ISSN: 2773-7330

Alpha publicaciones

Vol. 3, N° 2.2, p. 66-82
junio, 2021

Recibido: 06-04-2021 / Revisado: 14-04-2021 / Aceptado: 09-05-2021 / Publicado: 04-06-2021

Aplicación de MAGERIT para reducir riesgos en servicios Web en un contexto académico en Ecuador.

DOI: https://doi.org/10.33262/ap.v3i2.2.60

Application of MAGERIT to reduce risks in Web services in an academic context in Ecuador.

Diego Jácome Segovia. [1], Jessica Castillo Fiallos. [2], Carmita Mantilla Cabrera. [3] & Byron Ernesto Vaca Barahona. [4]

Abstract

This study presents an adaptation of the MAGERIT methodology that allows us to manage the IT security risks of a company's web services. For this purpose, the first step was to determine the company's information assets along with possible threats, and in the event of materialization the impact of these threats was then measured to identify the safeguards of these assets. After this, a web services vulnerability detection test was performed using a free software tool called VEGA. Finally, the level of risk was determined so that IT staff will be better able to make future decisions. In the analysis of the vulnerability of web services, the most common vulnerabilities found were: SQL Injection, PHP Error Detected and Directory Listing Detected, among others. With the implementation of this model, high risk vulnerabilities were reduced to 87.87% and 12.13% of all vulnerabilities were eliminated.

Keywords: Vulnerability, Risk, Web Services, Security, MAGERIT, VEGA

[1] Universidad Técnica de Cotopaxi, diego.jacome@utc.edu.ec, ORCID: https://orcid.org/0000-0001-7681-5386

[2] Universidad Técnica de Cotopaxi, jessica.castillo@utc.edu.ec, ORCID: https://orcid.org/0000-0002-3120-7229

[3] Escuela Superior Politecnica de Chimborazo, carmen.mantilla@espoch.edu.ec, ORCID: https://orcid.org/0000-0001-5422-7073

[4] Escuela Superior Politecnica de Chimborazo, bvacab@espoch.edu.ec, ORCID: https://orcid.org/0000-0002-3622-0668

Resumen

Se presenta un modelo basado en la metodología MAGERIT que permite gestionar los riesgos de seguridad informática de los servicios web de una empresa. Para este propósito se determinaron los activos de información con sus posibles amenazas y el impacto que estas tendrían si se materializaran, luego se identificó las salvaguardas de los activos y se realizó la detección de vulnerabilidades en los servicios web mediante el uso de la herramienta gratuita VEGA. Finalmente, se determinó el nivel del riesgo que a futuro permitirá al personal de tecnologías tomar mejores decisiones. En el resultado del análisis de vulnerabilidades de los servicios web se identificaron como más frecuentes: SQL Injection, PHP Error Detected y Directory Listing Detected, entre otras, con lo que al aplicar el modelo propuesto se redujo un 87.87% las vulnerabilidades altas encontradas y se eliminaron el 12.13% de las mismas.

Palabras clave: Inseguridad, Amenazas, URLs, Seguridad, MAGERIT, VEGA

Introducción.

Con la gran cantidad de vulnerabilidades informáticas conocidas y amenazas, los activos informáticos están más expuestos que nunca a riesgos de índole como: el robo de información, perdida de dinero y clientes, falta de disponibilidad del servicio, y sobre todo la credibilidad de la empresa hacia el mundo exterior, por esto es necesario que las empresas u organizaciones cuenten con un modelo de gestión de riesgos que ayude a reducir el impacto que ocasionaría que una de las vulnerabilidades si llegase a ser materializada. La gestión de riesgos es un enfoque que nos permite manejar la probabilidad de que una o varias amenazas tomen ventaja de una vulnerabilidad y el impacto asociado que estas causarían en el evento de que sucedan. Por lo que se utilizan diferentes controles, políticas, normas, estrategias para poder manejarlo y mitigarlo utilizando los recursos disponibles (Freitas, 2009).

La gestión de riesgos en los sistemas web puede ser compleja debido al desconocimiento con respecto a este tema. Por lo que el principal problema en Ecuador es la falta de un estándar específico para la gestión de este tipo de riesgos por lo que es necesario un modelo o metodología que establezca reglas, normas, controles, políticas y procedimientos para los mismos, con el objetivo de 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, lo que representa una debilidad latente en la seguridad, integridad, disponibilidad de la información sensible que estos sistemas manejan, la cual podría ser utilizada por terceras personas sin la autorización de la institución (Shirey, 2000).

El siguiente trabajo propone un modelo para la gestión de riesgos para servicios web basado en la metodología MAGERIT, el cual consta de seis etapas: identificación de activos según su importancia en la organización, determinación de amenazas que existen en el entorno, las salvaguardas que existen al momento de realizar la primera evaluación

de riesgos, el impacto asociado al riesgo, análisis de las posibles y conocidas vulnerabilidades y finalmente el riesgo actual de la organización. Este modelo está enfocado para ser implementado en Instituciones educativas ya que son un potencial objetivo para cometer delitos informáticos (Desogles, 2005).

Trabajos relacionados

Para el estudio se consideraron de mayor relevancia los resultados de las investigaciones que se listan a continuación:

En la publicación efectuada en octubre del 2018, titulada Modelo para la reducción de riesgos de seguridad informática en servicios web (Castillo, Cisneros, Mendez, & Jácome, 2018), manifiesta que se ha considerado la problemática y falencias internas que generan los tipos de vulnerabilidades a los que son sujetos los servicios web que brinda la ESPOCH. Debido a la utilización de software no certificado y a la escasa capacitación de recurso humano en el manejo de etapas críticas al momento de administrar y modificar los servicios web.

En un trabajo previo de (Monteverde y Campiolo, 2014), menciona que la seguridad web es importante para proporcionar protección a los clientes y a los servicios web. Múltiples vulnerabilidades web son explotadas 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. Un conjunto de Sitios Web heterogéneos y brasileños fueron seleccionados y analizados, en consecuencia, las principales formas de ataques utilizados en aplicaciones web se han investigado, los resultados muestran cómo las vulnerabilidades web pueden ser explotadas fácilmente, por lo que se verifica que los sitios web deben mejorar su seguridad con urgencia.

(Khari y Singh, 2014), en su investigación, indica que las aplicaciones resultan ser herramientas de uso cotidiano por muchos usuarios con la creciente popularidad de la web, con lo que los usuarios son más propensos a los ataques maliciosos en consecuencia la necesidad de pruebas de seguridad surge también para ayudar 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 como: Inyección SQL y Cross-Site Scripting (XSS), etc, estas vulnerabilidades son más a menudo explotados por atacantes que tengan acceso a información sensible. VEGA y ZAP son escáneres de código abierto que ofrecen una buena opción para probar vulnerabilidades potenciales en forma automatizada, como resultado de esto muchos sitios web en Internet que son vulnerables.

El estudio de (Vicente y Jimenez, 2014), indica 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) y proponen una extensión de la metodología MAGERIT basado en modelos computacionales difusos clásico, usa una escala lingüística término para representar los valores de activos, sus dependencias, frecuencia

y la degradación de los activos asociados a las amenazas y respecto a la selección de las salvaguardias preventivas para reducir los riesgos en IS, 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.

La investigación científica propuesta por (Kyushu, Hori y Sakurai, 2009), compara cuatro 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 riesgos para los sistemas de tecnología de la información y la seguridad es una guía de gestión de riesgos de seguridad desarrollado por Microsoft. Compara los métodos basados en dos criterios principales: los pasos que utilizan los métodos para llevar a cabo la evaluación del riesgo y el contenido de los métodos y documentos complementarios previstos con ellos. Se encontró que todos los métodos siguen los tres primeros pasos generales del análisis de riesgos. Sin embargo, el método Mehari, el método Magerit y la Guía de administración de seguridad de Microsoft no incluyen recomendaciones para el control.

La investigación planteada por (Kwan & Leung, 2010) indica que 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.

Las Instituciones educativas no cuentan con estándares para garantizar la integridad de su información, ya que existe una variedad de vulnerabilidades las cuales pueden aprovechar las amenazas y convertirse en riesgos que perjudican y ocasionan pérdidas económicas en las mismas, La investigación se realizó en el Departamento de Sistemas de una entidad académica, para la identificación las vulnerabilidades con la herramienta VEGA que es propi del sistema operativo Linux.

Metodología

Adaptación y aplicación de la metodología MAGERIT

Debido a que se necesita evaluar el nivel de riesgo en las Tecnologías de la Información y adoptar medidas apropiadas para el control, la metodología presentada se basó en MAGERIT, una de las más utilizadas en Latinoamérica ya que contempla más ampliamente los parámetros de la evaluación y existe soporte en español, su esquema metódico se presenta en la Figura 1.

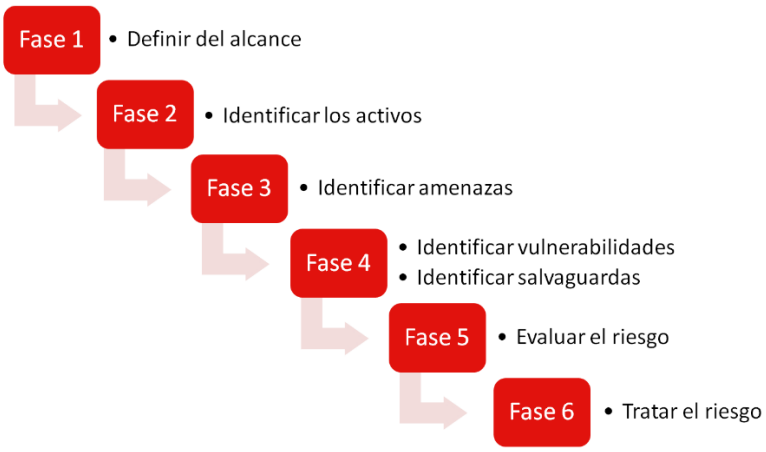

Figura N° 1: Metodología MAGERIT para determinar el riesgo

Fuente: http://polux.unipiloto.edu.co:8080/00004420.pdf

Definir del alcance. Se realizaron las pruebas para identificar los riesgos y sus posibles soluciones a los servicios Web de la Institución Educativa.

Identificar de Activos.

Los recursos que pertenecen a los sistemas de información para facilitar el funcionamiento de la organización en cumplimiento de sus objetivos, se los denominó activos (Aguilera, 2010, pág. 9). Es necesario realizar el estudio para la identificación de los activos ya que estos guardan relación entre ellos y si ocurre daño en uno puede afectar a otros, el administrador del Departamento de Computación fue quien los identificó a través de un inventario donde constan activos físicos, hardware y software, los de información tecnológicos de la organización se presentan.

Activos

- URLs. Servicios Web
- Equipos Informáticos
- Soportes de información (medios de backup)
- Instalaciones
- Personal

Identificar de Amenazas.

Para la identificación de las amenazas se consideraron las que pudieran causar cualquier tipo de evento que produciría un daño sobre los elementos de un sistema de información. Desde el punto de vista de la entidad que maneja los datos, existen amenazas de origen interno y externo como: agresiones técnicas, naturales o humanos (Erb, 2014), como menciona la Norma 27001, de acuerdo a las condiciones geográficas del lugar se planteó un check list con las posibles amenazas. A continuación, se presenta la encuesta realizada al administrador de TICs.

Amenazas

- Caída de energía
- Avería de origen físico o lógica
- Acceso no autorizado
- Errores de los usuarios
- Errores del Administrador
- Errores de configuración
- Errores de mantenimiento
- Manipulación de la configuración
- Hacking
- Pérdida de datos

Identificar de Salvaguardas.

Se consideró como salvaguarda a los mecanismos de protección que reducen la frecuencia de las amenazas, limitan el daño causado por estas y permiten mantener la triada de seguridad informática CID: Confidencialidad, Integridad y Disponibilidad. Donde la confidencialidad busca que solamente la persona o recurso autorizado tenga acceso a información o servicio específico, la integridad asegura que la información no haya sido alterada desde su origen al destino y la disponibilidad permite que la información y/o servicios estén disponibles. Existen diferentes aspectos que puede actuar como salvaguarda para alcanzar los objetivos de mitigar el riesgo. En vista de que se requieren procedimientos para la operación de salvaguardas preventivas, gestión de incidencias y recuperación tras las mismas, se solicitó al administrador un inventario de seguros de todos los activos de la organización para el levantamiento de la información de salvaguardas.

Identificar de Impacto.

Del inventario de los activos con su respectivo costo monetario, se realizó un análisis del impacto que tendría una amenaza sobre el activo sí llegara a materializarse, la identificación de los impactos se desarrollara con repercusiones a las dimensiones de valoración de Disponibilidad, Integridad, Confiabilidad, Autenticidad de la información, estas dimensiones hacen valioso al activo y la valoración de esta dimensión, es la medida del perjuicio para la organización. La escala de valoración que se implementa es Alto para un daño grave, Medio para un daño importante, Bajo para un daño menor.

Identificar de Vulnerabilidades.

Las vulnerabilidades están directamente interrelacionadas con las amenazas, si no existe una amenaza tampoco existe la vulnerabilidad porque no se puede ocasionar un daño. Entonces la capacidad, condiciones y características del sistema que lo pueden hacer susceptible a amenazas y sufrir algún daño se considera una vulnerabilidad. Dependiendo del contexto de la institución, se puede agrupar las vulnerabilidades en grupos

característicos: ambiental, física, económica, social, educativo, institucional y política. (Erb, 2014)

En la tabla 3., se presentan los servicios web identificados de la organización académica, con una descripción y estados del mismo, sobre los cuales se realizaran los análisis de vulnerabilidades, para este análisis se utilizó la herramienta Vega de código libre, escrito en Java basado en GUI y multiplataforma, incluye un escáner automatizado para pruebas rápidas, un proxy de interceptación para inspección táctica, XSS (cross-site scripting), inyección de SQL y otras vulnerabilidades (Kali Tools, 2014), se puede ampliar su funcionalidad usando una poderosa API en lenguaje Java script para realizar: escáner automatizado de rastreadores y vulnerabilidades, interfaz de usuario coherente, crawler del sitio web, proxy de interceptación, SSL MITM, análisis del contenido, alertas personalizables, base de datos y modelo de datos compartidos.

Servicios web

- Posgrado
- Ingreso de Calificaciones
- Eva-Docente
- Campus Virtual
- Bienestar
- Recursos

Identificar de Riego

Para que exista un riesgo debe existir tanto una vulnerabilidad como una amenaza, por lo que para el desarrollo de este punto se utilizó toda la información recolectada mediante el uso documentación bibliográfica, entrevistas al administrador del departamento de TICs, pruebas y análisis con VEGA (Aguilera, 2010). La valoración del nivel de riesgo existente sobre los activos, se asignó de 1 a 5, donde 5 es el más alto y se refiere un daño crítico a la organización, 4 un daño muy alto, 3 daño grave, 2 daño importante, 1 daño menor, 0 daño irrelevante.

Resultados

Identificar de activos relevantes. Se determinaron los activos que permitieron identificar las amenazas a las que se encuentran expuestos, en la Tabla 1., se enumeran los activos relevantes pero con igual grado de importancia como resultado del análisis de ventajas y desventajas de los mismos.

Tabla 1. Activos relevantes

No.	Activos de Información
1	UPS
2	Planta de energía
3	Procesador 1 (Servidor 1)
4	Procesador 2 (Servidor 2)
5	Procesador 3 (Servidor 3)
6	Procesador 4 (Servidor 4)
7	Memoria 1 (Servidor 1)
8	Memoria 2 (Servidor 2)
9	Memoria 3 (Servidor 3)
10	Memoria 4 (Servidor 4)
11	Storage

Identificar de Amenazas. En la Tabla 2., se presenta un listado de diez amenazas que pueden producir daños a los activos de la institución.

Tabla 2. Identificación de Amenazas

No.	Amenazas
1	Caída de energía
2	Avería de origen físico o lógica
3	Acceso no autorizado
4	Errores de los usuarios
5	Errores del Administrador
6	Errores de configuración
7	Errores de mantenimiento
8	Manipulación de la configuración
9	Hacking
10	Pérdida de datos

Identificar de salvaguardas. Los procedimientos y dispositivos que ayudan a reducir los riesgos como mecanismos de salvaguarda de los activos observados, se muestran en la Tabla 3.

Tabla 3. 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	Disponibilidad Integridad

	Protección del equipo dentro de la organización	Confidencialidad
Servidor 2	Claves	Disponibilidad
	Protección del equipo dentro de la organización	Integridad Confidencialidad
Servidor 3	Claves	Disponibilidad
	Protección del equipo dentro de la organización	Integridad Confidencialidad
Servidor 4	Claves	Disponibilidad
	Protección del equipo dentro de la organización	Integridad Confidencialidad
Memoria 1	Protección del equipo dentro de la organización	Disponibilidad Integridad Confidencialidad
Memoria 2	Protección del equipo dentro de la organización	Disponibilidad Integridad Confidencialidad
Memoria 3	Protección del equipo dentro de la organización	Disponibilidad Integridad Confidencialidad
Memoria 4	Protección del equipo dentro de la organización	Disponibilidad Integridad Confidencialidad
Storage	Protección del equipo dentro de la organización	Disponibilidad Integridad Confidencialidad
Servicios Web	Protección del servicio dentro de la organización	Disponibilidad Integridad Confidencialidad
Personal	Plan de contingencia	Disponibilidad Integridad Confidencialidad

Identificar de Vulnerabilidades. En la Tabla 4., en forma de resumen se muestra el resultado del análisis de vulnerabilidades realizado sobre seis de las aplicaciones web con la herramienta VEGA, como se puede observar la aplicación con mayor número de vulnerabilidades de grado alto recae sobre campus virtual y de igual manera el mayor en los de grado bajo, también se presenta una grave vulnerabilidad que consiste en envío de claves en texto plano en los servicios de campus virtual, académico y recursos.

Tabla 4. Resultados de las pruebas realizadas

Nombre	Grado	Posgrados	Ingreso de Calificaciones	Eva-Docente	Campus Virtual	Bienestar	Recursos	Total
		No.	No.	No.	No.	No.	No.	No.
Clear Password over HTTP	Alto	1	0	3	3	0	1	8
Cross Site Scripting	Alto	0	2	0	0	0	7	9
Page Fingerprint Differential Detected	Alto	5	0	0	0	0	2	7
Shell Injection	Alto	0	0	4	82	0	0	86
SQL Injection	Alto	6	2	9	33	0	0	50
HTTP Trace Support Detected	Medio	0	0	1	1	1	0	3
Local Filesystem Paths Found	Medio	0	1	62	4	1	0	68
PHP Error Detected	Medio	0	0	61		0	0	61
Possible Source Code Disclosure	Medio	0	0	0	31	0	0	31
Possible XML Injection	Medio	0	0	3	0	0	0	3
URL Injection	Medio	0	6	0	0	0	3	9
ASP/ASPX Error Detected	Bajo	0	0	0	1	0	0	1
Directory Listing Detected	Bajo	89	0	66	680	2	3	840
From Password Field with Autocomplete Enabled	Bajo	0	1	0	0	0	1	2
Total		101	12	209	835	4	17	1178

Modelo de gestión para reducción de riesgos de seguridad informática en servicios web. El análisis ejecutado a la organización educativa donde se realizó el estudio presenta vulnerabilidades que podrían poner en riesgo sus servicios web, por lo que en la Tabla 5., se presenta la propuesta para brindar posibles soluciones en las diferentes vulnerabilidades encontradas que en coordinación con los objetivos, estrategias y políticas de TICs, 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 del análisis preliminar.

Tabla 5. Propuesta de solución

Vulnerabilidad	Popuesta de Solución
URL Injection	1. El desarrollador debe examinar la etiqueta y determinar las posibles implicaciones de seguridad de la utilización de un URI suministrado de forma remota.
Directory Listing Detected	1. 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.
Cleartext Password over HTTP	1. Las contraseñas **NO** deben ser enviadas a través de texto plano. 2. Elaborar contraseñas fuertes y cifrarlas. 3. Una contraseña fuerte debe contener mínimo 8 caracteres: 2 caracteres especiales, 2 números, 2 letras mayúsculas y 2 minúsculas.
SQL Injection	1. La mejor defensa contra las vulnerabilidades de SQL Injection es utilizar instrucciones con parámetros. 2. Las variables de tipos de cadenas deben ser filtrados, y tipos numéricos deben ser evaluados para verificar que son válidos. 3. El uso de procedimientos almacenados puede simplificar consultas complejas y permitir la configuración de control de acceso más estricto. 4. Configuración de los controles de acceso de base de datos puede limitar el impacto de las vulnerabilidades explotadas.
Local Filesystem Paths Found	1. 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. 2. Esta salida debe ser enviada a otra log de salida, como un registro de errores.
From Password Field with Autocomplete Enable	1. El valor del atributo de autocomplete en el formulario debe tener el valor "OFF". 2. No generar autocomplete.
Page Fingerprint Differential Detected	1. 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. 2. 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. 3. 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	1. Los desarrolladores deben examinar el código correspondiente a la página en detalle para determinar si existe la vulnerabilidad. 2. La ejecución de comandos de sistema a través de un intérprete de comandos, como por ejemplo con el system (), debe ser evitado. 3. El desarrollador debe validar las entradas antes de que se pasa al intérprete.
HTTP Trace Support Detected	1. Para los servidores basados en Apache, la función TraceEnable () se puede utilizar para desactivar el soporte para HTTP TRACE. 2. Para los servidores basados en IIS, la función EnableTraceMethod () se puede utilizar para desactivar el soporte para HTTP TRACE.
Possible XML Injection	1. Los desarrolladores deben investigar el código para verificar manualmente que existe una vulnerabilidad de XML injection. 2. Caracteres que se pueden interpretar como XML deben ser filtrados como por ejemplo >, <, ‘ , ”, etc.
Cross Site Scripting	1. No confiar nunca en datos que se obtenga de los usuarios o de cualquier fuente de datos externa. 2. Filtrar datos poco confiables que son generados por el cliente. 2. 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

Aplicación del modelo. Al aplicar el modelo es necesario determinar el efecto sobre las vulnerabilidades, por lo que se detalla a continuación las vulnerabilidades de los servicios

web activos. En la Tabla 6., se observa las vulnerabilidades encontradas en el servicio web después de aplicar la propuesta de solución para la reducción de riesgos de seguridad informática. A pesar de la sugerencia de aplicar en su totalidad la propuesta de solución, se puede observar que disminuyeron casi toda la vulnerabilidad alta excepto Cleartext Password over HTTP debido a una decisión propia del administrador, pero en las de bajo impacto hubo una disminución considerable.

Tabla 6. Total de vulnerabilidades

Nombre	Grado	Posgrados	Ingreso de Calificaciones	Eva-Docente	Campus Virtual	Bienestar	Recursos	Total
		No.	No.	No.	No.	No.	No.	No.
Clear Password over HTTP	Alto	1	0	0	0	0	1	2
Cross Site Scripting	Alto	0	1	0	0	0	0	1
Page Fingerprint Differential Detected	Alto	1	0	0	0	0	2	3
Shell Injection	Alto	0	0	2	22	0	0	24
SQL Injection	Alto	1	2	2	13	0	0	18
HTTP Trace Support Detected	Medio	0	0	0	0	0	0	0
Local Filesystem Paths Found	Medio	0	1	12	4	0	0	17
PHP Error Detected	Medio	0	0	9	0	0	0	9
Possible Source Code Disclosure	Medio	0	0	0	3	0	0	3
Possible XML Injection	Medio	0	0	0	0	0	0	0
URL Injection	Medio	0	1	0	0	0	0	1
ASP/ASPX Error Detected	Bajo	0	0	0	0	0	0	0
Directory Listing Detected	Bajo	8	0	2	52	0	3	65
From Password Field with Autocomplete Enabled	Bajo	0	0	0	0	0	0	0
Total		11	5	27	94	0	6	143

En la Tabla 7., se observa las vulnerabilidades encontradas en los servicio web, antes y después de aplicar la propuesta de solución para la reducción de riesgos de seguridad informática. Por lo que se visualiza que el administrador consideró de relevancia mitigar la vulnerabilidad de.

Tabla 7. Total de vulnerabilidades

Total, de vulnerabilidades	**Total**	**%**
Antes	1178	100%
Después	143	12.13%
Eliminadas	**1035**	**87.87%**

Identificación de Impactos. Como resultado del análisis en la Tabla 8., indica las dimensiones con sus criterios de valoración del impacto del análisis, para conocer el daño producido sobre los activos del dominio como consecuencia de la materialización de las amenazas.

Tabla 8. Identificación de Impactos

Dimensión	Valoración
Disponibilidad	Alto
Integridad	Alto
Autenticidad	Alto
Confidencialidad	Medio

Identificación del Riego. La Tabla 9., muestra la escala de daño para identificar los riesgos como resultado del análisis de los activos, amenazas, salvaguardas existentes y la identificación de vulnerabilidades e impactos, como se puede observar el mayor riesgo se presentaría en la disponibilidad de los activos.

Tabla 9: Identificación de Riesgos

Activos	**Dimensiones**			
	Disponibilidad	**Integridad**	**Confidencialidad**	**Autenticidad**
UPS	3	0	0	0
Planta de energía	4	0	0	0
Procesador 1 (Servidor 1)	5	4	2	4
Procesador 2 (Servidor 2)	5	4	2	4

Procesador (Servidor 3)	3	4	4	2	4
Procesador (Servidor 4)	4	4	3	2	4
Memoria (Servidor 1)	1	5	4	3	4
Memoria (Servidor 2)	2	5	4	3	4
Memoria (Servidor 3)	3	5	4	3	4
Memoria (Servidor 4)	4	5	4	3	4
Storage		5	5	2	5
Personal		3	1	3	0

Conclusiones.

- Del análisis realizado se identificó 14 tipos de vulnerabilidades en los servicios web, los tres más frecuentes son: SQL Injection, PHP Error Detected y Directory Listing Detected, por lo que al plantear propuestas de mejora para un servicio web se debe replicar la solución en los demás para reducirlas o eliminarlas, enfocándose principalmente desde las vulnerabilidades de nivel alto (High) que implican un mayor riesgo e impacto para su funcionamiento.
- Con la aplicación de MAGERIT para reducción de riesgos en servicios Web se redujo en un 87,87% las vulnerabilidades encontradas en el análisis.
- A pesar de que existen un sin número de Metodologías se escogieron MAGERIT, debido a que es la más utilizada a nivel de Latinoamérica y permite tener una referencia de trabajo estándar que facilita la gestión de riesgos de una organización, que en lo posterior ayudara a reducir el impacto de las vulnerabilidades existentes en los servicios web.
- La herramienta VEGA permitió realizar un escaneo de vulnerabilidades de páginas web permitiendo determinar las debilidades y proponer alternativas de solución, debido a que es multiplataforma, gratuita, de código abierto y sobre todo puede ser adaptable a las necesidades del usuario u organización.

Referencias Bibliográficas

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

Castillo, J., Cisneros, S., Mendez, P., & Jácome, D. (2018). Modelo para la reducción de riesgos de seguridad informática en servicios web. Cumbres.

Desogles, J. X; Ayudantes técnicos de Informática. Madrid: Editorial Mad S.L, (2010).

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

Eterovic, E., y G. Pagliari; Metodología de Análisis de Riesgos Informáticos. Technical note, 10, (2010).

Freitas, V. D.; Análisis y evaluación del riesgo de la información: caso de estudio Universidad Simón Bolívar. Información Tecnológica, Scielo, 6, 13-22 (2009).

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

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

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

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

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

Shirey, R.; Internet Security Glossary. RFC 2828 (Informational). Obsoleted by RFC 4949, (2015).

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

Voutssas, J.; Documentary, digital and security information. SciELO, 24, 7 (2010).

Para citar el artículo indexado

Jácome Segovia, D., Castillo Fiallos, J., Mantilla Cabrera, C., & Vaca Barahona, B. E. (2021). Aplicación de MAGERIT para reducir riesgos en servicios Web en un contexto académico en Ecuador. AlfaPublicaciones, 3(2.2), 66–82. https://doi.org/10.33262/ap.v3i2.2.60