Pruebas de seguridad en aplicaciones web como imperativo en la calidad de desarrollo del software

S. M. Diaz Diaz

***Abstract*- Web applications are exposed to different types of threats that affect sensible data from clients and organizations, compromising information security and the companies' business processes. Hence, it is important to take the necessary actions in order to protect data. This paper presents the main vulnerabilities in web applications, the definition of a security - quality process and the impact that insecure web applications might have on an organization. A secure web application is built from the enforcement of good development and quality assurance practices; since it is from these activities that many attacks might be avoided. For this reason it is important to define those concepts as done in this paper.**

High quality software brings recognition and reliability among clients, reducing guaranty and maintenance costs at the same time. Therefore, information security in web applications must be an imperative issue in the development and quality assurance process. An analysis of the security aspects of web application development and testing will be done, in context with a software quality assurance, and the necessity of organizations to implement strategies for quality and security in software development.

***Key words*- Information Security. Quality Assurance (QA). Web Applications. Software Quality Assurance. Secure testing practices.**

I. INTRODUCCIÓN

Las aplicaciones web están expuestas a un gran número de amenazas, tales como el mal uso por parte de los usuarios, ataques de seguridad, mala calidad del software o falta de un proceso de desarrollo estructurado. Para evitar que estos ataques se aprovechen de las vulnerabilidades comunes al software y causen un daño, es necesario que desde el principio en el ciclo de desarrollo se tomen medidas para disminuir el impacto, lo cual comienza desde la fase de levantamiento de requerimientos hasta la de entrega y soporte al cliente, pasando por las pruebas funcionales.

Las empresas deben adoptar un modelo para estandarizar e implementar un proceso basado en el ciclo de desarrollo del software, que permita realizar acompañamiento en cada una de las fases y brinde el nivel de calidad y seguridad requeridas.

El proyecto OWASP (Open Web Application Software Project) [15] se orienta al estudio de prácticas, metodologías, herramientas, documentos y demás para aplicaciones web seguras, teniendo en cuenta el desarrollo del software, las pruebas y métricas. Sus investigaciones se reflejan a través de modelos y marcos de trabajo (todo el material está disponible sin restricciones, ya que se trata de una organización sin ánimo de lucro) que permiten a los interesados en este tema e involucrados en el negocio del desarrollo de aplicaciones web, obtener información que puede ser aplicable a distintas empresas.

La calidad del software debe ser un proceso implementado en las empresas dedicadas al desarrollo de software, pues un producto de alta calidad brinda reconocimiento y fiabilidad en los clientes, así como disminución de costos por mantenimiento y garantías [1]. Por otro lado, los aspectos de seguridad de información de las aplicaciones web deben tenerse en cuenta dentro de un proceso de calidad (basado en el ciclo de desarrollo claro está), pues también es un punto crítico viéndose involucrada información de bases de datos y la disponibilidad de la aplicación.

Se ve entonces la necesidad de integrar la seguridad de información con el concepto de calidad del software, ampliando el concepto de pruebas de software y cómo se puede establecer un modelo que permita llevar a cabo esta difícil tarea de encontrar las vulnerabilidades de una aplicación web.

En este artículo se busca definir las principales vulnerabilidades [2, 3] que se presentan en las aplicaciones web, en qué consiste el proceso de pruebas de seguridad y analizar el impacto que tiene sobre las empresas la mala calidad del software [4] y la falta de monitoreo de la seguridad de información en las aplicaciones web. Se pretende también establecer una relación entre el concepto de pruebas de seguridad y el proceso de calidad del software, además de resaltar la importancia de su implementación en las organizaciones.

II . PRINCIPALES VULNERABILIDADES Y ATAQUES A LAS APLICACIONES WEB

En esta sección se mencionarán las principales vulnerabilidades [1, 4, 12, 15] y ataques realizados a las aplicaciones web, y para las cuales se deben implementar medidas que disminuyan el impacto al negocio o incluso la posibilidad de que aparezcan, según las categorías definidas en la guía de pruebas OWASP [15].

***Recopilación de información*.** El principal objetivo de este ataque es conocer a fondo la víctima y obtener tanta información como sea posible [15], como por ejemplo el sistema operativo del servidor de aplicaciones. Generalmente se utilizan herramientas de fácil manejo (redes sociales, buscadores, *scanners*) que revelan la

S.M Diaz Diaz, smdiazdiaz@gmail.com

información sin levantar sospechas. Las debilidades en las aplicaciones web respecto a estos ataques tienen que ver con la falta de clasificación de la información y la definición de permisos, pues puede que hayan datos sensibles como públicos o permisos de administrador no correspondidos.

Entre los principales métodos y actividades se encuentran Spiders, Robots, buscadores (Google), identificación de entradas de la aplicación y de scripts en el cliente.

Autenticación. Consiste en la verificación de la identidad de un usuario que tiene permisos para acceder a la aplicación [15]. Los atacantes usualmente buscan quebrantar cualquier mecanismo de autenticación que se les presente con el fin de ingresar a la información que requieren. Existen diversas vulnerabilidades respecto a estos mecanismos, como por ejemplo el no uso de canales cifrados para intercambiar información entre cliente y servidor, contraseñas débiles (propensas a ser encontradas fácilmente con ataques de diccionario o fuerza bruta), no definición de permisos (se puede ingresar a funcionalidades sin autenticación), mal manejo de caché y variables en sesión y utilizar pocos factores de autenticación (solo usuario y contraseña).

Gestión de la configuración. El conocimiento de la arquitectura de la aplicación web puede revelar información valiosa para un atacante, como los protocolos utilizados, las funcionalidades administrativas, los métodos de cifrado utilizados o el servidor de aplicaciones utilizado [15] que facilitan la labor de un atacante.

La información se puede ver vulnerada debido al no uso de algoritmos de cifrado (que provee una ventaja al atacante), información de la configuración de las bases de datos (por medio del receptor de escucha), archivos y copias de seguridad (redundancia en directorios de archivos).

Sesiones. La gestión de sesiones en una aplicación web es un aspecto importante en la seguridad de información, puesto que estas necesitan mantener el estado de las peticiones del cliente y a partir de las sesiones se pueden asociar entre sí [15]; los navegadores almacenan información sobre las sesiones en cookies. Entre las principales vulnerabilidades en el desarrollo se encuentran el mal manejo de variables de sesión, ligar la identidad del usuario al ID de la sesión y mala configuración de los atributos de sesión.

Es importante definir las prácticas de desarrollo adecuadas sobre el manejo de sesiones para evitar ataques como CSRF (Cross Site Request Forgery) o fijación de sesión.

Validación de datos. Tal como su nombre lo dice, se refiere a la verificación de datos de entrada permitidos dentro de la aplicación, y es una de las debilidades más frecuentes en las aplicaciones web. Es importante partir de la premisa de que ningún dato ingresado es confiable; además, se debe definir qué se permite teniendo en cuenta el modelo de datos, la lógica del negocio y la definición de tipo de variables en la programación.

OWASP define las siguientes como las principales categorías de ataques de validación de datos [15]:

- Cross Site Scripting: busca manipular las entradas al sistema para robar información confidencial.
- Inyección SQL [11]: se pretende introducir una consulta SQL a la base de datos, esto se presenta cuando se permite el ingreso de caracteres propios del lenguaje SQL nativo, que la aplicación no valida y que el motor de base de datos interpreta.
- Desbordamiento de buffer: se pretende comprometer la integridad de la aplicación por inconsistencia entre los formatos, tipos y longitudes entre datos ingresados por el usuario, los definidos en la aplicación y los que están en la base de datos.

Denegación de servicio. El principal objetivo de este tipo de ataque es saturar al servidor de la aplicación de tráfico de red [15], que a pesar de que las buenas prácticas de desarrollo no tienen mucho alcance en la tarea de evitarlo, sí es importante construir una arquitectura de software robusta.

III. ¿QUÉ SON PRUEBAS DE SEGURIDAD?

Las pruebas de seguridad se podrían definir como el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en aplicaciones web, buscando disminuir el impacto de ataques a ellas y pérdida de información importante [9, 10, 15].

La seguridad en aplicaciones web busca asegurar la confidencialidad, disponibilidad e integridad de los datos y funciones que maneja el software, teniendo en cuenta el impacto que pueden tener fallas de seguridad según el contexto empresarial [8].

Este proceso debe ir de la mano con todas las fases del ciclo de desarrollo de software, el cual comúnmente consta del análisis de requerimientos, el diseño, la implementación, verificación y despliegue. Muchos modelos de calidad del software se basan en este ciclo, pero en la realidad las empresas implementan el proceso de pruebas solamente en la fase de verificación; es decir, cuando el producto está finalizado y es difícil enmendar los errores que se originaron desde un principio. Esto mismo sucede con las pruebas de seguridad, solo que con un impacto mayor: muchas vulnerabilidades de las aplicaciones web se encuentran al principio del ciclo, y la seguridad de la información requiere una visión más proactiva que se base en la gestión de riesgos y prevención de fallas.

A. Diferencia con pruebas convencionales

El concepto de pruebas de seguridad abarca mucho más de lo que se conoce como las pruebas tradicionales o convencionales de software, que consisten en ejecutar una serie de casos de prueba sobre los posibles caminos o flujos del sistema, buscando errores [8] (como desbordamiento de buffer, validación de datos, funcionamiento general, cumplimiento de los requerimientos, entre otros) para ser corregidos, y así entregar un producto de calidad al cliente. En el proceso de pruebas convencional, como se ha llamado en este artículo, se realizan pruebas a la aplicación como un producto, a la tecnología como tal. Sin embargo, cuando se habla de pruebas de seguridad se debe visualizar un enfoque mucho más amplio, pues este implica hacer pruebas sobre el proceso, las personas, las políticas y también al producto [16].

Por tal razón se hace necesario que las pruebas sobre la seguridad de una aplicación web no se realicen solamente cuando el producto esté terminado, sino que se debe hacer un seguimiento desde el inicio del proyecto, de manera que se cubran las etapas del ciclo de desarrollo y se tenga control sobre las vulnerabilidades que pueden surgir desde el principio, pasando por un análisis de riesgos.

B. Alcance y principios

Como en todo proceso de calidad, se debe definir el alcance de las pruebas que se realizarán. Debido a que las pruebas de seguridad abarcan más que el producto [15], se deben tener en cuenta:

Las personas. El equipo de desarrollo, el equipo de aseguramiento de la calidad (QA), los líderes, dueños de los proyectos y los clientes. Se debe identificar quiénes son los principales actores del proceso y cómo van a influir en este, y de esta manera incluir los tipos de riesgos de recurso humano para hacer seguimiento.

El objetivo no es señalar a los trabajadores ineficientes, sino obtener retroalimentación para todos los equipos, buscando aumentar sus fortalezas y encontrar las debilidades para mejorarlas, que haya un proceso de mejora y aprendizaje continuo que permita aumentar la calidad y seguridad del producto, así como la satisfacción de los grupos.

El proceso. Las empresas deben definir el proceso mediante el cual realizarán el aseguramiento de la calidad (y de la seguridad, valga la redundancia) del software. Este proceso se debe definir teniendo en cuenta el tamaño de la organización, los recursos humanos, el conocimiento y manipulación de tecnologías, los aspectos económicos, los objetivos de negocio y la estructura empresarial. Todo esto influye en la manera en que el proceso se adaptará a la organización, pues es ella (sus directivos y encargados de QA) quien debe conocer cómo es la mejor forma de implementar estos modelos, cómo reaccionará el personal y cómo lo motivará a que participe en esta tarea.

Por lo tanto se debe tener en cuenta qué fases del proceso se considerarán para la realización de pruebas a lo largo del desarrollo del proyecto.

El producto. Para el caso de las aplicaciones web, debido a su naturaleza serán utilizadas por todo tipo de usuarios y estarán expuestas a las amenazas que se encuentran en la red [15]; esto implica que habrá información sensible o privada de clientes y de las empresas. Por tal motivo, se debe identificar y clasificar el tipo de información que manejará la aplicación web, pues el alcance y los esfuerzos de las pruebas y las actividades de calidad dependerán de si se trata de una aplicación a través de la cual se realizan transacciones electrónicas o se hace una suscripción a un portal.

Políticas internas y normatividad. Otro aspecto muy importante y que no es tenido en cuenta muy a menudo, es cómo afecta la jurisdicción el manejo de la información, pues como se sabe muchos países han implementado leyes que sancionan el mal manejo de datos personales y protegen la privacidad de los datos de las personas, por tanto se debe tener cuidado y mucha atención a los requerimientos legales y a la normatividad vigente, pues esto influye fuertemente en el cuidado que se debe dar a los datos y los pactos de confidencialidad que se debe tener con los clientes.

C. Técnicas de pruebas de seguridad

La guía para pruebas de OWASP [15] define cuatro principales técnicas para la realización de pruebas de seguridad en las aplicaciones web:

Inspecciones manuales y revisiones. Consiste principalmente en hacer entrevistas, revisar manuales, políticas, documentos de requerimientos y diseño, entre otros. No se limita a hacer pruebas al proceso, sino también a las personas, ya que de esta manera se puede obtener información que puede no encontrarse en documentos [15] y también evidenciar el nivel de entendimiento que tiene el recurso humano respecto de las políticas o requerimientos.

Es importante realizar las inspecciones al principio del ciclo de desarrollo, pues el objetivo esencial es encontrar debilidades que pueden tomar un mayor nivel de impacto si no se solucionan al inicio del proceso, y que no son evidentes en las últimas etapas del desarrollo.

Se encuentran también algunas desventajas: consume tiempo(generalmente se trabaja contrarreloj debido a los cronogramas) y recurso humano, que además requiere de habilidades específicas para ejecutar estas tareas [15].

Modelamiento de amenazas. Es la aplicación de gestión de riesgos en aplicaciones, OWASP recomienda la NIST 800-30 [15] y The OWASP Code Review Guide Outlines and Application Threat Modeling Methodology [15]. Permite encontrar posibles amenazas e identificar vulnerabilidades; enfocándose en la gestión de riesgos y en la descomposición de requerimientos de la aplicación (conocer sus funcionalidades, activos, lógica de negocio, objetivos, etc.) para crear estrategias de mitigación.

También se debe realizar en las primeras etapas del ciclo de desarrollo, ya que se basa en el análisis de requerimientos, identificación de debilidades y administración de riesgos.

Revisión de código. Varias de las vulnerabilidades no se pueden hallar sin mirar el código fuente y no se pueden encontrar a través de otras técnicas [15], es necesario hacer pruebas de caja blanca también, brindando una ventaja no solamente en la seguridad de las aplicaciones sino también en la calidad del software y en la implementación de buenas prácticas de desarrollo.

Aspectos como debilidades en algoritmos criptográficos, problemas de concurrencia, lógica de negocio, problemas de control de acceso, tipos de datos, etc. [8, 15] se evalúan a través de esta técnica con gran precisión. También es importante tener en cuenta que el código esté actualizado y se realice la revisión con la última versión que concuerde con el que se esté utilizando para el despliegue, de lo contrario se pueden encontrar inconsistencias entre el código y el resultado de la revisión. Por otro lado, se requiere de gran conocimiento en desarrollo para analizar el código, y en ocasiones este no está al alcance de los encargados de las pruebas, por tanto se debe tener especial cuidado al utilizar esta técnica apoyándose de herramientas de análisis estático de código [12, 15], pues las fuentes pueden estar en constante cambio y con errores de compilación.

Pruebas de penetración. También conocidas como hacking ético, las pruebas de penetración (o intrusión) consisten en encontrar vulnerabilidades en la aplicación sin conocer su funcionamiento interno [15]. Se requiere que el encargado de las pruebas tenga conocimientos sobre seguridad informática y habilidades para actuar o pensar como un atacante para encontrar las vulnerabilidades. Se puede utilizar también como técnica para pruebas de regresión de vulnerabilidades encontradas y arregladas [8, 15]. Se diferencia con las pruebas de intrusión en redes en que en estas se busca explotar vulnerabilidades conocidas dependiendo de la tecnología, pero como las aplicaciones web suelen desarrollarse a la medida y son muy variables, es un proceso de conocimiento e investigación continuo.

IV. IMPACTO DE APLICACIONES WEB INSEGURAS

Muchas empresas actualmente no cuentan con planes de contingencia ante ataques de seguridad informática, y no es raro que aquellas dedicadas al desarrollo de software no vean la necesidad de implementar medidas para proteger la información, ya que se suele tener la creencia de que su tarea se limita a crear un producto que es ajeno a ataques de seguridad informática, pues tiende a verse como un asunto de infraestructura y redes de telecomunicaciones, y que no concierne al mundo de la programación.

A. En el entorno empresarial

A pesar de que la mayoría de empresas no ven la importancia de implementar prácticas e incluir requerimientos de seguridad (ya sea porque los clientes no los solicitan o la propia empresa no los define), los riesgos informáticos están siempre presentes y pueden ocasionar grandes daños en las organizaciones. El problema está en que muchas empresas no tienen el conocimiento de cómo reflejar la baja calidad del software en los objetivos del negocio, y por tanto no se puede medir el impacto que estas situaciones tienen sobre los costos, la eficacia y la eficiencia de un proyecto.

Los costos de una aplicación con muchos errores pueden llegar a ser muy altos, ya que requiere dedicación extra de tiempo por parte de los desarrolladores para corregir los defectos, y del equipo de calidad para verificar que estos se hayan solucionado. Por otro lado, esto puede causar también atrasos en los pagos del cliente y también insatisfacción de los usuarios, lo cual puede llevar a la no realización de otros proyectos con los clientes y pérdida de la fidelidad y confianza.

También se presentan repercusiones legales cuando de software inseguro se trata, pues la responsabilidad por mal tratamiento y manejo de información de clientes y sus usuarios (bases de datos privadas y/o sensibles) puede acarrear el incumplimiento de leyes [Ley 1581 de 2012 de protección de datos personales] y por ende sanciones o multas. Esto se debe tener en cuenta más aún cuando la aplicación no es solamente para uso interno de una organización sino pública y de fácil acceso.

V. PRUEBAS DE SEGURIDAD EN UN PROCESO DE CALIDAD DEL SOFTWARE

Como se ha mencionado anteriormente, es importante que las empresas definan un proceso para el aseguramiento de la calidad del software. Estos modelos suelen estar basados en el ciclo de desarrollo del software, puesto que para evitar huecos de seguridad se debe conocer a fondo el proyecto, su alcance, objetivos y usuarios, y esto se consigue haciendo un seguimiento continuo en todo el ciclo, verificando que se esté llevando a cabo el proceso correctamente, se cumplan los tiempos, el equipo de desarrollo sea eficiente y competente y que se apegue a los requerimientos del cliente.

Así como la calidad, la seguridad es un proceso continuo, no un producto [15]; por tanto está sujeto a la mejora continua y a la retroalimentación.

A. OWASP

Actualmente no existe gran variedad de modelos para pruebas de aplicaciones web seguras; sin embargo, el proyecto OWASP ha estado investigando al respecto y de esto surge la Guía de pruebas de OWASP [15], la cual presenta una serie de buenas prácticas para realizar pruebas de seguridad en las aplicaciones web e identificar vulnerabilidades que pueden aparecer desde las primeras fases del ciclo de desarrollo. Se diferencia de la Guía de prácticas de desarrollo [16], pues la tarea de realización de pruebas difiere de la de desarrollo, pero también requiere de un conocimiento y experiencia específica, al igual que tiene una gran responsabilidad y participación en el producto final.

La metodología de pruebas de OWASP se basa en las distintas categorías de las vulnerabilidades de las aplicaciones web [15], y sobre esto se construye el marco que se presenta en la guía. El enfoque de OWASP, además de ser orientada a las pruebas de seguridad en todo el ciclo de desarrollo, también se centra en las pruebas de intrusión o caja negra; que es la manera como la mayoría de atacantes puede tener acceso a la aplicación (teniendo en cuenta que obtener el código fuente no es una tarea fácil, pudiendo explotar vulnerabilidades del software).

B. ¿Qué se debe asegurar en las fases del ciclo de aseguramiento de la calidad del software?

Como se mencionó anteriormente en los ciclos de desarrollo se encuentran comúnmente las siguientes fases: análisis de requerimientos, diseño, implementación, verificación y despliegue y mantenimiento. La pregunta es, ¿el proceso de calidad del software debe seguir este ciclo? La respuesta, aunque sea obvia, no es usualmente implementada en las empresas, pues muchas no cuentan con un proceso para el aseguramiento de la calidad del software definido, y mucho menos para cumplir con los requerimientos de seguridad de sus aplicaciones.

Partiendo de la premisa de que el aseguramiento de la calidad y la comprobación de la seguridad en las aplicaciones web deben hacerse desde las primeras fases del ciclo de desarrollo, se describirán las fases más comunes en un ciclo de desarrollo de software SDLC (Systems Development Life Cycle) [8, 15].

1) Fase de análisis de requerimientos. En primera instancia se debe revisar la documentación que concierne al proyecto, como requerimientos, políticas (de seguridad y desarrollo), estándares y metodologías que se deben utilizar a lo largo del proceso. Es importante tener claro qué está establecido y con qué se cuenta, para así aplicarlo de la manera adecuada y evitar que alguna actividad se pase por alto por falta de conocimiento de los soportes para realizarlas.

En cuanto a los requerimientos, estos deben ser evaluados minuciosamente, pues se debe definir el contexto en que se desarrollarán, los alcances, costos, tiempos, recursos humanos y posibles dificultades. A pesar de que los requerimientos de seguridad (así como los requerimientos no funcionales [8]) a menudo no son definidos, estos deben identificarse y especificarse, pues todas las aplicaciones web precisan un mínimo de aspectos de seguridad informática en su desarrollo. Entre estos se encuentran [15]:

- Administración de usuarios y autenticación
- Confidencialidad de los datos
- Criptografía
- Disponibilidad de la aplicación
- Integridad de los datos
- Manejo de sesiones y variables en sesión
- Ejecución de código en el cliente
- Privacidad de los datos

Por otro lado, es importante priorizar requerimientos, definir cronogramas (que incluyan las actividades de calidad) y métricas que permitan hacer seguimiento de la eficacia del proceso.

2) Fase de diseño. Habiendo esclarecido ambigüedades e inconsistencias en los requerimientos en la fase anterior, ahora se deben planificar las pruebas que se realizarán en la fase de verificación; esto incluye definir el alcance y los involucrados en el proceso de desarrollo, identificar los posibles riesgos y su mitigación y convenir la intención de los encargados de las pruebas con los requerimientos de seguridad.

En esta fase también se debe revisar la seguridad de la arquitectura y el modelo de datos de la aplicación [2], pues estas son las bases tecnológicas sobre las cuales esta fase se despliega y por tanto requiere de un análisis profundo, en el cual se permitan identificar vulnerabilidades en la relación de los datos y en el flujo de información a través de la aplicación. Se deben tener en cuenta aspectos como:

- Patrones de arquitectura
- Interfaz de usuario
- Reutilización de componentes
- Estructura de datos
- Infraestructura tecnológica
- Herramientas de desarrollo y soporte
- Sistema operativo

3) Fase de implementación. La ejecución de esta fase corresponde al momento en el que se está desarrollando la aplicación. Lo ideal sería que se implementaran los requerimientos tal cual se definieron, pero se sabe que en el mundo real del desarrollo de software se debe estar abierto y preparado para los cambios, tanto en los diseños como en los requerimientos [3]. Para estar preparado para los cambios, se debe conocer claramente el objetivo de negocio que persigue el desarrollo de la aplicación web, el entorno en que se ejecutará y los posibles ataques de seguridad informática a los que se puede ver amenazada; también se debe tener conocimiento de la documentación para poder realizar los cambios pertinentes con el menor nivel de dificultad posible.

Entre las principales actividades que se realizan en esta fase respecto a las pruebas de seguridad se encuentran:

Inspecciones de código. Aplicando la técnica Revisión de código (sección 2.3) [15] se pueden encontrar vulnerabilidades que no son fáciles de identificar haciendo pruebas de caja negra. Existen herramientas que permiten realizar el análisis estático de código [3] que permiten identificar tempranamente debilidades y malas prácticas en el código y llevar un seguimiento de estas [15].

Revisión de estándares. Es importante cumplir con las políticas o estándares de desarrollo establecidos; por ejemplo el manejo de variables en sesiones, los algoritmos criptográficos, el manejo de flujos condicionales, entre otros.

Elaboración de casos de prueba. Los casos de prueba permiten llevar a cabo el proceso de comprobación de una manera ordenada y trazable, pues teniendo en control sobre los posibles flujos de la aplicación es más sencillo verificar errores encontrados y evitar pasar por alto defectos por no poder reproducirlos.

4) Fase de verificación. En esta fase se ejecutan los casos de prueba [3] y se realizan las pruebas de intrusión (sección 2.3) [15]. Es acá cuando se evidencian los resultados de las pruebas realizadas en las fases anteriores, pues a pesar de que las fallas en el software nunca serán inexistentes [7, 9], el número de defectos encontrados debe ser menor al igual que de las vulnerabilidades. Es importante reportar y hacer seguimiento de los errores encontrados, pues a partir de estos se obtendrán métricas que serán de gran apoyo en el momento de medir la eficacia de las pruebas realizadas.

5) Fase de mantenimiento. Cuando se han llevado a cabo todas las pruebas de las fases anteriores, y se obtiene el visto bueno por parte del equipo de calidad del producto terminado, será entregado al cliente y se comenzará con el proceso de implantación.

Con la aplicación web en producción, se deben tomar medidas para asegurar la disponibilidad y confiabilidad de esta, al igual que la integridad y confidencialidad de los datos que maneja. Para esto se ve necesario realizar revisiones periódicas tanto de la infraestructura tecnológica como de la lógica del negocio, al igual que de los cambios realizados y las mejoras propuestas por el equipo de calidad.

Este proceso de comprobación de seguridad en aplicaciones web es constante, se debe mantener una mejora del proceso y del producto continuo, de manera que se alcance un alto nivel de estabilidad y estandarización que permita brindar un mejor servicio.

C. Influencia del proceso de calidad en el desarrollo /Seguridad y calidad del software

Sin calidad del software no hay seguridad [8], esto indica que si se toma la calidad del software como el proceso que debe acompañar cada fase del desarrollo de aplicaciones para asegurar que el producto sea fiel a los

requerimientos (funcionales y no funcionales) y diseños, cumpla con los estándares definidos y presente el mínimo de errores (el software libe de errores en un 100% no existe) en producción y los más altos niveles de disponibilidad, confidencialidad e integridad.

Se suele relacionar la calidad del software con la seguridad, pues los encargados de las actividades de QA los que dan el visto bueno a las aplicaciones finalizadas y deciden si están listas (cumplen los requerimientos funcionales y no funcionales, están "libres" de errores y se

		Correctness	Reliability	Efficiency	Integrity	Usability
App.	Credential Theft				**	
	Functional Manipulation	**	*		*	*
	Data Theft/Manipulation	*	*		**	*
	App. Denial of Service		**	*	*	
Plat.	Unauthorized Admin. Access				**	
	System Denial of Service		**	*		*
	Application Modification	**	*	*	*	
Net.	Network Denial of Service		**	*		*
	Network Exposure/Manipul.	*	**	*	**	
	Network Credential Theft				**	

Fig 1. Riesgos de seguridad y su impacto en aspectos de calidad [13]

sometieron a un proceso de pruebas) para salir a producción. Por tanto es de gran importancia que antes de que una aplicación web se publique y sea utilizada por sus usuarios, se asegure que no se infiltrará información ni será víctima fácil de ataques. El objetivo es que se encuentren los errores y vulnerabilidades (y se hagan los ajustes pertinentes) a tiempo y no que los usuarios o los atacantes las encuentren y se aprovechen negativamente de ellas.

Si bien es tarea del equipo de calidad realizar las pruebas para encontrar vulnerabilidades en las aplicaciones web, también es responsabilidad de los desarrolladores conocer qué prácticas evitan la aparición de fallas en el código y cómo implementar las soluciones teniendo en cuenta los requisitos de seguridad. Por esta razón, tanto los desarrolladores como los encargados de las pruebas deben conocer las debilidades que se presentan en las prácticas de programación, como identificarlas y corregirlas antes de que se transformen en una amenaza de gran impacto para la aplicación web y para el negocio.

En la Figura 1 [13] se muestra la relación entre los aspectos de calidad del software y los riesgos de seguridad.

De acuerdo al modelo de calidad de software McCall [8], la exactitud, confiabilidad, eficiencia, integridad y usabilidad son los principales criterios para determinar la calidad del software. Esto en contraste con los riesgos de seguridad informática presentes en toda aplicación web, resultan en un impacto negativo para los datos y los procesos de negocio (en la Figura 1, dos asteriscos indican daño irreparable). Por ejemplo, dado el caso de robo de información, se ve afectada la exactitud, fiabilidad e integridad de la información y de la aplicación, por lo cual no se estaría cumpliendo con los aspectos básicos de la calidad del software.

Se deben tener en cuenta todos los niveles que construyen una aplicación al momento de revisar la seguridad de la misma: la aplicación como tal, la plataforma tecnológica sobre la cual está construida, la interconexión con otras aplicaciones e internet y los estándares o marcos que rigen el proceso de desarrollo.

VI. APORTES DE LAS PRUEBAS DE SEGURIDAD EN LAS APLICACIONES WEB EN EL ÁMBITO ORGANIZACIONAL

El proceso aseguramiento de la calidad del software debería ser implementado en todas las empresas que gestionan, monitorizan y ejecutan proyectos de desarrollo de aplicaciones (web, móviles y/o escritorio), dado que están expuestas a fallas y ataques de seguridad ya sea por errores en el código, ambigüedad de los requerimientos, uso incorrecto por parte de los usuarios, falta de un proceso de pruebas consistente, no definición de requisitos de seguridad y un sinfín de causas que pueden terminar en pérdidas económicas y de clientes.

No es tarea fácil dimensionar las consecuencias que puede acarrear a una empresa dedicada al desarrollo de software el hecho de no aplicar prácticas, metodologías o procesos para aplicar seguridad informática a sus productos, pues como tal se presta un servicio al cliente en el que en ocasiones no se ve la necesidad de implementar una estrategia que permita obtener el máximo de beneficios ya sea por:

- Disminución de sobrecostos por fallas en el software
- Disminución de costos por tiempo extra dedicado por parte del personal
- Cumplimiento normativo y legal (contratos, acuerdos de confidencialidad, leyes, etc.)
- Reconocimiento en el mercado
- Ventaja competitiva
- Fidelidad de los clientes
- Ventaja de comercialización
- Crecimiento organizacional

Por estas razones es importante que las empresas adopten un modelo de calidad de software orientado a la seguridad de información, pues le conllevará ganancias de todo tipo para estructurar los procesos de negocio, aumentar su preferencia y posicionamiento en el mercado y de esta manera obtener nuevos clientes y conservarlos, alcanzar certificaciones de reconocimiento a nivel mundial y contribuye a la expansión del negocio (ofrecer más productos y servicios) y el crecimiento de la empresa, ya que refuerza las actividades de la organización y sus objetivos.

No son muchas las empresas de desarrollo de software las que implementan procesos de seguridad de información para mejorar la calidad de sus productos, puesto que no hay un alto nivel especialización en el tema y la implementación de estas prácticas requiere de capacitaciones, recurso humano, recurso tecnológico e inversión económica, y el obstáculo está en que muchas empresas no ven la necesidad ni la utilidad de tener en cuenta los aspectos de seguridad de información a la hora de desarrollar aplicaciones web, ya que como se mencionó anteriormente, se suele percibir como un asunto de infraestructura de telecomunicaciones y redes, y no del desarrollo de software.

La mejora en la implementación de estos modelos de calidad y prácticas permiten avanzar a la organización a un nivel tal que incluso en ocasiones se puede prestar como un servicio, brindando apoyo, consultoría y/o seguimiento a otras empresas que no tengan implantado un proceso de

calidad y no lleven a cabo pruebas de seguridad (ya sea que no cuenten con el conocimiento necesario o el recurso).

VII. DISCUSIÓN

Tanto la seguridad de información como el aseguramiento de la calidad del software son temas muy importantes a la hora de desarrollar aplicaciones web; sin embargo, es difícil que las empresas que construyen software unifiquen estos dos aspectos al momento de llevar a cabo sus proyectos, comenzando por la definición de un proceso de calidad que tenga en cuenta todas las fases de desarrollo.

Previos estudios se han realizado sobre el tema, y existen marcos de trabajo y modelos que facilitan la identificación de vulnerabilidades y el manejo de incidentes; sin embargo, el cómo y el por qué implementar este tipo de medidas en las empresas de desarrollo requiere mayor estudio.

Por tanto se ve necesario un aporte donde se expongan las principales vulnerabilidades de las aplicaciones web en cuanto a seguridad de información y sus posibles consecuencias y la definición de un proceso de calidad completo que tenga en cuenta la seguridad de información, así como un análisis del impacto que puede tener en la calidad del software un mal manejo de los procesos y de los aspectos de seguridad, para finalmente concluir que es necesario para las empresas implementar un proceso de aseguramiento de calidad abarcando la seguridad de información; esto de una manera clara donde se identifiquen los puntos críticos y se desarrollen las ideas ordenadamente.

VIII. CONCLUSIONES

El alcance de las pruebas realizadas sobre las aplicaciones web no se deben limitar a la lógica del negocio y a los requerimientos funcionales, sino también deben tener en cuenta los aspectos de seguridad que van de la mano con la calidad del software, pues en lo que respecta a seguridad de información, es importante verlo como un todo dentro de los procesos de la organización, y si se implementan las mejores prácticas de desarrollo y de pruebas de software es necesario crear también una conciencia de los riesgos a la información y a la empresa, e interpretar los errores en el software como impacto a los objetivos estratégicos de la organización.

Por tanto son fundamentales los procesos de calidad del software y seguridad de información en las aplicaciones web desde el inicio de un proyecto y a través de todas las fases de desarrollo sin excepción, a fin de asegurar el cumplimiento del proceso, de los requerimientos y de los objetivos de negocio que se persiguen, de manera que los incidentes serán evidentes en etapas tempranas y podrán ser corregidos a tiempo, evitando esfuerzo extra de los desarrolladores por modificar código constantemente y de los encargados de calidad por repetir el proceso de pruebas; teniendo en cuenta que es más costoso corregir un error cuando ya está en manos del cliente y a medida que se cambia el código aumenta el riesgo de originar más errores. Es más sencillo encontrar y subsanar incidencias en el transcurso del proceso de desarrollo, puesto que a medida que avanza un proyecto son más perceptibles las debilidades y la incertidumbre respecto a su enfoque, lo cual puede identificarse y evitar dedicación extra de tiempo, retrasos en los cronogramas y finalmente incumplimiento al cliente.

En el marco de un proceso de aseguramiento de la calidad las empresas deben buscar definir el alcance de las pruebas de seguridad, los roles del equipo de calidad del software, las responsabilidades de los desarrolladores y las actividades respecto a los requisitos de seguridad de información, incorporando estos aspectos dentro de las pruebas de software, y no solamente cuando el cliente lo solicite. Es importante que las empresas dedicadas al desarrollo de software establezcan sus políticas y procesos de aseguramiento de la calidad del software, definiendo los requerimientos de seguridad con base en un análisis de riesgos, teniendo en cuenta los tiempos, recursos (tecnológicos, humanos y económicos), limitaciones y ventajas competitivas del equipo (tanto de desarrollo como de calidad) invirtiendo a su vez en capacitación; con el fin de brindar un alto nivel de integración en las pruebas y por tanto de calidad del software, que se evidenciará en la satisfacción del cliente y de los usuarios finales, así como en la disminución de costos por trabajo extra o incumplimiento de tiempos.

REFERENCIAS

[1] C. Mao, “Experiences in Security Testing for Web-based Applications” pp. 6–10, 2009.

[2] A. Noureddine and M. Damodaran, “Security in web 2.0 application development” *Proceedings of the 10th International Conference on Information Integration and Webbased Applications Services iiWAS 08*, no. c, p. 681, 2008.

[3] A. Avancini, “Security testing of web applications: A research plan” *Software Engineering ICSE 2012 34th International Conference on*, no. line 1, pp. 1491–1494, 2012.

[4] Departamento de Seguridad en Computo/UNAM−CERT, *Aspectos Básicos de la Seguridad en Aplicaciones Web*, UNAM-CERT, 2009. [En línea]. [Consultado 28 de Julio de 2013]. Disponible en: http://www.seguridad.unam.mx/documento/

[5] V. Kongsli, “Towards agile security in web applications,” *Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications - OOPSLA ’06*, p. 805, 2006.

[6] E. B. Katalinic, “Security testing of web applications,” vol. 22, no. 1, pp. 1533–1535, 2011.

[7] A. K. Dalai and S. K. Jena, “Evaluation of Web Application Security Risks and Secure,” pp. 1–4, 2011.

[8] S. Islam and W. Dong, “Security Requirements Addressing Security Risks for improving Software Quality,” no. 60673118.

[9] E. B. Katalinic, “Security testing of web applications,” vol. 22, no. 1, pp. 1533–1535, 2011.

[10] Y. Wang, W. M. Lively, and D. B. Simmons, “Software security analysis and assessment model for the web-based applications,” vol. 9, 2009.

[11] N. Patel and F. Mohammed, “SQL Injection Attacks : Techniques and Protection Mechanisms,” vol. 3, no. 1, pp. 199–204, 2011.

[12] N. Jovanovic, C. Kruegel, and E. Kirda, “Static analysis for detecting taint-style vulnerabilities in web applications” vol. 18, pp. 861–907, 2010.

[13] H. Wang and C. Wang, “Taxonomy of security considerations and software quality” vol. 46, no. 6, 2003.

[14] H. T. Le and P. K. K. Loh, “Unified Approach to Vulnerability Analysis of Web Applications,” vol. 2007, pp. 155–160, 2008.

[15] Open Web Application Security Project – OWASP Foundation, La comunidad libre y abierta sobre seguridad en aplicaciones, “OWASP testing guide v3,” 2008. [En línea]. [Consultado 1 de Agosto de 2013]. Disponible en: https://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf

[16] Open Web Application Security Project – OWASP Foundation, La comunidad libre y abierta sobre seguridad en aplicaciones, “Una Guía para Construir

[17] Aplicaciones y Servicios Web Seguros", v2, 2005. [En línea]. [Consultado 1 de Agosto de 2013]. Disponible en: https://www.owasp.org/images/b/b2/OWASP_Development_Guide_2.0.1_Spanish.pdf

Silvia Margarita Díaz Díaz, Bucaramanga. Especialista en Seguridad Informática de la Universidad Pontificia Bolivariana seccional Bucaramanga, Ingeniera de Sistemas Cum Laude de la Universidad Industrial de Santander. Analista y líder de QA en Mayasoft Ltda.