ANÁLISIS DE RIESGOS DE LAS APLICACIONES WEB DE LA SUPERINTENDENCIA DE BANCOS Y SEGUROS, UTILIZANDO LAS RECOMENDACIONES TOP TEN DE OWASP.

Lenin Salgado[1], Mario Ron[2], Fernando Solis[3]

1 Universidad de las Fuerzas Armadas (ESPE), Ecuador, leninsy@gmail.com
2 Universidad de las Fuerzas Armadas (ESPE), Ecuador, mbron@espe.edu.ec
3 Universidad de las Fuerzas Armadas (ESPE), Ecuador, efsolis@espe.edu.ec

RESUMEN

En la actualidad las aplicaciones web se han vuelto indispensables para el manejo de la información en una organización, convirtiéndose en una herramienta que permite al usuario acceder y utilizar un sistema informático a través de internet mediante un navegador web, permitiendo el acceso a la información desde cualquier parte del mundo. La Superintendencia de Bancos y Seguros al ser una institución Pública se ha visto obligada a la adopción de estándares abiertos y software libre para automatizar sus procesos, y ha desarrollado aplicaciones web utilizando la plataforma Java Enterprise Edition (JEE) sin embargo no se ha aplicado ningún tipo de estándar o buenas prácticas en el aseguramiento del aplicativo. El presente proyecto tiene como objetivo el análisis de riesgos de las aplicaciones web utilizando las recomendaciones OWASP Top 10 – 2010 para descubrir las vulnerabilidades que se presenta durante el desarrollo de un software y estimar el riesgo asociado para el negocio. A partir de los resultados obtenidos donde se identificaron la ocurrencia de almacenamiento criptográfico inseguro y protección insuficiente en la capa de transporte se realizó una propuesta de buenas prácticas para asegurar las aplicaciones, corregir los riesgos detectados y asegurar el proceso de desarrollo de nuevas funcionalidades y existentes.

Palabras Clave: OWASP, Seguridad, Riesgo, Desarrollo y Aplicaciones Web.

ABSTRACT

Nowadays Web applications have become essential to the management of information in an organization, making it a tool that allows users to access and use a computer system via the Internet using a web browser, allowing access to information from anywhere in the world. The Superintendencia de Bancos y Seguro is a public institution has been forced to adopt open standards and free software to automate their processes, and developed web applications using Java Platform, Enterprise Edition (JEE) but has not been applied any standard or good practices in ensuring the application. This project aims to risk analysis of web applications using the recommendations OWASP Top 10 - 2010 to discover vulnerabilities that occurs during software development and estimate the associated risk to the business. From the results obtained where the occurrence of insecure cryptographic storage and insufficient protection in the transport layer of a proposal identified good practices to ensure applications made, correct the identified risks and ensure the process of developing new features and existing

KeyWords: OWASP, Security, Risk, Development and Web Applications.

1. INTRODUCCIÓN

La mayoría de los problemas de seguridad en los sitios web se encuentran a nivel aplicación y que son el resultado de escritura defectuosa de código, debemos entender que programar aplicaciones web seguras no es una tarea fácil, ya que requiere por parte del programador, no únicamente mostrar atención en cumplir con el objetivo funcional básico de la aplicación, sino una concepción general de los riesgos que puede correr la información contenida, solicitada y recibida por el sistema. En la actualidad, aunque existen muchas publicaciones que permiten formar un criterio sobre el tema, no existen acuerdos básicos sobre lo que se debe o no se debe hacer, y lo que en algunas publicaciones se recomienda, en otras es atacado. Sin embargo, en lo sustancial sí existen algunas recomendaciones que son generales y serán las que describamos en este artículo. (UNAM-CERT, 2011)

The Open Web Application Security Proyect (OWASP) es un proyecto de código abierto de seguridad en aplicaciones web y determina las causas que hacen un software inseguro. Ofrece un Top 10 sobre los riesgos más importantes en aplicaciones web con el objetivo principal de educar a desarrolladores, diseñadores, arquitectos, gerentes y organizaciones.

OWASP Top 10 se enfoca en la identificación de los riesgos más serios para un amplio espectro de organizaciones. Para cada uno de estos riesgos, proveemos información genérica acerca de la probabilidad y el impacto técnico usando un esquema simple de calificación, que está basado en la Metodología de Evaluación de Riesgos OWASP. (OWASP, 2014).

Fueron analizados el Sistema de Población de Identificaciones – SOCI, Sistema de Auditorias para la Prevención de Lavado de Activos – SAPLA y Sistema para Otorgar Credenciales a Intermediarios de Seguros – SOCI. Durante el desarrollo del estudio se identificaron los riesgos más comunes en el desarrollo de las aplicaciones Web detectando que existen dos riesgos bien claros los cuales son: Almacenamiento Criptográfico Inseguro y Protección Insuficiente en la Capa de Transporte.

Frente a estas vulnerabilidades encontradas en el análisis de las aplicaciones web se propuso que para el riesgo de Almacenamiento Criptográfico Inseguro se debe identificar los datos más sensibles y que requieren ser cifrados utilizando un algoritmo de encriptación seguro como el Advanced Encryption Standard (AES). (OWASP, 2012), y para la Protección Insuficiente en la Capa de Transporte utilizar un certificado Secure Sockets Layer (SSL) lo cual establecerá comunicaciones seguras en la red.

Este artículo está organizado de la siguiente manera: la sección 2 presenta la metodología de Owasp Top 10 – 2010, la sección 3 muestra el análisis de riesgos en base a las recomendaciones Top Ten de OWASP en aplicaciones web, la sección 4 presenta los riesgos de mayor ocurrencia, la sección 5 presenta trabajos relacionados a nuestro proyecto y finalmente la sección 6 indica las conclusiones y trabajos a futuros sobre la base de los resultados obtenidos.

2. METODOLOGÍA

En esta sección se expone los fundamentos teóricos para el análisis de las aplicaciones web con arquitectura JEE de la Superintendencia de Bancos y Seguros, que sustentaron cada uno de los factores que conforman la probabilidad de un riesgo. El Top 10 de OWASP se utilizó para determinar el riesgo en la institución asociado al negocio y empezar con un programa de buenas prácticas para el desarrollo de sus aplicativos.

El Top 10 de OWASP permitió a este proyecto analizar y descubrir las vulnerabilidades de las aplicaciones web. Es una lista de los diez riesgos más importantes en aplicaciones basados en un esquema simple de calificación como lo muestra la ilustración 1.

Agentes De Amenaza	Vectores De Ataque	Prevalencia de Debilidades	Detectabilidad de Debilidades	Impacto Técnico	Impacto Al Negocio
?	Fácil	Difundido	Fácil	Severo	?
	Medio	Común	Medio	Moderado	
	Difícil	Poco Común	Difícil	Menor	

Ilustración 1: Esquema simple de calificación de riesgos. Fuente: (OWASP, 2014).

A continuación se lista el Top 10 de OWASP. (OWASP, 2014)

- R1: Inyección.
- R2: Secuencia de comandos en sitios cruzados (XSS).
- R3: Pérdida de Autenticación y Gestión de Sesiones.
- R4: Referencia Directa Insegura a Objetos.
- R5: Falsificación de Peticiones en Sitios Cruzados.
- R6: Configuración Defectuosa de seguridad.
- R7: Almacenamiento Criptográfico Inseguro.
- R8: Falla de Restricción de Acceso a URL.
- R9 Protección Insuficiente en la Capa de Transporte.
- R10: Redirecciones y Reenvíos no validados.

3. DISEÑO E IMPLEMENTACIÓN

3.1 ANALISIS DE RIESGOS DE APLICACIONES WEB

A continuación se presentan tres de los sistemas más importantes y robustos de la Superintendencia de Bancos y Seguros basados en la arquitectura JEE, los mismos que serán objeto de análisis del presente proyecto por ser los más complejos y que manejan gran cantidad de información.

- Sistema de Población de Identificaciones (SPI)
- Sistema de Auditoría de Prevención de Lavado de Activos (SAPLA)
- Sistema para otorgar Credenciales a Intermediarios de Seguros (SOCI)

La Tabla 1 muestra el análisis de los sistemas antes mencionados mediante el OWASP Top 10 – 2010.

Tabla 1: Análisis de Sistemas Web. Fuente: (Salgado, 2014)

	R1	R2	R3	R4	R5	R6	R7	R8	R9
SAPLA	Se utilizan consultas estáticas y variables parametrizadas	Validan los datos de entrada y las peticiones http para cada sesión	Si los usuarios no cierran las sesiones, éstas caducan a los 30 minutos de inactividad	Se definen los permisos sobre los menús o perfiles que tiene cada usuario	Cada enlace, sesión y formulario, contiene un token de seguridad no predecible para los usuarios	Se trabaja con un servidor web Jboss el cual mantiene subido un firewall que no permite el acceso a la consola de administración mediante su IP.	No utilizan ningún algoritmo para encriptar la información	Se emplean mecanismos de seguridad para el acceso a las páginas, mediante la autenticación y autorización	Las redirecciones y los reenvíos están validados
SPI									
SOCI							Se utiliza el algoritmo hash MD5		

A partir de la comparación y el análisis realizado se identificaron fundamentalmente la ocurrencia de los siguientes riesgos:

- R7: Almacenamiento Criptográfico Inseguro.
- R9: Protección Insuficiente en la capa de Transporte.

El sistema SOCI utiliza el algoritmo hash MD5 (*Message-Digest Algorithm 5*) como se muestra en la Ilustración 2, es un algoritmo de reducción criptográfico el cual es uno de los más utilizados hoy en día sin embargo es un algoritmo considerado débil. (OWASP, 2014). En el caso de los sistemas SPI y SAPLA no se utiliza ningún algoritmo para encriptar los datos, lo que constituye una vulnerabilidad considerable, pues si se accede a los datos, se encontrarían en texto plano, por lo que sería más fácil para los atacantes realizar un ataque exitoso a la aplicación.

MD5.java

```
package ec.gov.sbs.soci.general;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

import sun.misc.BASE64Encoder;

public class MD5 {

    public static String getEncriptedPassword(String dbPassword) throws NoSuchAlgorithmException  {
        byte[] digest = null;
        byte[] buffer = dbPassword.getBytes();
        try {
            MessageDigest messageDigest = MessageDigest.getInstance("MD5");
            messageDigest.reset();
            messageDigest.update(buffer);
            digest = messageDigest.digest();
            } catch (NoSuchAlgorithmException ex) {
                System.out.println("Error creando Digest");
            }
            return new BASE64Encoder().encode(digest).trim();
    }
}
```

Ilustración 2: Clase Java MD5. Fuente: (Salgado, 2014).

El riesgo protección insuficiente en la capa de transporte está presente en las tres aplicaciones como se muestra en la Ilustración 3, para proteger el tráfico de la autenticación no se utiliza un protocolo criptográfico como SSL (Secure Sockets Layer, capa de conexión segura), el cual proporciona privacidad entre dos aplicaciones de comunicaciones utilizando HTTPS. (IBM) Tampoco se utiliza para encriptar los canales de comunicación de datos, los servicios y los recursos. La utilización de este protocolo facilitaría la protección, confidencialidad y autenticación de la información transmitida.

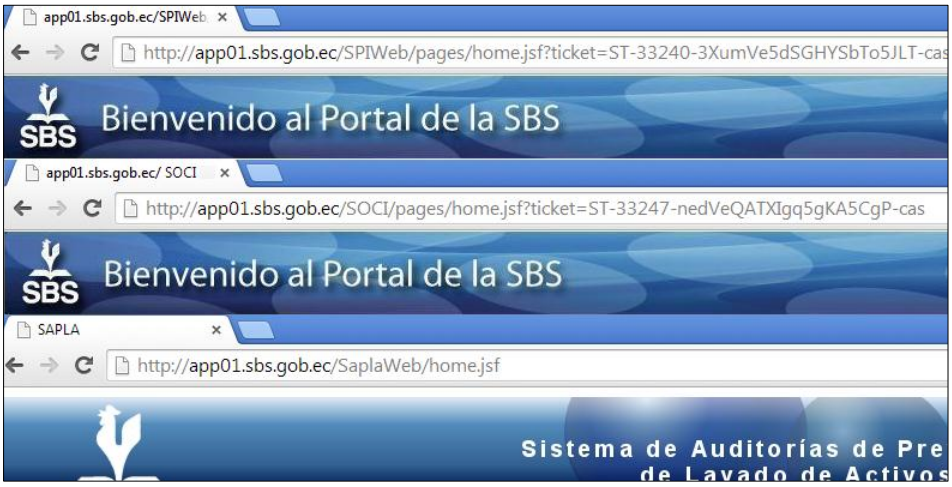

Ilustración 3: Sistemas SBS. Fuente: (Salgado, 2014).

En la Ilustración 4 se puede apreciar la probabilidad de ataque por riesgo asociado a cada sistema analizado en el presente proyecto, en base a la siguiente escala:

- De 0 – 1 BAJA
- De 1 – 2 MEDIA
- De 2 – 3 ALTA

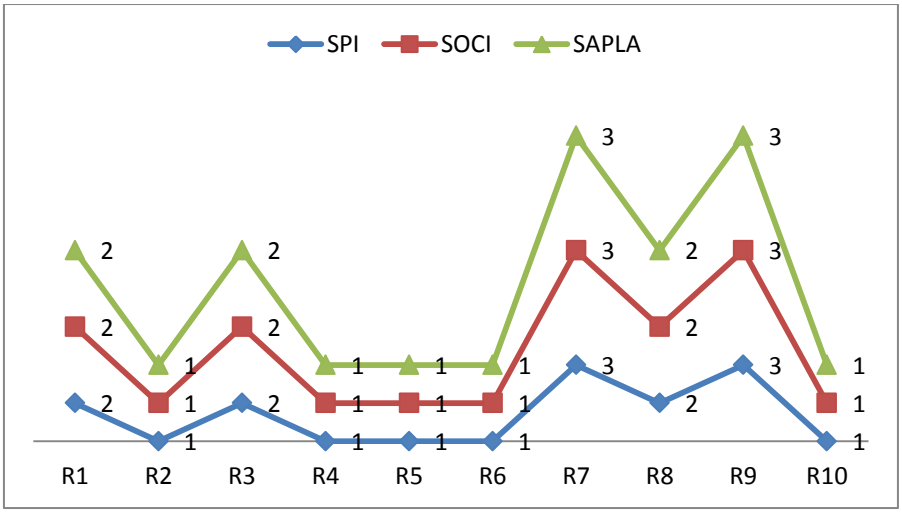

Ilustración 4: Probabilidad de riesgos por sistema. Fuente (Salgado, 2014).

3.2 EVALUACIÓN DE RIESGOS

En la ilustración 3 se muestra los 2 riesgos identificados en base al Top 10 de OWASP. Se incluye tres factores de probabilidad para cada debilidad (explotación, prevalencia, detección), dos factores de impacto (impacto técnico e impacto en el negocio) y los vectores de amenaza. A continuación se describe cada uno de los elementos.

	AGENTES DE AMENAZA	VECOTORES DE ATAQUE	DEFICIENCIAS DE SEGURIDAD		IMPACTOS TECNICOS	IMPACTOS EN EL NEGOCIO
	----------	Explotación	Prevalencia	Detección	Impacto	----------
R 7	Usuarios externos, internos y administradores	DIFICIL	POCO COMÚN	DIFICIL	SEVERO	Datos perdidos, responsabilidad legal si esos datos son expuestos. Reputación del negocio
R 9	Alguien que pueda capturar el tráfico de red de sus usuarios	DIFICIL	COMÚN	FACIL	MODERADO	Valor del negocio de la información expuesta en los canales de comunicación.

Ilustración 5: Identificación de Riesgos. Fuente: (Salgado, 2014).

- Agentes de amenazas: son los que realizan el ataque a la aplicación.
- Vectores de ataques: son las diferentes formas en las que se puede llevar a cabo el ataque. Pueden ser cadenas de texto que explotan la sintaxis del intérprete atacado.
- Prevalencia de debilidades: es la frecuencia con la que se manifiesta el riesgo analizado.
- Detectabilidad: es la probabilidad o facilidad con que se puede detectar un riesgo.
- Impacto técnico: es el la magnitud del daño que ocasiona el riesgo a la aplicación.
- Impacto en el negocio: es el daño que sufre el negocio con el éxito del riesgo. (OWASP, 2014).

A cada uno de los aspectos se le asignó una clasificación en base al esquema simple de calificación que se presenta en la Ilustración 1, en el caso de los vectores de ataques y la detectabilidad de debilidades se clasifi-

can en Difícil, Medio o Fácil, estos niveles se definen en cuanto a la complejidad de los ataques o capacidad de detección. Otros aspectos importantes son el impacto técnico y el impacto en el negocio, los cuales se clasifican en Severo o Moderado. Por último y no por eso menos importante, está la prevalencia de debilidades la cual se clasifica en Muy Difundida, Muy Común, Común o Poco Común, según la probabilidad de ocurrencia.

4. RESULTADOS

Luego del análisis realizado a los sistemas de la Superintendencia de Bancos y Seguros se determinaron que existe mayor probabilidad de incurrir en los siguientes riesgos Almacenamiento Criptográfico Inseguro y Protección Insuficiente en la Capa de Transporte de acuerdo a lo indicado en la Ilustración 2 e Ilustración 3 debido a las características que presentan los mismos, al objeto social al cual están destinados y a las especificaciones técnicas particulares de cada uno.

En base a los riesgos listados anteriormente se describe a continuación buenas prácticas que se deben tomar en cuenta para una posible solución en el desarrollo y mantenimiento de aplicaciones que garanticen la navegación, la confidencialidad y la integridad de los datos y de la información mostrada.

- En los sistemas SAPLA y SPI se deben utilizar algoritmos de encriptación de datos para proteger la información. Se recomienda utilizar algoritmos asimétricos que son más factibles en cuanto a seguridad.
- En el caso de SOCI se debe realizar un estudio para ver como sustituir el algoritmo MD5 que se utiliza actualmente por otro más fuerte como Advanced Encryption Standard (AES).
- Las tres aplicaciones deben utilizar algún protocolo criptográfico, que proporcione seguridad en la autenticación y privacidad de la información, como por ejemplo Secure Sockets Layer (SSL) y Transport Layer Security (TLS).
- Los proyectos que estén en desarrollo deben considerar estas vulnerabilidades en la documentación que realizan en cada una de las fases de requerimientos, diseño, implementación y pruebas. Deben asegurarse que en las aplicaciones no han sido introducidas estas vulnerabilidades y en caso de que existan deben ser eliminadas de forma correcta.

5. TRABAJOS RELACIONADOS

Según la investigación realizada existe una tesis desarrollada por Rodríguez Mario de la Escuela Superior de Ingeniería Mecánica y Eléctrica donde se muestra las técnicas de OWASP para asegurar aplicaciones web contra inyecciones SQL. (López).

6. CONCLUSIONES Y TRABAJOS FUTUROS

Se realizó una evaluación de los riesgos basados en el OWASP Top 10 - 2010, de las vulnerabilidades, amenazas, el impacto, detectándose que en las aplicaciones de la Superintendencia de Bancos y Seguros hay dos riesgos bien claros que presentan una vulnerabilidad media-alta, al igual que la amenaza, pudiendo llegar el impacto a ser grande en caso de ser explotada; estos riesgos son: Almacenamiento Criptográfico Inseguro y Protección Insuficiente en la Capa de Transporte. Aunque el riesgo Inyección no está presente se recomienda utilizar la función PreparedStatement de Java, debido a que es la primera opción de defensa que propone OWASP para prevenir ataques de inyección. Finalmente se recomienda que se reduzca el tiempo de cierre de las sesiones que actualmente este tiempo lo configura el CAS (Central Authentication Service) y la sesión se cierra 30 minutos después de que no se registre actividad por parte del usuario que está autenticado, este es un tiempo extenso por lo que se aconseja que el mismo se reduzca a 15. De esta forma se evita que los atacantes tengan menos oportunidad de realizar un ataque en ese tiempo.

Como trabajo a futuro se plantea utilizar la nueva versión del OWASP Top 10 – 2013 para el análisis de las aplicaciones Web de la Superintendencia de Bancos y Seguros, destinar presupuesto, tiempo e implementar políticas de desarrollo seguro, mecanismos de diseño, pruebas de intrusión y revisión de seguridad al código fuente.

7. REFERENCIAS BIBLIOGRÁFICAS

IBM. (s.f.). *Guía del Usuario*. Recuperado el 16 de Abril de 2014, de ibm.com: https://publib.boulder.ibm.com/tividd/td/TRM/SC23-4822-00/es_ES/HTML/user277.htm#Top_Of_Page

López, M. (s.f.). EFECTIVIDAD DE TÉCNICAS DE OWASP PARA ASEGURAR APLICACIONES WEB CONTRA INYECCIÓN DE SQL. Guadalajara, México.

OWASP. (25 de Febrero de 2012). *Guide to Cryptography*. Recuperado el Abril de 2014, de https://www.owasp.org/index.php/Guide_to_Cryptography

OWASP. (2014 de Abril de 2014). *Cryptographic Storage Cheat Sheet*. Recuperado el 16 de Abril de 2014, de https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet

OWASP. (29 de 3 de 2014). *OWASP Top Ten 2010 Project.* Recuperado el 2 de 04 de 2014, de http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202010.pdf

Salgado, A. L. (2014). ANÁLISIS DE LAS APLIACIONES WEB DE LA SUPERINTENDECIA DE BANCOS Y SEGUROS, UTILIZANDO LAS RECOMEN-DACIONES TOP TEN DE OWASP PARA DETERMINAR LOS RIESGOS MÁS CRÍTICOS DE SEGURIDAD. Quito.

UNAM-CERT. (26 de Mayo de 2011). *Aspectos Básicos de la Seguridad en Aplicaciones Web*. Recuperado el 22 de Abril de 2014, de http://www.seguridad.unam.mx/documento/?id=17