementado.

Tabla 22: Tipos de controles implementados.

		Tipo de Control		
Control	**Esquema**	**Preventivo**	**Detectivo**	**Correctivo**
Canal de comunicación	Seguridad en el canal	X		
Cuentas de usuario	Autenticación	X		
Restringir acceso cuando superen límite de intentos fallidos.		X	X	
Cifrado de sesiones activas		X		
Asignación de funciones	Autorización	X		
Identificar y auditar accesos		X	X	
Accesos a repositorios permitidos		X		
Parametrización de consultas a las bases de datos	Validación		X	X
Verificar entradas con listas blancas			X	X
Escapar los datos que se mostrarán al usuario			X	X
Capacitación		X		
Uso de manejadores de errores	Excepciones			X
Uso de páginas genéricas de errores				X
Logs de transacciones	Auditoría y registros		X	
Auditorías internas y externas			X	
Evaluación de riesgos		X	X	

Evaluación de resultados			X	X

5.5. Esquema de mitigación del riesgo.

5.5.1. Seguridad en el Canal de Comunicaciones.

Se creó un certificado digital autofirmado usando la herramienta gratuita Open SSL. Esta herramienta, mediante el uso de comandos, permite generar la clave privada RSA, la misma que es cifrada por el algoritmo simétrico triple DES.

El certificado digital autofirmado tiene un tiempo de vigencia e información de la compañía y lo agregamos en el directorio “standalone” del servidor de aplicaciones, luego agregamos ciertas líneas de configuración en el archivo *standalone.xml* del mismo servidor (figura 5.7)

No es recomendable crear estos tipos de certificados porque un atacante puede valerse de esta vulnerabilidad para suplantar la identidad de la compañía y así robar información. Sin embargo, lo llevamos a cabo debido a que la compañía no pudo obtener un certificado digital autorizado mientras implementamos los cambios.

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host" native="true">
  <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http" redirect-port="443"/>
  <connector name="https" protocol="HTTP/1.1" scheme="https" socket-binding="https" secure="true">
    <ssl name="prueba-intranet-ssl"
         password="
         certificate-key-file="                    /standalone/configuration/prueba_intranet_iiasa.pem"
         certificate-file="                    /standalone/configuration/prueba_intranet_iiasa_cer.pem"/>
  </connector>
```

Figura 5.6: Habilitación de protocolo SSL en servidor de aplicaciones Jboss 7.1.1.

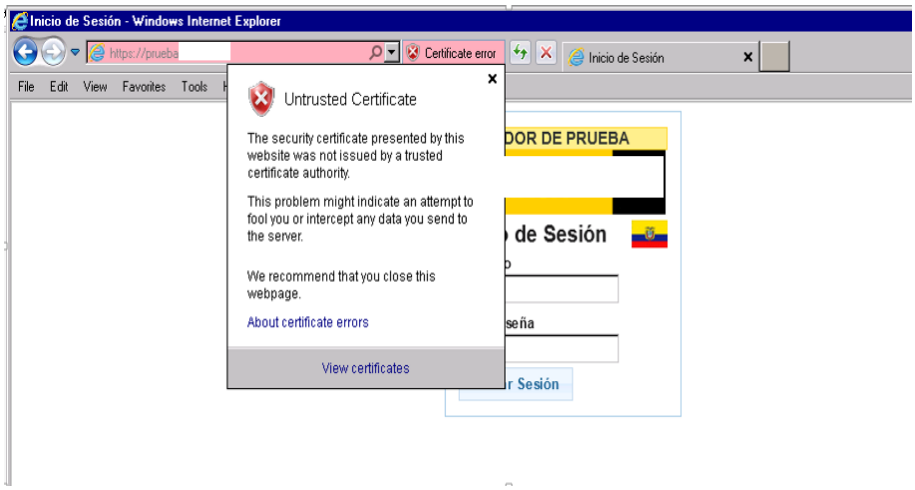


Figura 5.7: Despliegue de canal seguro sobre la página de inicio de la intranet.

5.5.2. Seguridad en la aplicación.

5.5.2.1. Autenticación.

La autenticación empleada en la intranet de la compañía para verificar que el usuario es quien dice ser, se lo hace a través del Active Directory lugar donde reside información del usuario (Nombres, usuario, clave, correo electrónico, etc). Esta comunicación se la realiza mediante el uso del protocolo

LDAP desde la pantalla login de intranet, se verifica que las credenciales sean las correctas y que el usuario este habilitado. Cabe indicar que la compañía mantiene una Política de Control de Acceso para la gestión de usuarios y acceso a la información.

Una vez aceptado el usuario, la intranet, almacena una data cifrada en una cookie la cual permite navegar sin problemas en las opciones asignadas al usuario. Esta verificación es revisada en cada uno de los filtros de cada aplicación a la cual el usuario desea acceder.

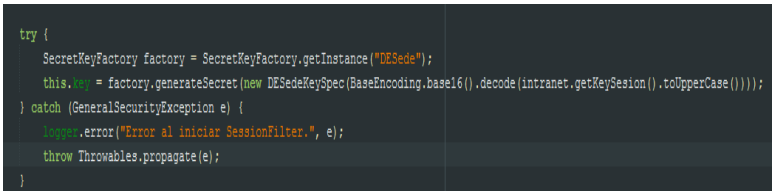


Figura 5.8: Uso de algoritmo de cifrado triple DES.

Figura 5.9: Funciones de cifrado y descifrado para cada sesión de usuario de intranet.

Una de las deficiencias que detectamos en esta sección es que no valida el número de intentos fallidos al momento de autenticarse; como sabemos es una falencia grave pues el

sitio puede ser víctima de ataques de fuerza bruta por agentes externos o internos que traten de acceder a información de otros usuarios sin el consentimiento del mismo.

Tal falencia fue analizada y se adicionó un control que permite saber la cantidad de veces que un usuario intenta autenticarse a la intranet y las veces que ha fallado, dando lugar a que el usuario una vez que supere un límite máximo de intentos se bloquee y no pueda acceder a ella después de superar el máximo intento. Para que pueda ser re-activado debe comunicarse con el departamento de sistemas e indicar el motivo por el cual se le bloqueo el usuario, sistemas procede a desbloquear el usuario y se resetea la clave del Active Directory para que en el siguiente inicio de sesión introduzca una clave nueva.

```
// Validamos que este activa la auditoria
if (intranet.getMaxLogin() != null) {
    String esquema = tipoValidacion == 1 ? "ACTIVE-DIRECTORY" : (tipoValidacion == 2 ? "BASE-DATOS" : "AS400");

    Util.checkUserMaxLogin(login, esquema, intranet);

    LOG = new IntranetLog();
    LOG.setEsquema(esquema);
    LOG.setIp(request.getRemoteAddr());
    LOG.setUsuario(login);
}
```

Figura 5.10: Validación que restringe el acceso cuando usuario supera el límite de intentos permitidos.

5.5.2.2. Autorización.

Las opciones de intranet son asignadas mediante un memo de opciones, el cual es revisado y autorizado por el área de

auditoría, dicho memo llega al departamento de sistemas y un administrador es el encargado de asignar las funciones al usuario. Se genera memo de opciones para usuarios temporales y permanentes.

Por el momento no existe una auditoría que registre el acceso fallido y exitoso a la plataforma de intranet, por lo que se implementó un control que permite conocer el acceso autorizado a la plataforma de manera tal que podamos encontrar algún intruso y detenerlo.

```
if (LOG != null){
    LOG.setEstado("F");
    Util.auditLogin(LOG);
}
```

Figura 5.11: Registro de auditoría en el acceso a intranet.

El uso de filtros en cada aplicativo web es de vital importancia porque permite segregar las funciones públicas de las privadas y además se configura un tiempo máximo de inactividad, de manera que dichas sesiones se reactivan una vez que el usuario registre sus credenciales de intranet nuevamente.

5.5.2.3. Validar Data.

Consideramos que esta sección es de las más importantes del esquema de seguridad debido a que todas las

vulnerabilidades analizadas se encasillan dentro de esta sección. En la mayoría de los casos dichas vulnerabilidades aparecen cuando no se tiene una validación correcta sobre los datos que el usuario proporciona o simplemente existen ciertas deficiencias de programación.

Todas las entradas realizadas al intérprete de datos (base de datos) fueron parametrizadas y se evitó la concatenación de cadenas. Esto fue corregido en todas las instancias en donde se pudo detectar este fallo de seguridad.

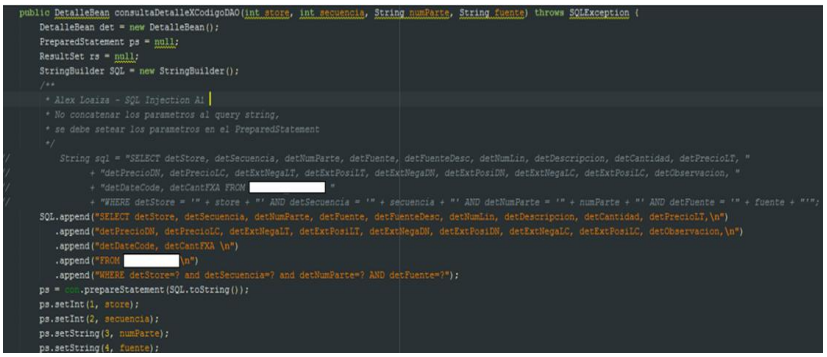


Figura 5.12: Envío de parámetros al intérprete de datos (Base de Datos).

La herramienta ESAPI valida estas entradas mediante el uso de listas blancas y no permite que la sentencia se ejecute en la base de datos.

```
/**
 * Alex Loaiza - Validamos entrada de
 * información para luego ser procesada.
 */
sos = ESAPI.validator().getValidInput(Req1912aAction.class.getName(),
        sos, "AlphaNumeric", 5, true);

store = ESAPI.validator().getValidInput(Req1912aAction.class.getName(),
        store, "AlphaNumeric", 5, true);

mes = Integer.parseInt(ESAPI.validator().getValidInput(Req1912aAction.class.getName(),
        String.valueOf(mes), "Numeric", 5, true));
```

Figura 5.13: Verificación de entradas uso de ESAPI.

La información que se devuelve al cliente debe ser “escapada” es decir mostrar la información tal y como fue escrita por el cliente para evitar que dichos datos se ejecuten del lado cliente.

Esta vulnerabilidad fue mitigada usando la herramienta ESAPI, con funciones que permiten codificar la entrada de datos y mostrarlos al usuario de acuerdo al archivo que desea visualizar (HTML, CSS, JavaScript, URL, etc.)

5.5.2.4. Manejo de excepciones.

Un adecuado manejo de excepciones permite al desarrollador o administrador de aplicaciones conocer las diferentes falencias del aplicativo. Esta información debe ser visible solo para personal de sistemas y no debe presentarse a otros usuarios.

Si las excepciones que se presentan en las aplicaciones no son manejadas, se corre el riesgo de que información sensible de la aplicación (paquetes, clases, líneas de código, error, etc)

sea visible para el usuario, además de dar un aspecto poco agradable, esta información pueda ser usada para otros fines.

Todas las aplicaciones manejan, de forma adecuada, las excepciones que puede producir una entrada de datos, el error es revisado en el Log y no desde la aplicación.

Se realizó una página de “error” en la que se indica al usuario que la aplicación tuvo un problema, pero se imprime información del error en el Log del servidor que luego será revisada por el departamento de sistemas.

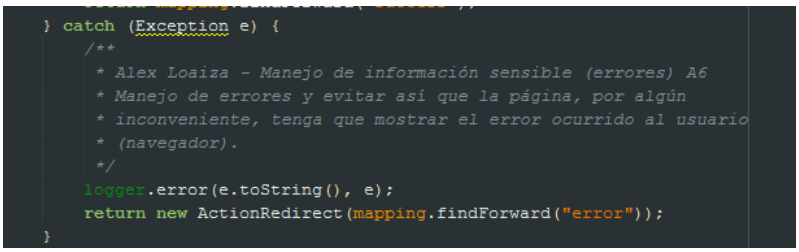


Figura 5.14: Manejo y auditoría de errores.

5.5.2.5. Auditoría y registros

Es necesario coordinar con el departamento de sistemas que todas las aplicaciones empleen funciones o herramientas de Logging, pues estas herramientas facilitan la detección oportuna de eventualidades propias de las aplicaciones, como de algún ataque interno o externo.

Inculcamos a los desarrolladores que las aplicaciones web, antes de ser subidas a producción, sean escaneadas y en el

caso de detectar fallos de seguridad tomen en cuenta lo desarrollado en el presente estudio, a fin de mejorar la seguridad de las aplicaciones.

CAPÍTULO 6.

6. ANÁLISIS DE RESULTADOS.

6.1. Validación de seguridades implementadas.

Para validar los riesgos se tomó en cuenta las diferentes instancias generadas a partir del primer escaneo de vulnerabilidades de la herramienta Zed Attack Proxy.

Las instancias fueron tomadas al azar de manera que podamos verificar la correcta implementación del esquema de seguridad propuesto. Estas instancias corresponden a vulnerabilidades cuyo nivel de severidad fue calculado como alto o medio según la metodología de análisis de riesgo OWASP.

A continuación, mediante diferentes ejemplos, mostramos como fueron mitigados los riesgos después de la implementación del esquema de seguridad.

6.1.1. Riesgo de inyección (A1).

Para validar el riesgo de inyección empleamos la aplicación **WEB-APP-C12**, en la que se detectó que unas de las opciones era susceptible al ataque de "Inyección SQL" (Figura 6.1). Al colocar cierta cadena maliciosa, el aplicativo mostró la información requerida sin detallar al usuario que el valor indicado no cumplía los parámetros necesarios (filtrado de entradas) para visualizar la información solicitada.

Después de implementar el esquema de seguridad "**Validar Data**", se pudo observar que al hacer el mismo ataque, la opción, ya no presenta los defectos antes vistos, por lo que dicho control mitigó el fallo detectado (Figura 6.2).

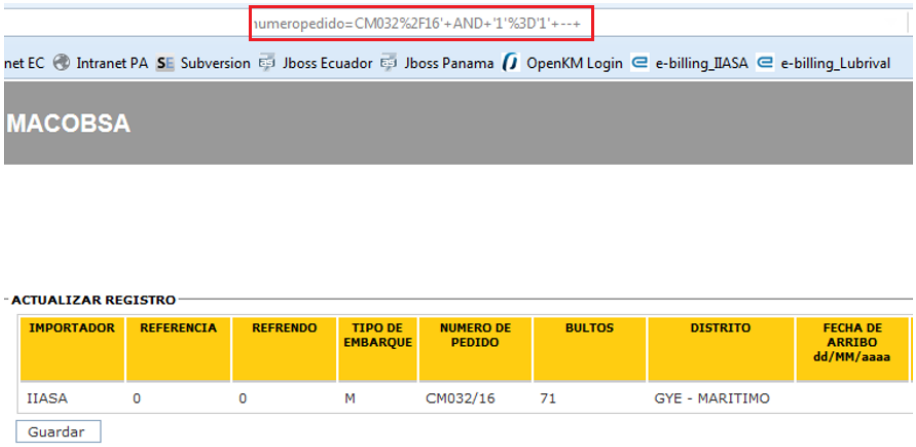


Figura 6.1: Riesgo de inyección SQL (A1) en aplicativo WEB-APP-C15.

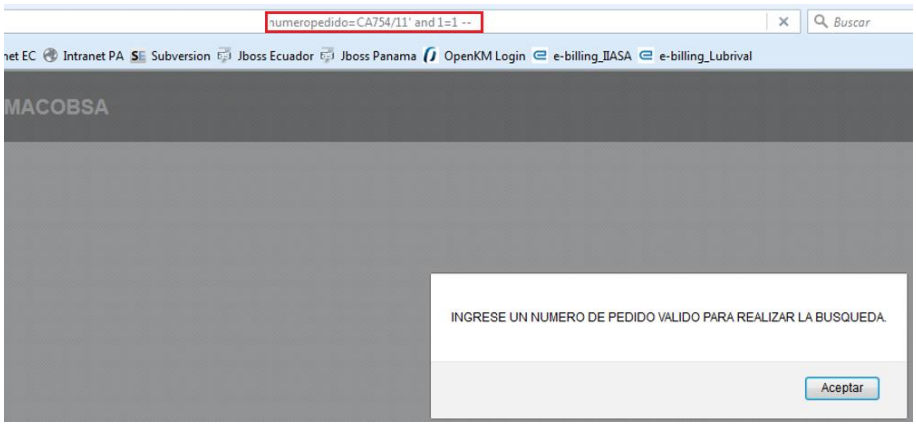


Figura 6.2: Riesgo mitigado de inyección SQL (A1) en aplicativo WEB-APP-C15.

6.1.2. Riesgo de secuencia de comandos en sitios cruzados (A3).

Usaremos como ejemplo la aplicación **WEB-APP-CC2**, la misma que era susceptible a ataques “XSS (reflejada)”, de la cual tomaremos una de las instancias consideradas como vulnerable y lanzaremos un código malicioso (generalmente Javascript) como parámetro en la consulta realizada al aplicativo demostrando así, que dicha ejecución pudo realizarse sin ningún contratiempo (Figura 6.3).

Una vez implementado el esquema de seguridad “**Validar Data**” para el aplicativo en mención se pudo observar que al lanzar el mismo código malicioso, dicha aplicación no permitió la ejecución del mismo sino que lo direccionó a una pantalla de error en donde se indica que la petición solicitada no pudo ser completada, mitigando así la vulnerabilidad descrita (Figura 6.4).

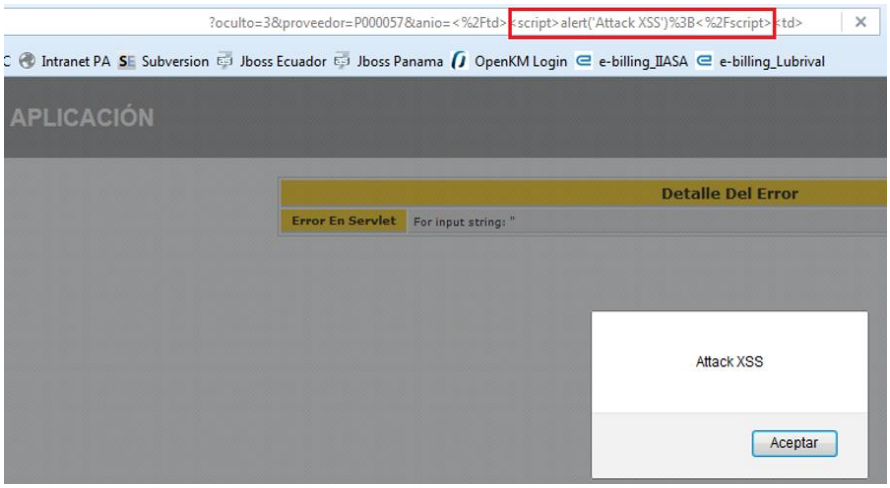


Figura 6.3: Riesgo de secuencia de comandos en sitios cruzados XSS (A3) en aplicativo WEB-APP-S02.

Figura 6.4: Riesgo mitigado de ejecución de comandos en sitios cruzados XSS (A3) en aplicativo WEB-APP-S02.

6.1.3. Riesgo de referencia directa insegura (A4).

En esta ocasión usaremos como ejemplo la aplicación **WEB-APP-I04**, la misma que presentó una instancia relacionada a la vulnerabilidad de "Directory Traversal". El ataque se lleva a cabo enviando como

parámetro una cadena maliciosa que le permite acceder a archivos del servidor de aplicaciones sin ninguna restricción (Figura 6.5).

Para este riesgo el esquema de seguridad está alineado en validar las entradas "**Validar Data**", asegurando que el parámetro enviado sea filtrado con una lista blanca de nombres de archivos, además solo accede a los archivos del directorio permitido. Al enviar nuevamente el ataque sobre la misma opción se puede observar que el sitio no permitirá descargar la información solicitada sino más bien se enviará una página de error dando a entender que los parámetros enviados no cumplen las especificaciones (Figura 6.6).

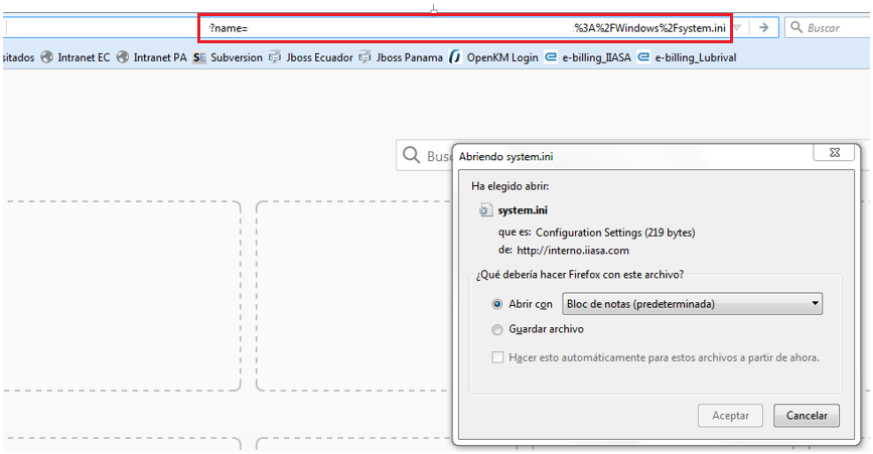


Figura 6.5: Riesgo de Referencia directa insegura a objetos (A4) en aplicativo WEB-APP-I02.

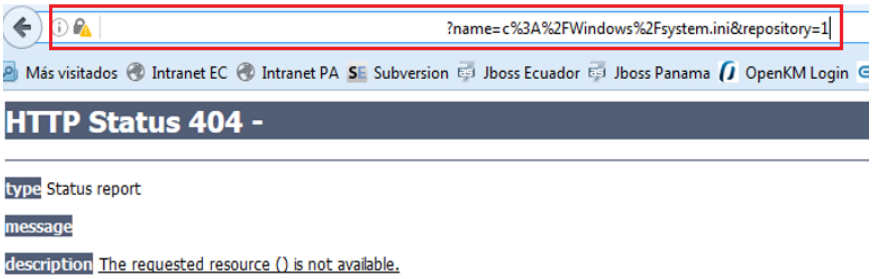


Figura 6.6: Riesgo mitigado de Referencia directa insegura a objetos (A4) en aplicativo WEB-APP-I02.

6.1.4. Riesgo de configuración de seguridad incorrecta (A5).

Usaremos la aplicación **WEB-APP-C12** donde se detectó vulnerabilidades de “Application Error Disclosure”, la misma que da a conocer información del error presentado y la línea de código en donde el aplicativo se detiene. Aunque parezca información no entendible para el usuario común, si lo es para un atacante con cierto conocimiento en programación.

Al igual que los demás riesgos analizados, implementamos las especificaciones indicadas por el esquema de seguridad en la sección “**Validar Data”**, en donde indica que toda traza de código debe tener un manejador de excepciones, ya que en el caso de presentarse algún error este solo debe mostrarse en el Log del servidor. Este error debe ser legible y entendible para los desarrolladores por ello es importante el uso “Loggers” en el aplicativo.

6.2. Plan de mitigación implementado.

El plan de mitigación presenta información en caso de que se presente una nueva vulnerabilidad en las aplicaciones web de la compañía o también sirve como guía para el desarrollo interno de futuras aplicaciones.

El plan de mitigación cuenta con la siguiente estructura:

1) Nombre del riesgo: En esta sección se informa del riesgo encontrado y se lo relaciona con el apartado OWASP Top 10 2013.
2) Descripción del riesgo: Se describe brevemente el riesgo para informar de los posibles daños que puede ocasionar en el aplicativo o su entorno.
3) Severidad del riesgo: En esta sección se presenta la severidad del riesgo, esta severidad se obtiene a partir del análisis de riesgo usando la metodología de OWASP.
4) Planes de acción: Se indica las diferentes formas de solucionar el fallo encontrado, revisiones periódicas del código del programa antes de su paso a producción y capacitación constante al personal del departamento de sistemas.
5) Plan de contingencia: A pesar de los controles implementados, se debe llevar a cabo un plan de contingencia que permita restaurar los servicios proporcionados por las aplicaciones y su entorno.
6) Responsables: Se indica el personal de la compañía encargado de informar y velar por las vulnerabilidades detectadas en el aplicativo web.
7) Recursos: Indica las diferentes herramientas necesarias para un control al momento de encontrar vulnerabilidades en las aplicaciones web de la compañía.

A continuación se detallan los anexos que corresponden a cada plan de mitigación elaborado para los riesgos con mayor importancia encontrados en las aplicaciones web de la plataforma intranet.

- Anexo G – Plan de mitigación para riesgos relacionados a inyección (A1).
- Anexo H – Plan de mitigación para riesgos relacionados a secuencia de comandos en sitios cruzados (A3).
- Anexo I – Plan de mitigación para riesgos relacionados a referencia directa insegura a objetos (A4).
- Anexo J – Plan de mitigación para riesgos relacionados a la configuración de Seguridad Incorrecta (A5).

6.3. Evaluación de la matriz de riesgos de las aplicaciones web.

La matriz de riesgo permitió conocer de entre todas las vulnerabilidades encontradas cuales de ellas deben ser mitigadas en primera instancia, por ello es importante revisar si esta matriz debe actualizarse y porque.

Para esto realizamos reuniones con el personal involucrado en la elaboración de la matriz de riesgo inicial, estas reuniones determinaron que dicha matriz no debía sufrir cambios, pero una vez maduro el esquema de seguridad propuesto se pueda adaptar más pesos relacionados a cada factor evaluado para definir de mejor manera la severidad de riesgo asociada.

Además se mencionó que dicho esquema deba actualizarse constantemente y que se mejore día a día, de manera que sea utilizado también para las aplicaciones web del grupo en Panamá.

6.4. Resultados obtenidos.

6.4.1. Efectividad del esquema de seguridad implementado en las aplicaciones web de la compañía.

El esquema de seguridad para las aplicaciones web implementado en la compañía dio resultados alentadores porque la mayoría de vulnerabilidades con severidad media fueron mitigadas, al igual que las vulnerabilidades catalogadas con severidad baja. En ciertos casos aparecieron nuevas vulnerabilidades debido a modificaciones en las aplicaciones después de realizado el escaneo de vulnerabilidades inicial.

Es determinante conocer si el esquema de seguridad implementado cumple los requisitos que la compañía planteó al inicio del presente proyecto, por lo que a continuación indicaremos mediante tablas los valores obtenidos después de implementar el esquema de seguridad y confirmar si el esquema cumple las especificaciones indicadas al inicio.

6.4.1.1. Resultados departamento de Contraloría.

La tabla 23 presenta información resumida de las instancias obtenidas en el primer escaneo de vulnerabilidades realizado a las aplicaciones web del departamento de contraloría, en

comparación con las instancias obtenidas en el nuevo escaneo de vulnerabilidades.

De manera general podemos mencionar que el esquema de seguridad resultó favorable para los intereses de la compañía pues el 99% del total de vulnerabilidades presentadas en las aplicaciones de este departamento fueron mitigadas y/o evitadas por los diferentes controles realizados en cada aplicativo.

Tabla 23: Efectividad de controles sobres las aplicaciones web del departamento de Contraloría.

APLICACIONES WEB CONTRALORIA					Análisis de riesgo
Vulnerabilidad	Instancias		Efec. control	Top 10 Owasp	
	Ini.	Act.			
Falla por Inyección SQL	58	3	95%	A1	Medio
Inyección Remota de Comandos OS	24	1	96%	A1	Medio
Content-Type Header Missing	16	0	100%	A2	Bajo
Cross Site Scripting	11	0	100%	A3	Medio
Directory Traversal	18	0	100%	A4	Medio
X-Frame-Options Header Not Set	193	4	98%	A5	Bajo
Application Error Disclosure	9	0	100%	A5	Medio
Web Browser XSS Protection Not Enabled	185	0	100%	A5	Bajo
X-Content-Type-Options Header Missing	186	0	100%	A5	Bajo
Password Autocomplete in Browser	13	1	92%	A5	Bajo

Cross-Domain JavaScript Source File Inclusion	1	0	100%	A5	Bajo
Efec. Total	**714**	**9**	**99%**		

6.4.1.2. Resultados departamento de Crédito & Cobranzas y Maquinaria.

El nuevo escaneo de vulnerabilidades realizado a las aplicaciones web de los departamentos de Crédito y Maquinaria muestra resultados alentadores porque la totalidad de vulnerabilidades halladas inicialmente fueron mitigadas en un 100% después de llevar a cabo los controles respectivos en cada aplicación. Las tablas 24 y 25 describen los resultados del escaneo de vulnerabilidades realizado en los diferentes departamentos evaluados.

Tabla 24: Efectividad de controles sobres las aplicaciones web del departamento de C. y Cobranzas.

APLICACIONES WEB CREDITO Y COBRANZAS					Análisis de riesgo
Vulnerabilidad	Instancias		Efec. Control	Top 10 Owasp	
	Ini.	Act.			
Falla por Inyección SQL	28	0	100%	A1	Medio
Content-Type Header Missing	3	0	100%	A2	Bajo
Cross Site Scripting	2	0	100%	A3	Medio
X-Frame-Options Header Not Set	66	0	100%	A5	Bajo
Web Browser XSS Protection Not Enabled	64	0	100%	A5	Bajo

X-Content-Type-Options Header Missing	68	0	100%	A5	Bajo
Efec. Total	231	0	100%		

Tabla 25: Efectividad de controles sobres las aplicaciones web del departamento de Maquinaria.

APLICACIONES WEB MAQUINARIA					Análisis de riesgo
Vulnerabilidad	Instancias		Efec. Control	Top 10 Owasp	
	Ini.	Act.			
Inyección Remota de Comandos OS	1	0	100%	A1	Medio
Falla por Inyección SQL	14	0	100%	A1	Medio
Cross Site Scripting	3	0	100%	A3	Medio
X-Frame-Options Header Not Set	68	0	100%	A5	Bajo
Web Browser XSS Protection Not Enabled	68	0	100%	A5	Bajo
X-Content-Type-Options Header Missing	68	0	100%	A5	Bajo
Password Autocomplete in Browser	8	0	100%	A5	Bajo
Efec. Total	**230**	**0**	**100%**		

6.4.1.3. Resultados departamento de Repuestos.

Podemos mencionar que después de llevar a cabo los controles de seguridad en los aplicativos web del departamento de repuestos obtuvimos que el 98% de la totalidad de vulnerabilidades fue mitigada gracias al esquema de seguridad propuesto, sin embargo un 2% de estas

vulnerabilidades se mantuvieron. Las vulnerabilidades que se mantuvieron y que tienen severidad de riesgo media deben ser revisadas nuevamente con el fin de evitar que estos fallos de seguridad se mantengan, al igual que aquellas con severidad baja.

La tabla 26 detalla la efectividad de los controles que se implementaron en las aplicaciones de dicho departamento.

Tabla 26: Efectividad de controles sobres las aplicaciones web del departamento de Repuestos.

APLICACIONES WEB REPUESTOS					Análisis de riesgos
Vulnerabilidad	Instancias		Efec. Control	Top 10 Owasp	
	Ini.	Act.			
Inyección Remota de Comandos OS	2	0	100%	A1	Medio
Falla por Inyección SQL	40	1	98%	A1	Medio
Cross Site Scripting	67	2	97%	A3	Medio
Directory Traversal	1	0	100%	A4	Medio
X-Frame-Options Header Not Set	69	0	100%	A5	Bajo
Web Browser XSS Protection Not Enabled	69	0	100%	A5	Bajo
X-Content-Type-Options Header Missing	73	2	97%	A5	Bajo
Efec. Total	**321**	**5**	**98%**		

6.4.1.4. Resultados departamento de Servicios.

Las vulnerabilidades encontradas en las aplicaciones de departamento de servicios mostraron que el 99% de la totalidad de vulnerabilidades fueron mitigadas y solo el 1% de

ellas persistieron a pesar de los controles realizados. Entre las que persistieron son aquellas catalogadas con severidad baja pues según el análisis de riesgos estas no representan mayor problema para las aplicaciones web de este departamento en la compañía. A continuación en la tabla 27 se presenta la información de la efectividad de los controles realizados en los aplicativos web del departamento de servicios.

Tabla 27: Efectividad de controles sobres las aplicaciones web del departamento de Servicios.

APLICACIONES WEB SERVICIOS					Análisis de riesgos
Vulnerabilidad	Instancias		Efec. control	Top 10 Owasp	
	Ini.	Act.			
Inyección Remota de Comandos OS	1	0	100%	A1	Medio
Falla por Inyección SQL	21	0	100%	A1	Medio
Cross Site Scripting	16	0	100%	A3	Medio
Directory Traversal	20	0	100%	A4	Medio
X-Frame-Options Header Not Set	55	0	100%	A5	Bajo
Web Browser XSS Protection Not Enabled	55	0	100%	A5	Bajo
X-Content-Type-Options Header Missing	55	2	96%	A5	Bajo
Password Autocomplete in Browser	2	0	100%	A5	Bajo
Private IP Disclosure	1	0	100%	A5	Bajo
Efec. Total	**226**	**2**	**99%**		

6.4.1.5. Resultados departamento de Sistemas.

Las vulnerabilidades encontradas en las aplicaciones de departamento de sistemas mostraron que el 99% de la totalidad de vulnerabilidades fueron mitigadas y solo el 1% de ellas persistieron a pesar de los controles realizados. Entre las que persisten son aquellas catalogadas con severidad baja es decir, las correcciones de seguridad no son tan prioritarias. Dentro de estas aplicaciones se realizaron las mejoras con respecto al esquema de seguridad relacionado con las secciones de autenticación y autorización.

A continuación en la tabla 28 se presenta la efectividad de los controles implementados a cada aplicación web de este departamento.

Tabla 28: Efectividad de controles sobres las aplicaciones web del departamento de Sistemas.

APLICACIONES WEB SISTEMAS					Análisis de riesgos
Vulnerabilidad	Instancias		Efec. Control	Top 10 Owasp	
	Ini.	Act.			
Falla por Inyección SQL	4	0	100%	A1	Medio
Content-Type Header Missing	2	0	100%	A2	Bajo
Cross Site Scripting	2	0	100%	A3	Medio
Directory Traversal	1	0	100%	A4	Medio
X-Frame-Options Header Not Set	68	0	100%	A5	Bajo
Web Browser XSS Protection Not Enabled	67	2	97%	A5	Bajo

X-Content-Type-Options Header Missing	68	0	100%	A5	Bajo
Private IP Disclosure	7	1	86%	A5	Bajo
Cookie No HttpOnly Flag	4	0	100%	A5	Bajo
Password Autocomplete in Browser	13	0	100%	A5	Bajo
Efec. Total	**236**	**3**	**99%**		

6.4.2. Resumen Final.

El presente trabajo permitió conocer la realidad de las aplicaciones web que forman parte de la plataforma de intranet de la compañía IIASA. Estas aplicaciones, como se nos había informado, se desarrollaban en función de las necesidades departamentales para mejorar sus procesos; se destinaba todo el esfuerzo, por parte del equipo de desarrolladores, para que el programa funcione correctamente. Las seguridades de los aplicativos quedaban a criterio del desarrollador de turno que bien o mal realizaban ciertos controles para asegurar las entradas de datos y acceso a las bases de datos.

El departamento de sistemas seleccionó, del grupo de aplicaciones que conforman la intranet, **50 aplicativos** a los cuales se les realizó la detección de vulnerabilidades, el análisis de riesgo y se implementó las soluciones requeridas para mitigar los riesgos con mayor criticidad anteriormente analizados.

El ***plan de detección de vulnerabilidades*** consistió en elegir la herramienta de escaneo alienada al apartado OWASP Top 10 2013 y que se ajuste a las necesidades de la compañía.

La ***metodología de riesgos de OWASP*** permitió determinar, del gran grupo de vulnerabilidades encontradas, cuál de ellas deben ser consideradas para la implementación de soluciones. El uso de esta metodología, implicó la participación de personal de los departamentos relacionados a las aplicaciones elegidas, transmitiendo sus criterios con respecto al impacto en el negocio de las vulnerabilidades encontradas.

El ***esquema de seguridad*** implantado en la compañía, permitió disminuir en gran medida los riesgos encontrados en las aplicaciones; es fácil de mantener porque emplea una herramienta genérica aplicable a los programas de la compañía y se puede actualizar permanentemente.

Los requerimientos solicitados por el departamento de sistemas fueron implementados en ambiente de prueba. Todas las verificaciones resultaron satisfactorias ***pero por disposición y la no disponibilidad de recursos***, los cambios se mantuvieron en prueba. Quedó a disposición de la gerencia de sistemas y de la alta gerencia tomar la decisión de implementar los cambios en ambiente de producción en un futuro cercano.

En resumen, podemos indicar que el trabajo realizado en la compañía cumplió las expectativas esperadas y se logró fomentar el uso del esquema de seguridad propuesto para las aplicaciones actuales y futuras de la compañía.

CONCLUSIONES Y RECOMENDACIONES.

CONCLUSIONES.

Después de conocer cada uno de los capítulos del presente trabajo de titulación podemos llegar a las siguientes conclusiones:

1. El apartado OWASP Top 10 2013, en donde se describen los diferentes riesgos que suelen existir en las aplicaciones web de todo tipo de organización (gobierno, comerciales, industriales, educativas, etc.), permitió tener un mejor enfoque de las diferentes vulnerabilidades que se pueden presentar en dichas aplicaciones, de manera que puedan ser identificadas y tratadas correctamente mitigando así el riesgo involucrado.

2. Entender la arquitectura de las aplicaciones con las que cuenta la compañía IIASA fue importante para la fase de detección de vulnerabilidades debido a que permitió conocer los diferentes fallos de seguridad en cada una de ellas y la concurrencia con las que se presentan. Además es importante seleccionar

adecuadamente el software de apoyo para la detección de vulnerabilidades de manera que pueda alinearse con el apartado de OWASP Top 10 2013.

3. En la fase de análisis de riesgos, la implementación de la metodología de análisis de riesgo proporcionada por OWASP, es de vital importancia porque permite medir cada vulnerabilidad detectada de acuerdo a criterios de probabilidad de ocurrencia e impacto global, determinando así la severidad de riesgo que puede presentar cada vulnerabilidad. De esta manera enfocamos nuestro esfuerzo en mitigar los riesgos más severos y luego los de menor severidad.

4. En la fase de implementación de soluciones es importante brindar información sobre los diferentes beneficios que se obtendrían al llevar a cabo el plan de mitigación de riesgos y sus diferentes costos que acarrearía dicha implementación. Esta información es importante para la toma de decisiones por parte de los ejecutivos de la compañía.

5. Dentro de las vulnerabilidades encontradas pudimos percatarnos que las relacionadas con riesgos de inyección (A1) y secuencia de comandos en sitios cruzados XSS (A3) fueron las más recurrentes en los aplicativos web de la compañía, esto debido a la escasez de validaciones en las entradas de información de cada aplicativo web, las mismas que fueron mitigadas empleando el esquema de seguridad “validar data”.

6. Dentro del grupo de vulnerabilidades encontradas en los aplicativos web de la compañía, un alto porcentaje se relaciona al riesgo de configuración de

seguridad incorrecta (A5), las mismas que pudieron ser mitigadas con la inclusión de líneas de código en una librería genérica para todas las aplicaciones web de intranet donde se gestiona, mediante el uso de filtros, el acceso a las opciones de intranet volviéndolo fácil y rápido de incluir.

7. El uso del API de Java, ESAPI, en las aplicaciones web de la compañía permitió establecer mejoras de seguridad en todas las aplicaciones analizadas, disminuyendo significativamente la presencia de fallos de seguridad. Este conjunto de librerías debe ser implementado en todas las aplicaciones web de la compañía con la finalidad de generar software seguro.

8. El empleo de comunicaciones seguras mediante el uso del protocolo HTTPS en la plataforma de intranet mejoró las conexiones, previniendo así el robo de información por parte de atacantes internos o externos. Esta implementación debe llevarse a cabo en su totalidad pues se dejó configurado el servidor de aplicaciones con un certificado digital autofirmado, lo correcto es adquirir un certificado digital validado por una entidad certificadora.

9. Es importante llevar a cabo una nueva detección de vulnerabilidades después de implementar las correcciones de seguridad con el fin de verificar la efectividad de los cambios realizados en cada una de las aplicaciones web analizadas en la fase de análisis de riesgos.

10. El esquema de seguridad planteado y ejecutado en la compañía IIASA demuestra que fue el adecuado para prevenir y corregir falencias de seguridad en sus aplicaciones web. Del 100% de vulnerabilidades encontradas y

analizadas, el 99.03% fueron mitigadas superando las expectativas iniciales de la compañía sobre el presente trabajo.

11. Los resultados obtenidos demuestran que llevando a cabo correctamente las diferentes fases del esquema de seguridad (detección, análisis de riesgos e implementación de soluciones) se puede mejorar sustancialmente la seguridad de las aplicaciones de cualquier organización que no ha llevado un control de seguridad necesario pues detectamos las vulnerabilidades, las analizamos y les damos valor a cada una de ellas y por último realizamos los cambios necesarios para evitar que la vulnerabilidad persista en las aplicaciones web, por esto y más el esquema de seguridad propuesto superó las expectativas iniciales adaptándolo incluso como una directriz de seguridad para las aplicaciones web de la compañía actuales, nuevas o de terceros.

12. No solo basta tener los suficientes conocimientos técnicos en el desarrollo de aplicativos web o de cualquier índole, también es necesario considerar que todo programa debe regirse bajo directrices de seguridad que permiten disminuir la posibilidad de un ataque informático interno o externo asegurando así la infraestructura informática de la organización.

RECOMENDACIONES.

A continuación se detalla una serie de recomendaciones, las mismas que deben llevarse a cabo a corto o mediano plazo con la finalidad de mejorar el esquema de seguridad implementado como también las aplicaciones web que no fueron consideradas en el presente estudio:

1. La implementación de las soluciones realizado en ambiente de prueba de intranet debe ser llevado a producción con los cambios generados en cada aplicativo web analizado y subir al servidor de fuentes todos los cambios pertinentes.

2. Adquirir un certificado digital que use 2048 bits y SHA-2 el mismo que debe ser adquirido a una autoridad certificadora confiable, es decir que el certificado emitido sea confiable a los navegadores web.

3. Se recomienda implementar procedimientos que permitan realizar pruebas de seguridad en las aplicaciones web de la compañía con la finalidad de encontrar vulnerabilidades de una forma más exhaustiva dando lugar a nuevas formas de control sobre los riesgos de seguridad encontrados.

4. Realizar Hardening a los equipos que conforman la intranet (servidor de aplicaciones, servidor de archivos, Bases de datos y As400) con el fin de prevenir posibles ataques informáticos a la infraestructura tecnológica de la compañía. Mantener programas de seguridad (antivirus, anti spam, antispyware, etc), evitar configuraciones por defecto, restricciones de software, configuraciones de permisos de seguridad en archivos y directorios, respaldos de seguridad, configuración de accesos remotos, etc.

5. Estar siempre al tanto sobre las diferentes vulnerabilidades que aparecen día en equipos de TI (firewall, routers, switches, sistemas operativos, etc.). Recibir notificaciones de CSIRT confiables en donde se puede revisar temas de seguridad informática.

6. Fortalecer aún más las políticas de acceso de usuarios a las diferentes plataformas tecnológicas de la compañía.

7. Mantener el software de terceros siempre actualizado, a fin de evitar bugs de seguridad que comprometan las aplicaciones web que lo usan.

BIBLIOGRAFÍA.

[1] OWASP, OWASP Top 10-2013 Los diez riesgos más críticos en Aplicaciones Web, https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Español.pdf , fecha de consulta mayo 2016.

[2] EcuRed, Seguridad Informática https://www.ecured.cu/Seguridad_Informática , fecha de consulta mayo 2016.

[3] Hernández Ana y Mejía Jezreel, Guía de Ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web, http://recibe.cucei.udg.mx/beta/revista/es/vol4-no1/computacion05.html , fecha de consulta mayo 2016.

[4] Wichers Dave, Owasp Top 10 2013, https://docs.google.com/viewer?url=https%3A%2F%2Fowasp.org%2Fimages%2F1%2F17%2FOWASP_Top-10_2013--AppSec_EU_2013_-_Dave_Wichers.pdf , fecha de consulta mayo 2016.

[5] Firvida Daniel, Seguridad Web: Auditorías y Herramientas, http://www.fundaciondedalo.org/archivos/ACTIVIDADES/SSI09/SeguridadWebAuditoriaHerramientas_INTECO.pdf , fecha de consulta junio 2016.

[6] Lonita Dan, Current established risk assessment methodologies and tools, http://eprints.eemcs.utwente.nl/23767/01/D_Ionita_-_Current_Established_Risk_Assessment_Methodologies_and_Tools.pdf , fecha de consulta junio 2016.

[7] Beroes Mario, Latinoamérica es “sumamente vulnerable” a ataques informáticos, http://www.cioal.com/2015/10/20/gigamon-latinoamerica-es-sumamente-vulnerable-a-ataques-ciberneticos/ , fecha de consulta junio 2016.

[8] Aguilera Vicente, OWASP Top 10 2013: actualización de los riesgos más extendidos asociados a las aplicaciones web, https://docs.google.com/viewer?url=http%3A%2F%2Fwww.isecauditors.com%2Fsites%2Fdefault%2Ffiles%2Ffiles%2FSIC106_OWASP-ISECA.pdf , fecha de consulta junio 2016.

[9] Jurcenoks Jesper, OWASP to WASC to CWE Mapping, https://docs.google.com/viewer?url=http%3A%2F%2Fwww.criticalwatch.com%2Fassets%2Fc-Owasp-to-Wasc-to-CWE-Mapping-Tech-Paper-0710131.pdf , fecha de consulta junio 2016.

[10] OWASP, OWASP Risk Rating Methodology, https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology , fecha de consulta julio 2016.

[11] Salgado Lenin, Ron Mario y Solís Francisco, Análisis de riesgos de las aplicaciones web de la superintendencia de bancos y seguros, utilizando las recomendaciones top ten de Owasp, https://docs.google.com/viewer?url=http%3A%2F%2Frepositorio.espe.edu.ec%2Fbitstream%2F21000%2F8246%2F1%2FAC-SI-ESPE-047920.pdf , fecha de consulta agosto 2016.

[12] Mohan Ram y Pant Durgesh, Security risk assessment of Geospatial Weather Information System (GWIS): An OWASP based approach, https://docs.google.com/viewer?url=https%3A%2F%2Fwww.researchgate.net%2Fprofile%2FRam_Mohan_Rao_K%2Fpublication%2F46217237_Security_risk_assessment_of_Geospatial_Weather_Information_System_(GWIS)_An_OWASP_based_approach%2Flinks%2F00b7d51f7a1c0d1795000000.pdf , fecha de consulta septiembre 2016.

[13] Mochaca Álvaro, Análisis de riesgos aplicando la metodología OWASP, https://docs.google.com/viewer?url=https%3A%2F%2Fwww.owasp.org%2Fimages%2Fb%2Fb3%2FAnalisis_de_riesgo_usando_la_metodologia_OWASP.pdf , fecha de consulta septiembre 2016.

[14] Paguay Paúl, Propuesta de técnicas de aseguramiento de aplicaciones web desarrolladas en Java, https://docs.google.com/viewer?url=http%3A%2F%2Fdspace.espoch.edu.ec%2Fbitstream%2F123456789%2F4029%2F1%2F20T00467.pdf , fecha de consulta octubre 2016.

[15] OWASP, Proyecto OWASP API de seguridad empresarial (ESAPI), https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API/es , fecha de consulta octubre 2016.

[16] Gómez Iván, Diseño de una metodología para verificar la seguridad en aplicaciones web contra inyecciones sql, https://docs.google.com/viewer?url=http%3A%2F%2Frepository.unimilitar.edu.co%2

Fbitstream%2F10654%2F7212%2F2%2FGomezGonzalezIvanCamilo2012.pdf , fecha de consulta octubre 2016.

[17] Pérez Ignacio, Comprendiendo la vulnerabilidad XSS (Cross-Site scripting) en sitios web, http://www.welivesecurity.com/la-es/2015/04/29/vulnerabilidad-xss-cross-site-scripting-sitios-web/ , fecha de consulta octubre 2016.

[18] Ortega Israel, Desarrollo seguro de aplicaciones y servicios web en ambiente Java, https://docs.google.com/viewer?url=http%3A%2F%2Fwww.redisybd.unam.mx%2Fmod%2Fresource%2FSeguridadDGPE.pdf , fecha de consulta noviembre 2016.

[19] Torres Daniel, Headers seguros HTTP, https://www.owasp.org/images/6/6a/Headers_seguros.pdf , fecha de consulta noviembre 2016.

ANEXOS.

Anexo A. Matriz de riesgos de las aplicaciones del departamento de Contraloría.

APLICACIONES WEB DEL DEPARTAMENTO DE CONTRALORIA (VULNERABILIDAD/CÁLCULO SEVERIDAD DEL RIESGO)	PROBABILIDAD												IMPACTO												SEVERIDAD DEL RIESGO
	Factores agentes de amenazas					Factores de vulnerabilidad					TOTAL	PROBABILIDAD GENERAL	NIVEL PROBABILIDAD	Factores de Impacto tecnico					Factores de Impacto del negocio					TOTAL	IMPACTO GLOBAL	NIVEL DE IMPACTO	
	Habilidades Técnicas	Motivación	Oportunidad	Tamaño	TOTAL	Facilidad de Descubrimiento	Facilidad de Explotación	Conciencia o Conocimiento	Detectores de Intrusión	TOTAL				Pérdida de la Confidencialidad	Pérdida de la Integridad	Pérdida de la Disponibilidad	Pérdida de la Autibilidad	TOTAL	Daño Económico	Daño de Imagen	Incumplimiento	Violación a la Privacidad	TOTAL				
WEB-APP-C01																										
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	3	7	12	1	1	2	3	7	19	2.375	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	3	7	12	1	1	2	3	7	19	2.375	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	3	7	12	1	1	2	3	7	19	2.375	LOW	LOW
Password Autocomplete in Browser	3	1	4	4	12	7	5	4	7	23	35	4.375	MEDIUM	1	1	3	7	12	1	1	2	3	7	19	2.375	LOW	LOW
Content-Type Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	3	7	12	1	1	2	3	7	19	2.375	LOW	LOW
WEB-APP-C02																										
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	5	12	32	4	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C03																										
Directory Traversal	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	3	7	22	1	1	5	3	10	32	4	MEDIUM	MEDIUM
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	3	7	22	1	1	5	3	10	32	4	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Application Error Disclosure	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	3	7	22	1	1	5	3	10	32	4	MEDIUM	MEDIUM
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C04																										
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	3	7	22	1	1	5	3	10	32	4	MEDIUM	MEDIUM
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	3	7	22	1	1	5	3	10	32	4	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Application Error Disclosure	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	3	7	22	1	1	5	3	10	32	4	MEDIUM	MEDIUM
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C05																										
Inyección Remota de Comandos OS	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3.75	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C06																										
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3.75	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C07																										
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	5	12	32	4	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C08																										
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	5	12	32	4	MEDIUM	MEDIUM
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	5	12	32	4	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C09																										
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3.75	MEDIUM	MEDIUM
WEB-APP-C10																										
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	3	7	22	1	1	5	3	10	32	4	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C11																										
Directory Traversal	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3.75	MEDIUM	MEDIUM
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3.75	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Application Error Disclosure	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	2	3	7	27	3.375	MEDIUM	MEDIUM
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C12																										
Cross Site Scripting (Reflejada)	6	4	4	4	18	7	5	7	7	26	44	5.5	MEDIUM	6	7	7	7	27	1	1	5	3	10	37	4.625	MEDIUM	MEDIUM
Falla por Inyección SQL	6	4	4	4	18	7	5	7	7	26	44	5.5	MEDIUM	6	7	7	7	27	1	1	5	5	12	39	4.875	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2.875	LOW	LOW
Application Error Disclosure	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	6	7	7	7	27	1	1	5	3	10	37	4.625	MEDIUM	MEDIUM
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2.875	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2.875	LOW	LOW
Password Autocomplete in Browser	3	1	4	4	12	7	5	4	7	23	35	4.375	MEDIUM	3	3	1	7	14	1	1	2	3	7	21	2.625	LOW	LOW
WEB-APP-C13																										
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	5	7	24	1	1	5	3	10	34	4.25	MEDIUM	MEDIUM
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	5	7	24	1	1	5	3	10	34	4.25	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C14																										
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C15																										
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C16																										
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-C17																										
Inyección Remota de Comandos OS	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3.75	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW

Anexo B. Matriz de riesgos de las aplicaciones del departamento de Crédito y Cobranzas.

APLICACIONES WEB DEL DEPARTAMENTO DE CREDITO Y COBRANZAS [VULNERABILIDAD/CALCULO SEVERIDAD DEL RIESGO]	PROBABILIDAD												IMPACTO											SEVERIDAD DEL RIESGO
	Factores agentes de amenazas				Factores de vulnerabilidad								Factores de impacto técnico						Factores de impacto del negocio					
	Habilidades Técnicas	Motivación	Oportunidad	Tamaño	TOTAL	Facilidad de Descubrimiento	Facilidad de Explotación	Conciencia o Conocimiento	Detectores de Intrusión	TOTAL	PROBABILIDAD GENERAL	NIVEL PROBABILIDAD	Pérdida de la Confidencialidad	Pérdida de la Integridad	Pérdida de la Disponibilidad	Pérdida de la Auditabilidad	TOTAL	Daño Económico	Daño de Imagen	Incumplimiento	Violación a la Privacidad	TOTAL	IMPACTO GLOBAL	NIVEL DE IMPACTO	
WEB-APP-CC1																									
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	6	7	3	7	23	1	1	5	3	10	33	4,125	MEDIUM	MEDIUM
Falla pal Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	6	7	3	7	23	1	1	5	3	10	33	4,125	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
WEB-APP-CC2																									
Falla pal Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	6	7	3	7	23	1	1	5	3	10	33	4,125	MEDIUM	MEDIUM
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
WEB-APP-CC3																									
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-CC4																									
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-CC5																									
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-CC6																									
X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW

Anexo C. Matriz de riesgos de las aplicaciones del departamento de Maquinaria.

APLICACIONES WEB DEL DEPARTAMENTO DE MAQUINARIA [VULNERABILIDAD:CÁLCULO SEVERIDAD DEL RIESGO]	PROBABILIDAD												IMPACTO										SEVERIDAD DEL RIESGO
	Factores agentes de amenazas				Factores de vulnerabilidad								Factores de impacto técnico						Factores de impacto del negocio				
	Habilidades Técnicas	Motivación	Oportunidad	Tamaño	TOTAL	Facilidad de Descubrimiento	Facilidad de Explotación	Conciencia o Conocimiento	Detectores de Intrusión	TOTAL	TOTAL	PROBABILIDAD GENERAL	NIVEL PROBABILIDAD	Pérdida de la Confidencialidad	Pérdida de la Integridad	Pérdida de la Disponibilidad	Pérdida de la Auditibilidad	TOTAL	Daño Económico	Daño de Imagen	Incumplimiento	Violación a la Privacidad	TOTAL	IMPACTO GLOBAL	NIVEL DE IMPACTO	
WEB-APP-M01																										
/X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
/X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
Password Autocomplete in Browser	3	1	4	4	12	7	5	4	7	23	35	4,375	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
WEB-APP-M02																										
Falsa por Inyección iSQL	6	4	4	4	18	7	5	7	7	26	44	5,5	MEDIUM	7	7	7	7	28	1	1	5	3	10	38	4,75	MEDIUM	MEDIUM
/X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
/X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
WEB-APP-M03																										
Falsa por Inyección iSQL	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	5	5	1	7	18	1	1	5	3	10	28	3,5	MEDIUM	MEDIUM
/X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
/X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-M04																										
Falsa por Inyección iSQL	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	5	5	1	7	18	1	1	5	3	10	28	3,5	MEDIUM	MEDIUM
/X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
/X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Password Autocomplete in Browser	3	1	4	4	12	7	5	4	7	23	35	4,375	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-M05																										
Cross Site Scripting (Reflejado)	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	5	5	1	7	18	1	1	5	3	10	28	3,5	MEDIUM	MEDIUM
Falsa por Inyección iSQL	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	5	5	1	7	18	1	1	5	3	10	28	3,5	MEDIUM	MEDIUM
/X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	5	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
/X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-M06																										
/X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
/X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-M07																										
Falsa por Inyección iSQL	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	1	1	1	7	10	1	1	5	3	10	20	2,5	LOW	LOW
/X-Frame-Options Header Not Set	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
/X-Content-Type-Options Header Missing	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW

Anexo D. Matriz de riesgos de las aplicaciones del departamento de Repuestos.

APLICACIONES WEB DEL DEPARTAMENTO DE REPUESTOS TALLER HERMANOS CALCULO SEVERIDAD DEL RIESGO	PROBABILIDAD														IMPACTO											SEVERIDAD DEL RIESGO
	Factores agentes de amenazas				Factores de vulnerabilidad										Factores de impacto técnico						Factores de impacto del negocio					
	Habilidades Técnicas	Motivación	Oportunidad	Tamaño	TOTAL	Facilidad de Descubrimiento	Facilidad de Explotación	Conciencia o Conocimiento	Detectores de Intrusion	TOTAL	PROBABILIDAD GENERAL	NIVEL PROBABILIDAD	Pérdida de la Confidencialidad	Pérdida de la Integridad	Pérdida de la Disponibilidad	Pérdida de la Auditabilidad	TOTAL	Daño Económico	Daño de Imagen	Incumplimiento	Violación a la Privacidad	TOTAL	IMPACTO GLOBAL	NIVEL DE IMPACTO	
WEB-APP-001																										
X-Frame-Options Header Not Set (exposición clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled (explotación XSS)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing (explotación mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-002																										
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	7	7	5	7	26	1	1	5	3	10	36	4,5	MEDIUM	MEDIUM
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	7	7	5	7	26	1	1	5	3	10	36	4,5	MEDIUM	MEDIUM
X-Frame-Options Header Not Set (exposición clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled (explotación XSS)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing (explotación mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-003																										
Cross Site Scripting (Reflejada)	6	4	4	4	18	7	5	7	7	26	44	5,5	MEDIUM	6	5	5	7	23	1	1	5	3	10	33	4,125	MEDIUM	MEDIUM
Falla por Inyección SQL	6	4	4	4	18	7	5	7	7	26	44	5,5	MEDIUM	6	5	5	7	23	1	1	5	3	10	33	4,125	MEDIUM	MEDIUM
X-Frame-Options Header Not Set (exposición clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	1	1	7	12	1	1	2	3	7	19	2,375	LOW	LOW
Web Browser XSS Protection Not Enabled (explotación XSS)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	1	1	7	12	1	1	2	3	7	19	2,375	LOW	LOW
X-Content-Type-Options Header Missing (explotación mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	1	1	7	12	1	1	2	3	7	19	2,375	LOW	LOW
WEB-APP-004																										
Cross Site Scripting (Reflejada)	6	4	4	4	18	7	5	7	7	26	44	5,5	MEDIUM	6	7	5	7	25	1	1	5	3	10	35	4,375	MEDIUM	MEDIUM
Falla por Inyección SQL	6	4	4	4	18	7	5	7	7	26	44	5,5	MEDIUM	6	7	5	7	25	1	1	5	3	10	35	4,375	MEDIUM	MEDIUM
X-Frame-Options Header Not Set (exposición clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	1	1	7	12	1	1	2	3	7	19	2,375	LOW	LOW
Web Browser XSS Protection Not Enabled (explotación XSS)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	1	1	7	12	1	1	2	3	7	19	2,375	LOW	LOW
X-Content-Type-Options Header Missing (explotación mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	1	1	7	12	1	1	2	3	7	19	2,375	LOW	LOW
WEB-APP-005																										
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3,75	MEDIUM	MEDIUM
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3,75	MEDIUM	MEDIUM
X-Frame-Options Header Not Set (exposición clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	5	3	10	20	2,5	LOW	LOW
Web Browser XSS Protection Not Enabled (explotación XSS)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing (explotación mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-006																										
Falla por Inyección SQL	6	1	4	4	15	7	5	7	7	26	41	5,125	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3,75	MEDIUM	MEDIUM
X-Frame-Options Header Not Set (exposición clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	5	3	10	20	2,5	LOW	LOW
Web Browser XSS Protection Not Enabled (explotación XSS)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing (explotación mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APP-007																										
X-Frame-Options Header Not Set (exposición clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled (explotación XSS)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	1	1	7	12	1	1	2	3	7	19	2,375	LOW	LOW
X-Content-Type-Options Header Missing (explotación mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4,75	MEDIUM	3	1	1	7	12	1	1	2	3	7	19	2,375	LOW	LOW
WEB-APP-008																										
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	44	5,5	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3,75	MEDIUM	MEDIUM
Falla por Inyección SQL	6	1	4	4	18	7	5	7	7	26	44	5,5	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3,75	MEDIUM	MEDIUM
X-Frame-Options Header Not Set (exposición clickjacking)	6	1	4	4	18	7	5	7	7	26	44	5,5	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3,75	MEDIUM	MEDIUM
Web Browser XSS Protection Not Enabled (explotación XSS)	6	1	4	4	18	7	5	4	7	23	41	5,125	MEDIUM	3	1	1	7	12	1	1	2	3	7	19	2,375	LOW	LOW
X-Content-Type-Options Header Missing (explotación mime-sniffing)	6	1	4	4	18	7	5	4	7	23	41	5,125	MEDIUM	3	1	1	7	12	1	1	2	3	7	19	2,375	LOW	LOW
WEB-APP-009																										
X-Frame-Options Header Not Set (exposición clickjacking)	6	1	4	4	18	7	5	4	7	23	41	5,125	MEDIUM	3	1	3	7	14	1	1	2	3	7	21	2,625	LOW	LOW
Web Browser XSS Protection Not Enabled (explotación XSS)	6	1	4	4	18	7	5	4	7	23	41	5,125	MEDIUM	3	1	3	7	14	1	1	2	3	7	21	2,625	LOW	LOW
X-Content-Type-Options Header Missing (explotación mime-sniffing)	6	1	4	4	18	7	5	4	7	23	41	5,125	MEDIUM	3	1	3	7	14	1	1	2	3	7	21	2,625	LOW	LOW
Cross-Domain JavaScript Source File Inclusion	6	1	4	4	18	7	5	4	7	23	41	5,125	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW

Anexo E. Matriz de riesgos de las aplicaciones del departamento de Servicios.

APLICACIONES WEB DEL DEPARTAMENTO DE SERVICIOS [VULNERABILIDAD/CALCULO SEVERIDAD DEL RIESGO]	PROBABILIDAD												IMPACTO									SEVERIDAD DEL RIESGO
	Factores agentes de amenazas					Factores de vulnerabilidad							Factores de impacto técnico						Factores de impacto del negocio					
	Habilidades Técnicas	Motivación	Oportunidad	Tamaño	TOTAL	Facilidad de Descubrimiento	Facilidad de Explotación	Conciencia o Conocimiento	Detectores de Intrusión	TOTAL	PROBABILIDAD GENERAL	NIVEL PROBABILIDAD	Pérdida de la Confidencialidad	Pérdida de la Integridad	Pérdida de la Disponibilidad	Pérdida de la Auditabilidad	TOTAL	Daño Económico	Daño de Imagen	Incumplimiento	Violación a la Privacidad	TOTAL	IMPACTO GLOBAL	NIVEL DE IMPACTO	
WEB-APP-001																									
X-Frame-Options Header Not Set (exposicion clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled (exposicion XSS)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing (exposicion mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-002																									
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	7	5	1	7	20	1	1	5	3	10	30	3.75	MEDIUM	MEDIUM
X-Frame-Options Header Not Set (exposicion clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled (exposicion XSS)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing (exposicion mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Password Autocomplete in Browser (exposicion a información sensible)	3	1	4	4	12	7	5	4	7	23	35	4.375	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Private IP Disclosure (exposicion a información sensible)	3	1	4	4	12	7	5	4	7	23	35	4.375	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-003																									
Cross Site Scripting (Reflejada)	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	6	5	1	7	25	1	1	5	5	12	37	4.625	MEDIUM	MEDIUM
Falla por Injeccion SQL	6	1	4	4	15	7	5	7	7	26	41	5.125	MEDIUM	6	5	1	7	25	1	1	5	5	12	37	4.625	MEDIUM	MEDIUM
WEB-APP-004																									
X-Frame-Options Header Not Set (exposicion clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled (exposicion XSS)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing (exposicion mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-005																									
X-Frame-Options Header Not Set (exposicion clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
Web Browser XSS Protection Not Enabled (exposicion XSS)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
X-Content-Type-Options Header Missing (exposicion mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2.125	LOW	LOW
WEB-APP-006																									
X-Frame-Options Header Not Set (exposicion clickjacking)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	3	7	12	1	1	5	3	10	22	2.75	LOW	LOW
Web Browser XSS Protection Not Enabled (exposicion XSS)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	3	7	12	1	1	5	3	10	22	2.75	LOW	LOW
X-Content-Type-Options Header Missing (exposicion mime-sniffing)	6	1	4	4	15	7	5	4	7	23	38	4.75	MEDIUM	1	1	3	7	12	1	1	5	3	10	22	2.75	LOW	LOW

Anexo F. Matriz de riesgos de las aplicaciones del departamento de Sistemas.

APLICACIONES WEB DEL DEPARTAMENTO DE SISTEMAS [VULNERABILIDAD/CALCULO SEVERIDAD DEL RIESGO]	PROBABILIDAD											IMPACTO											SEVERIDAD DEL RIESGO
	Factores agentes de amenazas				Factores de vulnerabilidad							Factores de impacto técnico						Factores de impacto del negocio					
	Habilidades Técnicas	Motivación	Oportunidad	Tamaño	Facilidad de Descubrimiento	Facilidad de Explotación	Conciencia o Conocimiento	Detectores de Intrusión	TOTAL	PROBABILIDAD GENERAL	NIVEL PROBABILIDAD	Pérdida de la Confidencialidad	Pérdida de la Integridad	Pérdida de la Disponibilidad	Pérdida de la Auditabilidad	TOTAL	Daño Económico	Daño de Imagen	Incumplimiento	Violacion a la Privacidad	TOTAL	IMPACTO GLOBAL	NIVEL DE IMPACTO	
WEB-APK#1																								
X-Frame-Options Header Not Set (exposicion clickjacking)	6	1	4	4	15	7	5	7	26	41	5,125	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled (exposicion XSS)	6	1	4	4	15	7	5	7	26	41	5,125	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing (exposicion mime-sniffing)																								
Content-Type Header Missing (exposicion mime-sniffing)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APK#2																								
X-Frame-Options Header Not Set (exposicion clickjacking)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled (exposicion XSS)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing (exposicion mime-sniffing)																								
Content-Type Header Missing (exposicion mime-sniffing)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APK#3																								
Falla por Inyeccion SQL	6	1	4	4	15	7	5	7	26	41	5,125	MEDIUM	7	5	1	7	20	1	1	5	3	10	38	3,75	MEDIUM	MEDIUM
X-Frame-Options Header Not Set (exposicion clickjacking)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
Web Browser XSS Protection Not Enabled (exposicion XSS)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
X-Content-Type-Options Header Missing (exposicion mime-sniffing)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	1	1	1	7	10	1	1	2	3	7	17	2,125	LOW	LOW
WEB-APK#4																								
Directory Traversal (exposicion a informacion sensible)	6	4	4	4	18	7	5	7	26	44	5,5	MEDIUM	7	5	7	7	26	1	1	5	5	5	38	4,75	MEDIUM	MEDIUM
Cross Site Scripting (Reflejada)	6	4	4	4	18	7	5	7	26	44	5,5	MEDIUM	7	5	7	7	26	1	1	5	5	5	38	4,75	MEDIUM	MEDIUM
Falla por Inyeccion SQL	6	4	4	4	18	7	5	7	26	44	5,5	MEDIUM	7	5	7	7	26	1	1	5	5	5	38	4,75	MEDIUM	MEDIUM
X-Frame-Options Header Not Set (exposicion clickjacking)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	1	1	3	7	12	1	1	2	5	9	21	2,625	LOW	LOW
Web Browser XSS Protection Not Enabled (exposicion XSS)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	1	1	3	7	12	1	1	2	5	9	21	2,625	LOW	LOW
X-Content-Type-Options Header Missing (exposicion mime-sniffing)																								
Private IP Disclosure (exposicion a informacion sensible)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	1	1	3	7	12	1	1	2	5	9	21	2,625	LOW	LOW
Private IP Disclosure (exposicion a informacion sensible)	3	1	4	4	12	7	5	4	23	35	4,375	MEDIUM	1	1	3	7	12	1	1	2	5	9	21	2,625	LOW	LOW
WEB-APK#5																								
Cross Site Scripting (Reflejada)	6	4	4	4	18	7	5	7	26	44	5,5	MEDIUM	6	7	7	7	27	1	1	5	5	12	39	4,875	MEDIUM	MEDIUM
X-Frame-Options Header Not Set (exposicion clickjacking)	6	4	4	4	18	7	5	4	23	41	5,125	MEDIUM	6	7	7	7	27	1	1	2	3	7	34	4,25	MEDIUM	MEDIUM
Cookie No HttpOnly Flag (exposicion a inform acion sensible)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
Web Browser XSS Protection Not Enabled (exposicion XSS)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	6	7	7	7	27	1	1	2	3	7	34	4,25	MEDIUM	MEDIUM
Password Autocomplete in Browser	3	1	4	4	12	7	5	4	23	35	4,375	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
X-Content-Type-Options Header Missing (exposicion mime-sniffing)																								
Private IP Disclosure (exposicion a informacion sensible)	6	1	4	4	15	7	5	4	23	38	4,75	MEDIUM	6	7	7	7	27	1	1	2	3	7	34	4,25	MEDIUM	MEDIUM
Private IP Disclosure (exposicion a informacion sensible)	3	1	4	4	12	7	5	4	23	35	4,375	MEDIUM	3	3	3	7	16	1	1	2	3	7	23	2,875	LOW	LOW
Exposicion de consola del servidor web	6	4	4	4	18	7	5	7	26	44	5,5	MEDIUM	6	7	7	7	27	1	1	5	3	10	37	4,625	MEDIUM	MEDIUM

Anexo G. Documentación para mitigación de riesgos de inyección (A1).

Nombre del riesgo

A1 – Inyección

Vulnerabilidad asociada Inyección SQL.

Descripción del riesgo

Ejecución de consultas o comandos con datos maliciosos, permitiendo al atacante poder acceder, alterar o perder información del repositorio o equipo.

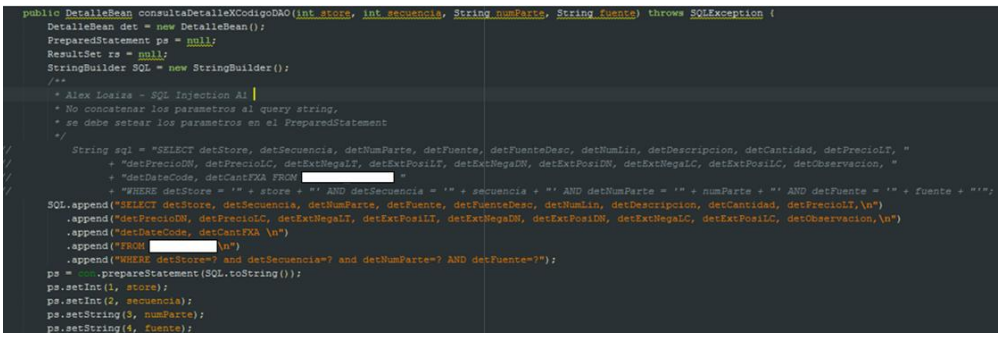
Severidad del riesgo

Media

Probabilidad General: 5.12 (Media)

Impacto Global: 4.5 (Media)

Planes de acción



```
public DetalleBean consultaDetalleXCodigoDAO(int store, int secuencia, String numParte, String fuente) throws SQLException {
    DetalleBean det = new DetalleBean();
    PreparedStatement ps = null;
    ResultSet rs = null;
    StringBuilder SQL = new StringBuilder();
    /**
     * Alex Loaiza - SQL Injection A1
     * No concatenar los parametros al query string,
     * se debe setear los parametros en el PreparedStatement
     */
//      String sql = "SELECT detStore, detSecuencia, detNumParte, detFuente, detFuenteDesc, detNumLin, detDescripcion, detCantidad, detPrecioLT, "
//              + "detPrecioDN, detPrecioLC, detExtNegaLT, detExtPosiLT, detExtNegaDN, detExtPosiDN, detExtNegaLC, detExtPosiLC, detObservacion, "
//              + "detDateCode, detCantFXA FROM [redacted] "
//              + "WHERE detStore = '" + store + "' AND detSecuencia = '" + secuencia + "' AND detNumParte = '" + numParte + "' AND detFuente = '" + fuente + "'";
    SQL.append("SELECT detStore, detSecuencia, detNumParte, detFuente, detFuenteDesc, detNumLin, detDescripcion, detCantidad, detPrecioLT,\n")
       .append("detPrecioDN, detPrecioLC, detExtNegaLT, detExtPosiLT, detExtNegaDN, detExtPosiDN, detExtNegaLC, detExtPosiLC, detObservacion,\n")
       .append("detDateCode, detCantFXA \n")
       .append("FROM [redacted] \n")
       .append("WHERE detStore=? and detSecuencia=? and detNumParte=? AND detFuente=?");
    ps = con.prepareStatement(SQL.toString());
    ps.setInt(1, store);
    ps.setInt(2, secuencia);
    ps.setString(3, numParte);
    ps.setString(4, fuente);
```

- Se comenta sentencia SQL anterior y se describe una nueva en donde los parámetros ya no son concatenados, sino enviados a través de métodos proporcionados por la clase PreparedStatement de Java.
- Esto debe ser practicado por los desarrolladores del departamento y mejor aún usar procedimientos almacenados para todo tipo de transacción con las bases de datos.
- Lo recomendable es enviar los parámetros limpios es decir, libre de caracteres maliciosos y esto debe ser validado en el controlador (servlet, action, backing bean, etc.). Uso de herramienta ESAPI para validar la entrada de datos.
- Revisión de código antes del pase a producción.
- Reuniones entre desarrolladores para disipar dudas.
- Capacitación en temas de desarrollo seguro.

Plan de contingencia

- Contar con copias de seguridad probadas y verificadas de las bases de datos afectadas.
- Restablecer servicios en equipos de backup en caso de que los equipos principales se vean afectados.
- Determinar opción afectada y de ser posible evitar el acceso a ella hasta tener una solución real al inconveniente.

Responsable(s)

Jefe de software.

Desarrollador encargado de la aplicación.

Key User del departamento.

Recursos

Fuentes actualizadas de la aplicación web.

Tiempo que conllevaría la modificación y pruebas de los cambios realizados.

Documentación de los cambios.

Anexo H. Documentación para mitigación de riesgos de secuencia de comandos en sitios cruzados XSS (A3).

Nombre del riesgo

A3 – Secuencia de comandos en sitios cruzados

Vulnerabilidad asociada XSS Reflejada.

Descripción del riesgo

Permite la ejecución de código JavaScript o lenguaje similar para robar información delicada, secuestrar sesiones y comprometer el navegador.

Severidad del riesgo

Media

Probabilidad General 5.12 (Media)

Impacto Global 4.50 (Media)

Planes de acción

```
  /**
   * Alex Loaiza - XSS Reflejado A3 Validamos la entrada de
   * información para luego ser procesada.
   */
  String document2, cantida2, notas2;
String document2 = request.getParameter("document2");
String cantida2 = request.getParameter("cantidad2");
String notas2 = request.getParameter("notas2");
String tipEmb2 = ingresoGuiaRemisionForm.getOcultoTiEmbalaje().trim() + "#" + ingresoGuiaRemisionForm.getOcultoIdEmbalaje().trim().substring(5);
String embnote = ingresoGuiaRemisionForm.getOcultoNtEmbalaje().trim();

  document2 = ESAPI.validator().getValidInput("Ingreso Guia Remision",
          request.getParameter("document2"), "AlphaNumeric", 50, true);

  cantida2 = ESAPI.validator().getValidInput("Ingreso Guia Remision",
          request.getParameter("cantidad2"), "Numeric", 10, false);

  notas2 = ESAPI.validator().getValidInput("Ingreso Guia Remision",
          request.getParameter("notas2"), "AlphaNumeric", 500, true);
```

- Los parámetros enviados al controlador son verificados con diferentes listas blancas en las que se valida el tipo, formato y tamaño del dato enviado.
- Establecer listas blancas comunes a las diferentes entradas de las aplicaciones web de la plataforma intranet.
- Escapar los datos ingresados y mostrarlos tal y como los introdujo el usuario.
- Uso del método ValidInput de la herramienta ESAPI para el control de entradas.
- Monitoreo de entradas maliciosas mediante el empleo de Logs.
- Actualizaciones de seguridad de navegadores web usados en la compañía.
- Revisión de código antes del pase a producción.
- Reuniones entre desarrolladores para disipar dudas.
- Capacitación en temas de desarrollo seguro.

Plan de contingencia

- Restablecer servicios en equipos de backup en caso de que los equipos principales se vean afectados.
- Determinar opción afectada y de ser posible evitar el acceso a ella hasta tener una solución real al inconveniente.

Responsable(s)

Jefe de software.

Desarrollador encargado de la aplicación.

Key User del departamento.

Recursos

Fuentes actualizadas de la aplicación web.

Tiempo que conllevaría la modificación y pruebas de los cambios realizados.

Documentación de los cambios.

Anexo I. Documentación para mitigación de riesgos de referencia directa insegura de objetos (A4).

Nombre del riesgo

A4 – Referencia directa insegura a objetos

Vulnerabilidad asociada Directory Traversal.

Descripción del riesgo

Permite acceder a directorios y archivos que residen fuera del directorio raíz de la aplicación web comprometiendo información del equipo, servidor de aplicaciones y la red.

Severidad del riesgo

Media

Probabilidad General 5.50 (Media)

Impacto Global 4.75 (Media)

Planes de acción

```
/**
 * Alex Loaiza
 * Evitamos colocar la direccion absoluta del archivo a visualizar, se modificó servlet
 * para que aceptara campo que determina el repositorio donde debe extraer el
 * archivo a visualizar.
 */
  estr.setUrlArchivo("PdfViewer?name="+f.getName()+"&openFile="+ f.getAbsolutePath());
estr.setUrlArchivo("PdfViewer?name="+f.getName()+"&repository=4");
```

```
/**
 * Alex Loaiza
 * Validamos que el nombre del archivo a descargar este
 * dentro de la lista blanca de extensiones.
 */
ESAPI.validator().getValidInput(PdfViewer.class.getName(),
        name, "UnicodeStringPunctuationMultilineURL", 500, true);

openFile = URLDecoder.decode(fileServer + File.separatorChar + name,
                "iso-8859-1");
```

- Limitar el acceso de archivos por extensión (.xls, .pdf, .doc, etc.)
- Limitar el acceso a los archivos según repositorio habilitado.
- Los parámetros enviados al controlador son verificados con diferentes listas blancas en las que se valida el tipo, formato y tamaño del dato enviado.
- Establecer listas blancas comunes a las diferentes entradas de las aplicaciones web de la plataforma intranet

- Uso del método ValidInput de la herramienta ESAPI para el control de entradas.
- Monitoreo de entradas maliciosas mediante el empleo de Logs.
- Revisión de código antes del pase a producción.
- Reuniones entre desarrolladores para disipar dudas.
- Capacitación en temas de desarrollo seguro.

Plan de contingencia

- Restablecer servicios en equipos de backup en caso de que los equipos principales se vean afectados.
- Determinar opción afectada y de ser posible evitar el acceso a ella hasta tener una solución real al inconveniente.

Responsable(s)

Jefe de software.

Desarrollador encargado de la aplicación.

Recursos

Fuentes actualizadas de la aplicación web.

Tiempo que conllevaría la modificación y pruebas de los cambios realizados.

Documentación de los cambios.

Anexo J. Documentación para mitigación de riesgos de configuración de seguridad incorrecta (A5).

Nombre del riesgo

A5 – Configuración de Seguridad Incorrecta

Vulnerabilidad asociada Application Error Disclosure.

Descripción del riesgo

Se despliega información sensible de la aplicación trazas de código o errores que no son correctamente manejados cuando se presente alguna excepción.

Severidad del riesgo

Media

Probabilidad General 5.12 (Media)

Impacto Global 3.37 (Media)

Planes de acción

```
} catch (NumberFormatException | ValidationException | IntrusionException | ServletException | IOException e) {
    /**
     * Alex Loaiza - Manejo de información sensible (errores) A6 Manejo
     * de errores y evitar así que la página, por algún inconveniente,
     * tenga que mostrar el error ocurrido al usuario (navegador).
     */
    logger.error(e, e);
    try {
        response.sendRedirect(request.getContextPath() + "/errors/error.jsp");
    } catch (Exception ee) {
        logger.error(ee, ee);
    }
}
```

- Uso de sentencias {try catch} para manejar los errores y presentarlos en el Log del servidor.
- El error que se presente en una petición no debe mostrarse al usuario, más solo se debe gestionar el error con páginas genéricas.
- Empleo de funciones de Logging para el manejo correcto y exacto de una excepción.
- Monitoreo de entradas maliciosas.
- Revisión de código antes del pase a producción.
- Reuniones entre desarrolladores para disipar dudas.
- Capacitación en temas de desarrollo seguro.

Plan de contingencia

- Restablecer servicios en equipos de backup en caso de que los equipos principales se vean afectados.

Responsable(s)

Desarrollador encargado de la aplicación.

Recursos

Fuentes actualizadas de la aplicación web.

Tiempo que conllevaría la modificación y pruebas de los cambios realizados.

Documentación de los cambios.