Aspectos fundamentales en la construcción de páginas web seguras basados en OWASP

Juan Carlos Guerrero Ortega
Robinson Andrés Jiménez Toledo
Jesús Andrés Muñoz Guzmán
Docentes del Programa de Ingeniería de Sistemas
Universidad Mariana

Andrés Mauricio Zambrano Villota
Geovanni Alberto Ojeda Ortiz
Estudiantes del Programa de Ingeniería de Sistemas
Universidad Mariana

En la actualidad existen demasiados riesgos que atacan continuamente a los equipos informáticos, sistemas de información y sistemas de comunicación, causándoles un gran daño debido a que no cuentan con controles de seguridad. El desconocimiento, el mal uso, o la inexistente utilización de buenas prácticas para el desarrollo de aplicaciones web, hacen del software un blanco fácil para posibles ataques. Cada día, se desarrollan nuevos métodos que afectan la seguridad de la información de las organizaciones, de ahí la necesidad de contar con una estrategia completa de seguridad, seguir estándares y modelos de seguridad para el desarrollo de software como lo es OWASP (Open Web Application Security Project), que permite prevenir futuras fugas y fallas en los sistemas software y páginas web.

La mayoría de las empresas centran sus esfuerzos en el desarrollo de la página web y su correcto funcionamiento, pero no le prestan mucha atención a las pautas de seguridad que tendrían que implementar en su desarrollo; las organizaciones modernas hacen uso de la tecnología e intentan sacarle el mayor provecho para llevar a cabo sus actividades empresariales, pero la tecnología evoluciona de igual manera como lo hacen sus amenazas, y puede verse vulnerable en el día a día. Por ello, es necesario que las organizaciones tomen en cuenta los riesgos a los cuales pueden verse enfrentadas, teniendo como consecuencias, robo de información, pérdida de información, alteración de información, cese de sus actividades, entre otros. Debido a esto, se hace indispensable seguir los lineamientos y estándares universalmente establecidos, para proteger los activos de la empresa a través de la seguridad en su sitio web.

Una vulnerabilidad se puede considerar como una debilidad presente en el sistema, la cual podría ser aprovechada por un atacante.

> En las páginas Web se pueden encontrar las principales vulnerabilidades que se relacionan con la no validación de entrada de datos en las aplicaciones Web, como por ejemplo: inyecciones SQL, Cross Site Scripting (XSS), inclusiones de ficheros locales (LFI) y remotos (RFI), Server SideIncludes (SSI). (Web Application Security Statistics, 2015, s.p.).

Las cuales se muestran por porcentajes en la Figura 1.

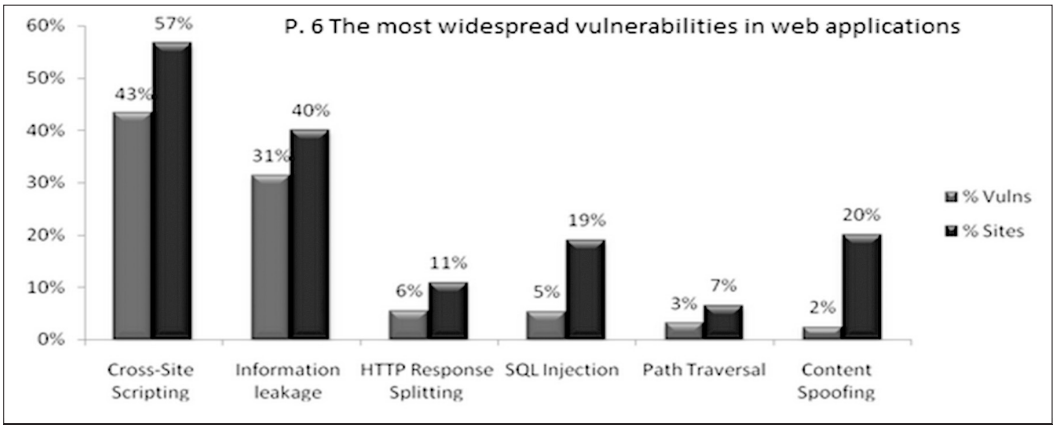

Figura 1. Estadísticas de vulnerabilidades.
Fuente: Web Application Security Statistics.

Según la Figura 1, es posible observar que las más amplias vulnerabilidades de propagación son Cross-site Scripting, diferentes tipos de fuga de información, SQL Injection, Response Splitting HTTP. Dichas vulnerabilidades afectan el rendimiento del sistema, encontrando fallos de seguridad en el mismo que desvían información (fuga de información), y permiten a los atacantes suplantar identidad, alterar datos existentes, causar problemas como anulación de transacciones y/o operaciones y cambiar balances, permite la revelación de datos del sistema, destrucción de datos o si no volverlos inasequibles, y convertirse en administradores del servidor de base de datos, trayendo consigo un deterioro del sistema y activos informáticos de una empresa, organización o desarrolladores. Frente a las amenazas nunca se puede estar 100% protegido, pero siempre hay algo que se puede hacer para disminuir al máximo las probabilidades de sufrir un ataque.

Por otra parte, las vulnerabilidades en los sistemas pueden estar ligadas a la mala administración del producto software (aplicaciones web) una vez liberado. La falta de seguimiento, mantenimiento y pruebas para fortalecer la seguridad pueden llegar a hacer que el sistema sea vulnerable frente a diversas amenazas. Otra causa que podría incidir en el bajo nivel de seguridad en los aplicativos web seria la deficiencia en el código y la carencia de buenas prácticas, como lo indica la siguiente imagen.

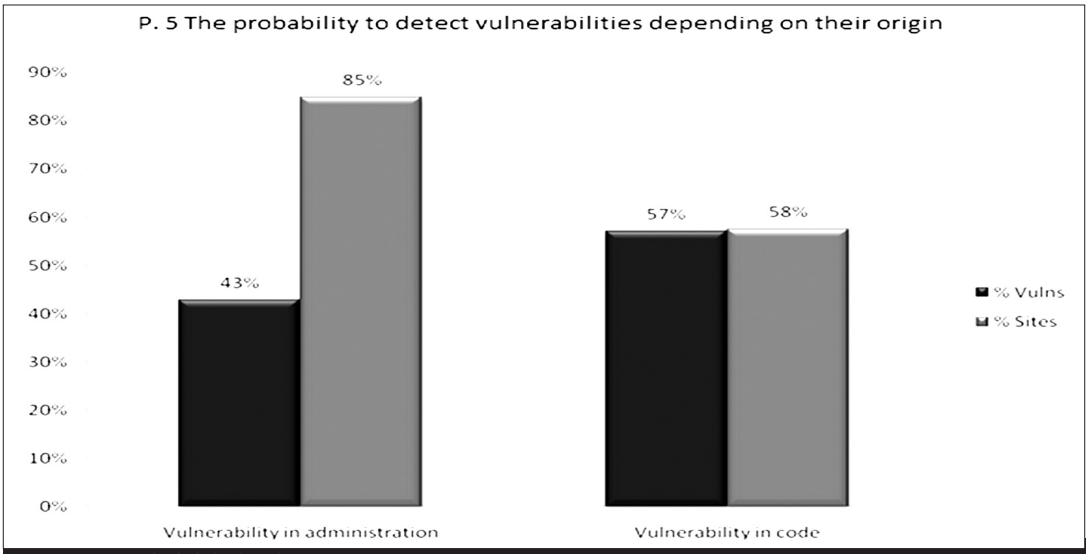

Figura 2. Probabilidades de ataques.
Fuente: Web Application Security Statistics.

Como aporte a la ausencia de buenas prácticas tanto en la codificación como en la administración de los recursos web, esta tesis pretende aportar al mejoramiento de la seguridad en la construcción de aplicaciones orientadas a la web, a través de un modelo que siga los lineamientos de OWASP, mediante la identificación de las principales amenazas o vulnerabilidades detectadas en ellas, e identificando los respectivos controles de seguridad, necesarios para corregirlos y que serán incluidos en una guía de implementación para el tratamiento de vulnerabilidades en la construcción de este tipo de aplicativos.

Aspectos metodológicos

Población: según Balestrini, la población se puede clasificar como un conjunto finito o infinito de individuos, casos, elementos y documentos que tienen características en común (Pacheco, González R., González Y., Zurita y Figueroa, 2016); además con este conjunto de elementos, se trata de sacar conclusiones. Por lo tanto, en el proyecto de investigación, la población va a estar encaminada a una revisión documental finita, que se encuentra disponible en la web, como documentos de OWASP e informes de empresas de seguridad web, que se encuentran muy bien posicionadas en el mercado como lo son Acunetix, Cenzic, Akamai e Imperva. Esto ayudará a lograr el objetivo principal de este proyecto.

Muestra: según Barrera, una muestra se realiza cuando la población es tan grande o inaccesible, lo que quiere decir que no se pueda estudiar toda, por lo tanto, el investigador tendrá la posibilidad de seleccionar una parte de ella. El muestreo no es obligatorio en una investigación, depende de los propósitos que tenga el investigador y lo referente a su proyecto (Pacheco et al., 2016). En este proyecto de investigación, no se trabajará con la muestra, debido a que la población serán todos los posibles documentos y estudios referentes a amenazas y vulnerabilidades que se encuentren mencionados en la población.

Proceso de la investigación

Tabla 1. *Detalle de metodología para cumplimiento de objetivos*

Objetivos específicos	Identificar las principales amenazas o vulnerabilidades detectadas en aplicaciones web.
Fuente	Top ten OWASP Reportes de Acunetix, Cenzic, Akamai e Imperva
Tr*	Revisión documental
Instrumento	Plantilla recopilación de problemas o vulnerabilidades
Tp**	Análisis documental
Resultado	Principales problemas de seguridad o vulnerabilidades detectadas en las aplicaciones web
Objetivos específicos	Identificar los controles de seguridad necesarios para la corrección de las principales vulnerabilidades, de acuerdo a los modelos y estándares de OWASP.
Fuente	Documentación de OWASP.
Tr*	Revisión documental.
Instrumento	Formato de clasificación de controles.
Tp**	Análisis documental.
Resultado	Clasificación de los controles de seguridad según OWASP, de acuerdo a las principales problemáticas o vulnerabilidades que presentan las aplicaciones web.
Objetivos específicos	Elaborar una guía de implementación para el tratamiento de vulnerabilidades orientado al desarrollo de páginas web.
Fuente	Segundo objetivo.
Resultado	Se elaboró una guía de implementación para el tratamiento de vulnerabilidades orientado al desarrollo de páginas web.

* Técnica de recolección.
** Técnica de procesamiento.

Resultados

Para la identificación de las principales vulnerabilidades en aplicativos web, se llevó a cabo la construcción de una tabla donde se hizo un top 5, de las vulnerabilidades más críticas según cada reporte consultado.

Tabla 2. *Top 5 vulnerabilidades encontradas*

Top 5	Akamai	Acunetix	Cenzic	Imperva	Owasp
1	Lfi (Local file inclusion)	Xss (Secuencia de comandos en sitios cruzados)	Xss (Secuencia de comandos en sitios cruzados)	Rce (Remote code excution)	Sqli
2	SQLi	Web server vulnerabilities	Information Leakage	Http	Pérdida de autentificación y gestión de sesiones
3	Xss (Secuencia de comandos en sitios cruzados)	SQLi	Authentication and Authorization	SQLi	Xss (Secuencia de comandos en sitios cruzados)
4	Rfi (Remote file inclusion)	Vulnerable JavaScript Libraries	Session Management	Xss (Secuencia de comandos en sitios cruzados)	Referencia directa insegura a objetos
5	Phpi	Code excution	SQLi	Fu	Configuración de seguridad incorrecta

Observando la anterior tabla, se puede evidenciar que ataques como "xss (Cross-site Scripting)" son frecuentes en todos los reportes y se han mantenido vigentes en los últimos años, encontrándose en todos los reportes consultados desde el año 2011 hasta el presente; al igual que "SQL injection", lo que permite conocer, son fallas típicas de las aplicaciones web que posibilita a terceros sacar provecho de esto, ocasionando graves problemas como fuga de información, secuestro de sesiones de usuario, redirección a otro sitio web, cambio de contenido, malware o un comportamiento indeseado del aplicativo, con un impacto en el proceso del negocio, dependiendo del tipo de información que se vea afectada. Esta situación se origina entre otros, a la no validación correcta de los datos de entrada, que son usados por la aplicación antes de ser procesados por el sistema, para ser incluidos en la página de salida o realizar procesos en las bases de datos, permitiendo al atacante afectar la integridad del sistema.

Por otra parte, según los reportes consultados y de acuerdo a la Tabla 2, se puede observar otro tipo de vulnerabilidades que también afectan a las aplicaciones web y que aún tienen auge en la actualidad. Encontramos fallas en el código fuente, configuración de seguridad incorrecta, mal manejo en la autenticación y sesiones de usuarios, contraseñas débiles. Todas ellas tienen un origen en la ausencia de buenas prácticas y fallas en las configuraciones del sistema y código.

Según OWASP:

> Las configuraciones de seguridad incorrectas pueden ocurrir a cualquier nivel de la aplicación, incluyendo la plataforma, servidor web, servidor de aplicación, base de datos, framework, y código personalizado. Los desarrolladores y administradores de sistema necesitan trabajar juntos para asegurar que las distintas capas están configuradas apropiadamente. (OWASP, 2013, s.p.).

Para ello, OWASP propone el uso de herramientas automatizadas que permitan la detección de fallos en la configuración o servicios innecesarios, que entorpezcan los procesos y den espacio a fallas en la seguridad de las aplicaciones.

Otro factor que contribuye en las debilidades de seguridad, tiene que ver con la falta de actualizaciones de software. Esto incluye el sistema operativo, Servidor Web/Aplicación, DBMS, aplicaciones, y todas las librerías de código.

Conclusiones

Las vulnerabilidades en los sistemas están ligadas a la mala administración del producto software (aplicaciones web) una vez liberado, la falta de seguimiento, mantenimiento y pruebas para fortalecer la seguridad pueden llegar a ser el sistema vulnerable frente a amenazas.

Vulnerabilidades como XSS e inyección SQL, han permanecido vigentes en los últimos años debido a la falta de validación de los datos suministrados por el usuario y, el desconocimiento de buenas prácticas por parte de los desarrolladores, generando deficiencia en el código, lo cual es aprovechado por los atacantes.

La reducción en los tiempos de lanzamiento de las aplicaciones web y el enfoque a una correcta funcionalidad, sin hacer énfasis en la seguridad, puede generar un producto vulnerable.

Las configuraciones de seguridad incorrectas, pueden ocurrir a cualquier nivel de la aplicación, incluyendo la plataforma, servidor web, servidor de aplicación, base de datos, framework, y código personalizado.

Las vulnerabilidades en sitios web pueden ser de diferentes tipos como fallas en el código fuente, configuración de seguridad incorrecta, mal manejo en la autenticación y sesiones de usuarios, contraseñas débiles, entre otros. Todas ellas tienen un origen, en la ausencia de buenas prácticas y fallas en las configuraciones del sistema y código

Bibliografía

Pacheco, M., González, R., González, Y., Zurita, R. y Figueroa, G. (2016). Población y muestra. Recuperado de http://msctecnologiaeducativa3.blogspot.com.co/p/poblacion-y-muestra_19.html

OWASP. (2013). OWASP Top 10 – 2013. Los diez riesgos más críticos en aplicaciones web. Recuperado de https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf>

Web Application Security Statistics. (2015). Recuperado de http://projects.webappsec.org/w/page/13246989/Web%20Application%20Security%20Statistics