RECIBIDO/RECEIVED: 2017-02-15 | ACEPTADO/ACCEPTED: 2018-01-30 | PP. 19 -30

Modelo para la reducción de riesgos de seguridad informática en servicios web

Standard for the reduction of computer security risks in web services

Jessica Castillo Fiallos
Universidad Técnica de Cotopaxi (Ecuador)
jessica.castillo@utc.edu.ec

Andrés Cisneros Barahona
Universidad Nacional de Chimborazo (Ecuador)

Pablo Méndez Naranjo
Escuela Superior Politécnica de Chimborazo (Ecuador)

Diego Jácome Segovia
Universidad Técnica de Machala, Machala (Ecuador)

Revista Cumbres Vol.4 Nº2
Versión impresa ISSN 1390-9541
Versión electrónica ISSN 1390-3365
http://investigacion.utmachala.edu.ec/revistas/index.php/Cumbres

RESUMEN

Para el estudio de la presente investigación se consideró la problemática y falencia interna que generan las vulnerabilidades en los servicios web que brinda la ESPOCH. Debido a defectos del software, al recurso humano puesto que no cumple los procesos adecuados al momento de modificar los servicios web. La falta de capacitaciones constantes y acorde con las funciones que desempeñan estos factores provocan que sea de fácil acceso a la información con la que cuentan los servicios web por parte de los atacantes y podría ocasionar fallas en la seguridad accidental o intencionalmente. Por lo expuesto anteriormente fue necesario contar con una herramienta que permita mitigar y proteger la información contra este tipo de ataques.
Por esta razón se desarrolló la propuesta que ayudó a reducir el riesgo de seguridad informática en los servicios web de la Institución que se encuentran alojados en el Departamento de Tecnologías de Información y Comunicación. Se utilizó la metodología MAGERIT y se realizó el análisis utilizando la herramienta VEGA de Linux, con los resultados se identificaron alternativas que mitigaron las vulnerabilidades encontradas en el análisis preliminar de la investigación: vulnerabilidades altas 416, vulnerabilidades medias 175 y vulnerabilidades bajas 1475, las vulnerabilidades más frecuentes son: SQL Injection, PHP Error Detected y Directory Listing Detected, entre otras. La aplicación de la propuesta permitió reducir las vulnerabilidades altas encontradas.

Palabras clave: vulnerabilidad, riesgo, servicios web, seguridad en sitios web, MAGERIT, VEGA.

ABSTRACT

For the study of the present investigation, the problem and internal flaw generated by the vulnerabilities in the web services provided by ESPOCH was considered. Due to software defects, the human resource since it does not comply with the appropriate processes at the time of modifying the web services. The lack of constant training and according to the functions performed by these factors make it easy to access the information that web services have on the part of the attackers and could accidentally or intentionally cause security failures. Therefore, it was necessary to have a tool to mitigate and protect the information against this type of attacks.
For this reason, the proposal was developed that helped reduce the risks of computer security in the Institution's web services that are housed in the Department of Information and Communication Technologies. The MAGERIT methodology was used and the analysis was carried out using the tool Linux VEGA, with the results were identified alternatives that mitigated the vulnerabilities found in the preliminary analysis of the investigation: high vulnerabilities 416, average vulnerabilities 175 and low vulnerabilities 1475, the most frequent vulnerabilities are: SQL Injection, PHP Error Detected and

Directory Listing Detected, among others. The application of the proposal allowed to reduce the high vulnerabilities found.

Keywords: vulnerability, risk, web services, website security, MAGERIT, VEGA.

INTRODUCCIÓN

La gestión de seguridad referente a los riesgos en los sistemas web puede ser compleja debido al desconocimiento o falta de cultura con respecto a este tema. Por lo que el principal problema es la falta de un estándar específico de seguridad informática para la gestión del riesgo que establezca reglas, normas, controles, políticas y procedimientos para los mismos. El objetivo principal de la gestión de riesgos es analizar, prevenir, proteger o mitigar las posibles vulnerabilidades que son condiciones que cuando son explotadas por personas malintencionadas pueden dar lugar a fallas de seguridad (Shirey, 2000). Esta debilidad se encuentra latente y puede afectar la seguridad, integridad, disponibilidad de la información sensible que estos sistemas manejan.

Con el amplio uso de Internet, viene un aumento en las amenazas a la seguridad de la información. Para protegerse contra estas amenazas, se ha descubierto que la tecnología por sí sola no es suficiente, ya que puede ser utilizada por los usuarios y convertirse en vulnerable a diversas amenazas, perdiendo así su utilidad. (Alohali, Clarke, Furnell, & Albakri, 2017)

Los servicios web se han convertido en una de las tecnologías más utilizadas en los sistemas orientados a servicios. Su popularidad se debe a su propiedad de adaptarse a cualquier contexto. Como consecuencia de la creciente cantidad de servicios web en Internet y su importante papel en muchas aplicaciones actuales, la calidad de los servicios web se ha convertido en un requisito crucial y demandado por los consumidores de servicios (Ruiz & Rubira, 2016). Los evaluadores de penetración con frecuencia se centran en esta área. Este enfoque generalmente se debe a la gran cantidad de vulnerabilidades que se pueden encontrar en las aplicaciones web y la facilidad con la que pueden introducirse (Faircloth, 2017).

Los ciberataques contra las vulnerabilidades de las aplicaciones web en los últimos años se han incrementado, por lo que es necesario que las aplicaciones web sean más seguras. Sin embargo, verificar todas las vulnerabilidades web a mano es muy difícil y lleva mucho tiempo. Por lo tanto, es necesario utilizar herramientas para realizar escaneos de vulnerabilidades de las aplicaciones web. (Makino & Klyuev, 2015)

Los sistemas informáticos que poseen las Instituciones de Educación Superior no cumplen los estándares mínimos necesarios para garantizar mayor seguridad, por lo que existen vulnerabilidades que podrían materializarse y convertirse en riesgos para la organización. La investigación se realizó en el Departamento de Tecnologías de la Información y Comunicación (DTIC) de la Escuela Superior Politécnica de Chimborazo (ESPOCH). Se identificaron

activos, amenazas, salvaguardas, impactos, vulnerabilidades y con ellos se determinó el nivel de riesgo existente en los sistemas web analizados con la herramienta VEGA de Linux.

Se propone un Modelo para la Reducción de Riesgos de Seguridad Informática en servicios web de la ESPOCH que incluye las sugerencias de solución a las vulnerabilidades encontradas. Se concluye que implementar adecuadamente una metodología de reducción de riesgos fortalece a la organización brindando mayor seguridad, disponibilidad, confidencialidad, autenticidad e integridad.

Existen investigaciones previas relacionadas al tema, por ejemplo:

Vicente y Jiménez (2014), mencionan que se han desarrollado varios métodos basados en la norma ISO 27000 norma internacional / IEC para lidiar con el análisis de riesgos en los sistemas de información (SI). Proponen una extensión de la metodología MAGERIT basado en modelos computacionales difusos clásicos, respecto a la selección de las salvaguardias preventivas para reducir los riesgos, se propone un método basado en programación dinámica que incorpora recocido simulado para hacer frente a los problemas optimizaciones con el objetivo de minimizar los costos mientras se mantiene el riesgo a niveles aceptables.(Vicente & Jimenez, 2014)

Kyushu, Hori, & Sakurai (2009), comparan los métodos de análisis de riesgos: Mehari, Magerit, NIST800-30 y la Guía de Gestión de Seguridad de Microsoft. Mehari es un método para el análisis y gestión de riesgos. Magerit es un análisis de riesgos y metodología de gestión de sistemas de información. NIST 800-30 es una guía de gestión de riesgos para los sistemas de tecnología de la información. La seguridad es una guía de gestión de riesgos de seguridad desarrollado por Microsoft. Se comparan los métodos basados en dos criterios principales: el primer criterio es los pasos que utilizan los métodos para llevar a cabo la evaluación del riesgo, el segundo es el contenido de los métodos y los documentos complementarios previstos con ellos. (Kyushu, Hori, & Sakurai, 2009)

Kwan y Leung (2010), tratan acerca de los riesgos no siempre son independientes ya que no existe una administración clara entre ellos. Las dependencias pueden ser identificadas de forma explícita y analizadas, los administradores del proyecto deben ser capaces de planificar estrategias efectivas contra los riesgos con la finalidad de tomar decisiones, sin embargo, la investigación fue solamente teórica y no fue implementada para verificar la reducción de riesgos que pruebe la propuesta planteada. (Kwan & Leung, 2010).

MATERIALES Y MÉTODOS

Como referencia se ha analizado las metodologías previas utilizadas por otros autores, por ejemplo:

Monteverde y Campiolo (2014), mencionan que la seguridad web es importante para proporcionar protección a los clientes y a los servicios web. Múltiples vulnerabilidades web son explotados todos los días y los ataques tienen aumentado debido a las nuevas herramientas y aplicaciones web, se

lleva a cabo un análisis de vulnerabilidades web en diferentes tipos de aplicaciones con la herramienta de escáner VEGA.

Khari y Singh (2014), las aplicaciones resultan ser herramientas de uso cotidiano por muchos usuarios con la creciente popularidad de la web. Con esta aplicación web los usuarios son más propensos a los ataques maliciosos en consecuencia la necesidad de pruebas de seguridad surge también. Las pruebas de seguridad ayudan a mitigar las vulnerabilidades en la web. Lo que ocurre frecuentemente en aplicaciones web son el resultado de problemas de validación de entrada de genéricos. Las herramientas VEGA y ZAP son escáneres que ofrecen una buena opción para probar vulnerabilidades en forma automatizada.

Por lo que, para la presente investigación se utilizó la Metodología MAGERIT (Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información) y la herramienta VEGA de Linux.

La Metodología MAGERIT es un método formal para investigar los riesgos que soportan los Sistemas de información, y para recomendar las medidas apropiadas que deberían adoptarse para controlar estos riegos (Desogles, 2005). Además, permite manejar gráficos de complejidad variable y permite la valoración de activos cualitativa y cuantitativa. (Fernáncez & Garcia, 2016).

Es una de las más utilizadas a nivel de Latinoamericano, la cual consta de:

- Activos: Son los recursos que pertenecen al propio sistema de información o que están relacionados con este. La presencia de los activos facilita el funcionamiento de la empresa u organización y la consecuencia de los objetivos. (Aguilera, 2010, pág. 9)
- Amenazas: Una posibilidad de ocurrencia de cualquier tipo de evento que puede producir un daño sobre los elementos de un sistema e información, las amenazas y por consecuentes daños que puede causar un evento de este tipo. (Erb, 2014)
- Salvaguardas: Una salvaguarda es un mecanismo de protección frente a las amenazas, reducen la frecuencia de las amenazas y limitan el daño causado por estas. (Reyes, 2015, pág. 78)
- Impacto: Es la consecuencia sobre éste de la materialización de una Amenaza en agresión, consecuencia que puede desbordar ampliamente el Dominio y requerir la medida del daño producido a la organización. (Reyes, 2015, pág. 34)
- Vulnerabilidad: Es la capacidad, las condiciones y características del sistema mismo (incluyendo la entidad que lo maneja), que lo hace susceptible a amenazas, con el resultado de sufrir algún daño. En otras palabras, es la capacitad y posibilidad de un sistema de responder o reaccionar a una amenaza o de recuperarse de algún daño. (Erb, 2014)
- Riego: La posibilidad de que se materialice o no una amenaza aprovechando una vulnerabilidad. No constituye riesgo una amenaza cuando no hay vulnerabilidad ni una vulnerabilidad cuando no existe amenaza para la misma. (Aguilera, 2010, pág. 45).

La investigación requirió de técnicas como las documentales bibliográficas,

entrevistas al administrador de Departamento de Tecnologías de Información y Comunicación, pruebas y análisis con la Plataforma VEGA. El universo que se tomó en consideración para la realización de las pruebas son los servicios web de la Plataforma de la ESPOCH. VEGA es un escáner de código abierto y libre y una plataforma de prueba para probar la seguridad de las aplicaciones web. Esta herramienta puede ayudarle a encontrar y validar SQL Injection, Cross-Site Scripting (XSS), reveló inadvertidamente información confidencial y otras vulnerabilidades. Está escrito en Java, basado en GUI, y se ejecuta en Linux, OS X y Windows.

VEGA incluye un escáner automatizado para pruebas rápidas y un proxy de interceptación para la inspección táctica. El escáner de VEGA encuentra XSS (cross-site scripting), inyección de SQL y otras vulnerabilidades. VEGA se puede ampliar usando una poderosa API en el lenguaje de la web: escáner automatizado de rastreadores y vulnerabilidades, interfaz de usuario coherente, crawler del sitio web, proxy de interceptación, SSL MITM, análisis del contenido, extensibilidad a través de un potente API de módulo Javascript, alertas personalizables, base de datos y modelo de datos compartidos. (Kali Tools, 2014). Para la identificación de las vulnerabilidades se utilizó el escáner VEGA de Linux, estas pruebas se realizaron a 10 de los 13 servicios web que se encontraron activos. En el escáner VEGA se introduce la URL de cada servicio web donde se identifican las vulnerabilidades, cada escaneo duro más de 48 horas.

RESULTADOS Y DISCUSIÓN

A continuación, se muestra el desarrollo y los resultados de la investigación aplicando la metodología MAGERIT y utilizando el escáner VEGA.

Identificación de activos de información

La identidad de activos es importante ya que permite materializar con precisión el alcance de la investigación, permite valorar los activos con veracidad e identificar las amenazas a las que se encuentran expuestos, estos son: servicios web, equipos informáticos, soportes de información, instalaciones y personal.

Identificación de activos relevantes

Para la identificación de amenazas, salvaguardas y vulnerabilidades se realizó un análisis de los activos más importantes con el Técnico responsable de la Dirección de Tecnologías de la Información y Comunicación, mediante entrevistas donde se analizaron las ventajas y desventajas así de definieron los activos relevantes que son: UPS, planta de energía, storage, servidores 1, 2, 3, 4, memorias: 1, 2, 3, 4 de los servidores: 1, 2, 3, 4 respectivamente

Identificación de amenazas

Luego de identificar los activos, vamos a identificar las amenazas, tomando en cuenta una o varias amenazas que pueden afectar a cada activo.

Amenaza son eventos que pueden desatar un incidente dentro de la institución, produciendo daños materiales o pérdida de datos. Para la identificación de las amenazas que pudieren afectar a los activos, conviene clasificar por su naturaleza, para así facilitar su ubicación. Para la identificación de las amenazas se utiliza encuestas dirigidas a los técnicos que administran los servidores web de la Dirección de Tecnologías de la Información y Comunicación de la ESPOCH sobre las diferentes amenazas que existen de acuerdo a la siguiente clasificación: amenazas naturales, amenazas a instalaciones, amenazas humanas, amenazas tecnológicas, amenazas operacionales, amenazas sociales.

Identificación de salvaguardas

Una vez identificadas las amenazas, se identifica los mecanismos de salvaguarda implantados en aquellos activos, describiendo las dimensiones de seguridad que estos mantienen tomando en consideración las siguientes dimensiones: Disponibilidad, Integridad, Confiabilidad, Autenticidad. Los mecanismos de salvaguarda son procedimientos, dispositivos que ayudan a reducir los riesgos. En la Tabla 1, se muestran las diferentes salvaguardas que tienen los activos observados.

Tabla 1. Identificación de Salvaguardas

ACTIVOS DE INFORMACIÓN	SALVAGUARDA	DIMENSIÓN
UPS	Protección del equipo dentro de la organización	Disponibilidad
Planta de energía	Protección del equipo dentro de la organización	Disponibilidad
Servidor 1	Claves Protección del equipo dentro de la organización	Disponibilidad, autenticidad, confiabilidad
Servidor 2	Claves Protección del equipo dentro de la organización	Disponibilidad, autenticidad confiabilidad
Servidor 3	Claves Protección del equipo dentro de la organización	Disponibilidad, autenticidad confiabilidad
Servidor 4	Claves Protección del equipo dentro de la organización	Disponibilidad, autenticidad, confiabilidad
Memoria 1	Protección del equipo dentro de la organización	Disponibilidad, integridad, confiabilidad
Memoria 2	Protección del equipo dentro de la organización	Disponibilidad, integridad, confiabilidad
Memoria 3	Protección del equipo dentro de la organización	Disponibilidad, integridad, confiabilidad
Memoria 4	Protección del equipo dentro de la organización	Disponibilidad, integridad, confiabilidad
Storage	Protección del equipo dentro de la organización	Disponibilidad, autenticidad, confiabilidad, integridad
Servicios web	Protección del servicio dentro de la organización	Disponibilidad, autenticidad, integridad, confiabilidad
Personal	Plan de contingencia	Disponibilidad, integridad, confiabilidad

Identificación de Vulnerabilidades

Para la identificación de las vulnerabilidades se utilizó la herramienta VEGA que es un escáner de código para pruebas de plataforma libre y abierta para probar la seguridad de las aplicaciones web.

Se escaneó los URLs de los servicios web de la ESPOCH.

- **Sistemas activos**: http://sisepec.espoch.edu.ec/,http://academicoseg.espoch.edu.ec/, http://evaluacion.espoch.edu.ec/,http://recursos.espoch.edu.ec/,http://elearning.espoch.edu.ec/, http://bibliotecas.espoch.edu.ec/,http://medicina.espoch.edu.ec/,http://bienestar.espoch.edu.ec/, http://empleos.espoch.edu.ec/,http://passportsignin.espoch.edu.ec/SignIn.aspx?IdSitioSocio=4&URLSitioSocio=/Default.aspx.
- **Sistemas pasivos**: http://biblioteca.espoch.edu.ec/herbario.htm,http://infopagos.espoch.edu.ec/, http://ide.espoch.edu.ec/.

Tabla 2. Resultados de las pruebas realizadas en los servicios web de la ESPOCH.

Nombre	Grado	Escuela de Posgrado y Educación Contínua	OASIS	Evaluación Institucional	Talento Humano	Educación Virtual	Biblioteca	Médico	Bienestar Politécnico	Bolsa de Empleos	Passport	TOTAL	
Cleartext Password over HTTP	High		1		1	3	269			3	2	279	
Cross Site Scripting	High				7		2					9	
Page Fingerprint Differential Detected	High	5			2						1	8	416
Shell Injection	High			4		68	5					77	
SQL Injection	High		2	9		32						43	
HTTP Trace Support Detected	Medium			1		1			1			3	
Local Filesystem Paths Found	Medium		1			4	2	1	1	2		11	
Local Filesystem Phaths Found	Medium			62								62	
PHP Error Detected	Medium			61								61	175
Possible Source Code Disclosure	Medium							1				1	
Possible XML Injection	Medium			3		31						34	
URL Injection	Medium				3							3	
ASP/ASPX Error Detected	Low										1	1	
Directory Listing Detected	Low			6	3	1166			2			1177	
Email Addresses Found	Low								1			1	1457
Form Password Field with Autocomplete Enabled	Low				1	3	268			3	2	277	
From Password Field with Autocomplete Enable	Low		1									1	
TOTAL		5	5	146	17	1308	546	2	5	8	6		

En la Tabla 2, se resumen los resultados de las pruebas realizadas de las diferentes URLs de los servicios web de la ESPOCH.

Identificación de vulnerabilidades después de aplicar la propuesta de solución para la reducción de riesgos de seguridad informática en servicios web de la ESPOCH.

A continuación, se detallan como ejemplo las vulnerabilidades encontradas después de la aplicación de la propuesta de reducción de riesgos en tres de sus servicios web analizados y comparándolos con las vulnerabilidades encontradas.

SERVICIO WEB: Educación Virtual, **URL**: http://elearning.espoch.edu.ec/.

En la Tabla 3, se observa las vulnerabilidades encontradas en el servicio web de Educación Virtual después de aplicar la propuesta de solución.

SERVICIO WEB: Bienestar Politécnico, **URL**: http://bienestar.espoch.edu.ec/

En la Tabla IV se observa las vulnerabilidades encontradas en el Servicio Web de Bienestar Politécnico después de aplicar la propuesta de solución.

SERVICIO WEB: Talento Humano, **URL**: http://recursos.espoch.edu.ec/

En la Tabla V se observa las vulnerabilidades encontradas en el Servicio Web de Talento Humano después de aplicar la propuesta de solución.

Tabla 3. Total de vulnerabilidades

Nombre	Grado	Antes	Después
Cleartext Password over HTTP	High	3	3
SQL Injetion	High	32	12
Shell Injection	High	68	5
HTTP Trace Support Detected	Medium	1	1
Local Filesystem Paths Found	Medium	4	3
Possible XML Injection	Medium	31	1
Directory Listing Detected	Low	1166	157
Form Password Field with Autocomplete Enabled	Low	3	3

Tabla 4. Total de vulnerabilidades

Nombre	Grado	Antes	Después
HTTP Trace Support Detected	Medium	1	1
Local Filesystem Paths Found	Medium	1	1
Directory Listing Detected	Low	2	2
Email Addresses Found	Low	1	0

Tabla 5. Total de vulnerabilidades

Nombre	Grado	Antes	Después
Cleartext Password over HTTP	High	1	1
Cross Site Scripting	High	7	7
Page Fingerprint Differential Detected	High	2	0
URL Injection	Medium	3	3
Directory Listing Detected	Low	3	3
From Password Field with Autocomplete Enabled	Low	1	1

Identificación de Impactos

El objetivo de esta actividad es conocer el alcance del daño producido en el dominio (y por lo tanto sobre todos los activos que se encuentran en dicho dominio), como consecuencia de la materialización de las amenazas sobre los activos. La identificación de los impactos o valoración de los dominios se desarrollará con repercusiones a las dimensiones de valoración que son (D) Disponibilidad, (I) Integridad, (C) Confiabilidad, (A) Autenticidad de la información. Las cuatro dimensiones con sus criterios de valoración del análisis realizado con los técnicos de la DTIC son: Disponibilidad (Alto), Integridad (Alto), Confiabilidad (Medio), Autenticidad (Alto).

Tabla 6. Identificación de Impactos

ACTIVOS	DIMENSIONES			
	D	I	C	A
UPS	5	0	0	0
Planta de energía	5	0	0	0
Procesador 1 (Servidor 1)	4	3	2	4
Procesador 2 (Servidor 2)	4	3	2	4
Procesador 3 (Servidor 3)	4	3	2	4
Procesador 4 (Servidor 4)	4	3	2	4
Memoria 1 (Servidor 1)	4	4	3	4
Memoria 2 (Servidor 2)	4	4	3	4
Memoria 3 (Servidor 3)	4	4	3	4
Memoria 4 (Servidor 4)	4	4	3	4
Storage	4	5	2	5
Personal	3	1	3	0

Identificación del Riego.

En esta actividad, luego del análisis de los activos en los que se refiere a las amenazas, salvaguardas, existentes, vulnerabilidades e identificación de impactos, se identifica los activos y su nivel de riesgo existente. En la Tabla 6, se muestra la escala de daño para identificar los riesgos.

Propuesta para la reducción de riesgos para los servicios web de la ESPOCH.

Tomando en consideración que la ESPOCH se encuentra vulnerable en un 40% a los riesgos en sus servicios web se ha diseñado una propuesta que brindará posibles soluciones para las diferentes

vulnerabilidades encontradas. En coordinación con los objetivos, estrategias y políticas de la Dirección de Tecnologías de la Información y Comunicación, las actividades de gestión de riesgos permiten elaborar un plan de seguridad que implantado y operado satisfaga los objetivos propuestos con un nivel de riesgo inferior al que se obtuvo en el análisis preliminar. En la Tabla 7, se muestra la propuesta de solución.

Tabla VII Propuesta de solución

VULNERABILIDAD	PROPUESTA DE SOLUCIÓN
Cleartext Password over HTTP	Las contraseñas nunca deben ser enviadas a través de texto plano. Elaborar contraseñas fuertes y cifrarlas. Una contraseña fuerte debe contener mínimo 8 caracteres: 2 caracteres especiales, 2 números, 2 letras mayúsculas y 2 minúsculas.
SQL Injection	La mejor defensa contra las vulnerabilidades de SQL Injection es utilizar instrucciones con parámetros. Las variables de tipos de cadenas deben ser filtrados, y tipos numéricos deben ser evaluados para verificar que son válidos. El uso de procedimientos almacenados puede simplificar consultas complejas y permitir la configuración de control de acceso más estricto. Configuración de los controles de acceso de base de datos puede limitar el impacto de las vulnerabilidades explotadas.
Local Filesystem Paths Found	Cuando se obtenga una salida de error que contiene información confidencial, como rutas de sistema absolutos no debería ser enviada a los clientes remotos en servidores de producción. Esta salida debe ser enviada a otro log de salida, como un registro de errores.
From Password Field with Autocomplete Enable	El valor del atributo de autocomplete en el formulario debe tener el valor “off”. No generar autocomplete.
Page Fingerprint Differential Detected	Para evitar este tipo de vulnerabilidad, el desarrollador debe predeterminar el camino de cualquier recurso del sistema de archivos que tiene una trayectoria compuesta de entrada suministrada externamente y luego realizar una comprobación de autorización previa para el acceso. Cuando se desarrolle en PHP, Perl y Python se debe utilizar la función realpath (), cuando se utilice aplicaciones ASP.NET se debe utilizar la función GetFullPath (), cuando se utiliza en código Java se utilizar la función getCanonicalPath () estas funciones devuelven la ruta predeterminada así se evita este tipo de vulnerabilidad. Protección adicional contra el acceso no autorizado al sistema de ficheros de recursos se puede obtener mediante el uso de chroot () o mecanismos similares para limitar el acceso del sistema de archivos para el proceso de servidor de aplicaciones web y http, aunque esto puede ser difícil de manejar.
Shell Injection	Los desarrolladores deben examinar el código correspondiente a la página en detalle para determinar si existe la vulnerabilidad. La ejecución de comandos de sistema a través de un intérprete de comandos, como por ejemplo con el system (), debe ser evitado. El desarrollador debe validar las entradas antes de que se pasa al intérprete.
HTTP Trace Support Detected	Para los servidores basados en Apache, la función TraceEnable () se puede utilizar para desactivar el soporte para HTTP TRACE. Para los servidores basados en IIS, la función EnableTraceMethod () se puede utilizar para desactivar el soporte para HTTP TRACE.
Possible XML Injection	Los desarrolladores deben investigar el código para verificar manualmente que existe una vulnerabilidad de XML injection. Caracteres que se pueden interpretar como XML deben ser filtrados como por ejemplo >, <, ‘ , ”, etc.
Directory Listing Detected	Para Apache, realizar una de las siguientes opciones: añadir “IndexIgnore “ para archivo .htaccess del directorio, o bien eliminar “Índices” de la línea “Opciones Todos los índices FollowSymLinks MultiView” en su archivo de configuración de Apache.

Cross Site Scripting	No confiar nunca en datos que se obtenga de los usuarios o de cualquier fuente de datos externa. Filtrar datos poco confiables que son generados por el cliente. Esta regla es la única que tenemos que seguir para prevenir los ataques XSS. Para evitar ataques XSS, se debe llevar a cabo la validación de datos, el saneamiento y escapar lo que se va a mostrar
URL Injection	El desarrollador debe examinar la etiqueta y determinar las posibles implicaciones de seguridad de la utilización de un URI suministrado de forma remota.
Email Addresses Found	Las direcciones de correo electrónico incrustados en el contenido proporcionado por el usuario deben ser filtrados para evitar la divulgación no intencional. No es recomendable mostrar las librerías de javascript ya que el servidor automáticamente puede mostrar direcciones de correo.

Las investigaciones previas relacionadas al tema definen un modelo de reducción de riesgos, sin embargo, no son implementados para comprobar su eficiencia o no plantean sugerencias de control, sin embargo, en la presente investigación, se plantea el modelo propuesto para la reducción de riesgos de seguridad informática en servicios web de la ESPOCH y se lo implementa, lo cual permite reducir las vulnerabilidades encontradas de forma teórica y práctica con valores cuantitativos, demostrando su efectividad. Además, se enfoca en el grado de impacto que generan los mismos, priorizándolos con la finalidad de reducir los riesgos principales. En cada sistema web en funcionamiento se aplicó la propuesta y se obtuvieron resultados positivos de mejora, lo que permite un mejor desempeño reduciendo las vulnerabilidades existentes en los sistemas de la ESPOCH, lo que brinda mayor seguridad y un mejor servicio a la comunidad. De los 17 tipos de vulnerabilidades encontradas en los servicios web, varias de ellas fueron comunes. Al plantear propuestas de mejora para una de ellas se las debe replicar en todas para reducirlas o eliminarlas, enfocándose principalmente desde las vulnerabilidades de nivel alto (High) que implican un mayor riesgo e impacto para su funcionamiento. Estas medidas permiten optimizar recursos y obtener resultados significativos.

CONCLUSIONES

- La aplicación de la propuesta de un modelo para la reducción de riesgos de seguridad informática en servicios web de la ESPOCH permitió reducir en un 80,59% las vulnerabilidades altas encontradas en el análisis y la eliminación del 19.41% de las mismas.
- Las tres vulnerabilidades más frecuentes fueron: SQL Injection, PHP Error Detected y Directory Listing Detected.
- El personal técnico no dispone de conocimientos técnicos avanzados lo que es una vulnerabilidad adicional en el servicio.
- A pesar de que existen un sin número de metodologías de reducción de riesgos se escogieron MEGERIT y OCTAVE debido a las ventajas que brindan.

REFERENCIAS BIBLIOGRÁFICAS

Aguilera, P. (2010). *Informática y comunicaciones.* Madrid: Editex S.A.

Alohali, M., Clarke, N., Furnell, S., & Albakri, S. (2017). Information security behavior: Recognizing the influencers . *Computing Conference.* London: IEEE.

Desogles, J. (2005). *Ayudantes técnicos de Informatica.* Madrid: Editorial Mad S.L.

Erb, M. (2014). *Amenazas y vulnerabilidades*. Obtenido de https://protejete.wordpress.com/gdr_principal/amenazas_vulnerabilidades/

Erb, M. (2014). *Gestión de Riesgo en la Seguridad Informática*. Obtenido de https://protejete.wordpress.com/gdr_principal/amenazas_vulnerabilidades/

Faircloth, J. (2017). *Chapter 5 – Web applications and services.* Michael Rogers, Technical Editor.

Fernáncez, A., & Garcia, D. (2016). Complex vs. simple asset modeling approaches for information security risk assessment: Evaluation with MAGERIT methodology. *Innovative Computing Technology (INTECH).* Dublin: IEEE.

Khari, M., & Singh, N. (2014). Web Services Vulnerability Testing Using Open Source. *International Journal of Advanced Engineering and Global Technology*, 790-799.

Kwan, T., & Leung, H. (2010). A Risk Management Methodology for Project Risk Dependencies. *IEEE Transactions on Software Engineering* (págs. 635-648). IEEE.

Kyushu, F., Hori, Y., & Sakurai, K. (2009). Comparison of Risk Analysis Methods: Mehari, Magerit, NIST800-30 and Microsoft's Security Management Guide. *Availability, Reliability and Security, 2009. ARES '09.* (págs. 726-731). IEEE.

Makino, Y., & Klyuev, V. (2015). Evaluation of web vulnerability scanners . *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS).* Warsaw, Poland: IEEE.

Monteverde, W. A., & Campiolo, R. (2014). *Estudo e Analise de Vulnerabilidades Web.* Obtenido de http://es.slideshare.net/wamverde/estudo-e-anlise-de-vulnerabilidades-web

Reyes, J. (2015). *MAGERIT*. Obtenido de https://seguridadinformaticaufps.wikispaces.com/MAGERIT

Ruiz, J., & Rubira, C. (2016). Quality of Service Conflict During Web Service Monitoring: A Case Study. *Science Direct, 321*, 113-127.

Shirey, R. (2000). Internet Security Glossary. *RFC 2828 (Informational). Obsoleted by RFC 4949*.

Vicente, E., & Jimenez, A. (2014). Risk analysis in information systems: A fuzzification of the MAGERIT methodology. *Elsevier*, 1-12.