IX Jornadas de Ciencia y Tecnología
de Facultades de Ingeniería del NOA
Santiago del Estero, 3 y 4 de Octubre de 2013

Clasificación de Amenazas a la seguridad en Aplicaciones Web

Cecilia Elizabeth Gallardo[1] & Oscar Eduardo Quinteros[1] & Daniel Armando Rivas[1]

(1) Departamento de Informática, Facultad de Tecnologías y Ciencias Aplicadas, Universidad Nacional de Catamarca.
ceciliagallardo@tecno.unca.edu.ar, oequinteros@tecno.unca.edu.ar, darivas@tecno.unca.edu.ar

RESUMEN: Numerosas organizaciones brindan sus servicios o productos a través de Aplicaciones Web, haciendo uso de las ventajas que éstas proveen como la facilidad de mantenimiento, actualización y el alcance a miles de usuarios a través de la Web. La seguridad en este tipo de aplicaciones, es un factor de suma importancia ya que un ataque mal intencionado podría acarrear graves consecuencias para la Organización. Durante el desarrollo de aplicaciones web más seguras hay que ser capaz de anticipar las amenazas a los que puede ser vulnerable y esto se puede lograr mediante una clasificación de amenazas a la seguridad web, distinguiendo las vulnerabilidades y ataques que pueden llevar a comprometer seriamente, su estructura, sus datos o sus usuarios. Este trabajo se centra especialmente en esta clasificación de estas amenazas o riesgos a la seguridad web, revisando diferentes modelos propuestos por varias entidades dedicadas a la seguridad en aplicaciones web o al desarrollo de software en general. Este documento puede resultar útil para los desarrolladores de aplicaciones web, ya que se recopilan y describen los tipos conocidos de ataques o riesgos que han presentado una amenaza a aplicaciones web en el pasado y se expone la forma de prevenirlos o combatirlos.

1 INTRODUCCION

En la actualidad la World Wide Web tiene una gran influencia en nuestra vida cotidiana, la razón de esto radica especialmente a la naturaleza de la Web, que se caracteriza por la disponibilidad global y permanente, ya que cualquier persona, en cualquier parte del mundo, puede agregar información a la Web para que luego pueda ser consultada por el resto de los usuarios.
Originalmente se promovía a la World Wide Web como una gran biblioteca virtual, los primeros sitios web fueron un conjunto de páginas ordenadas jerárquicamente a partir de una página de inicio.
HTML (HyperText Markup Language), lenguaje compuesto de una serie de etiquetas o marcas que permiten definir el contenido y la apariencia de las páginas web, permite enlazar una página a otra, y un conjunto de páginas enlazadas podría ser considerado un Sitio Web.
Inicialmente lo que se compartía a través de Internet consistía en su mayoría de la información estática, pero con la aparición de la tecnología Common Gateway Interface (CGI), que permite la creación de páginas dinámicas, todo esto cambió, ya no era suficiente decir que se diseñaban Sitios Web, ahora se hizo necesario el diseño de Aplicaciones Web.

Un programa CGI es: un programa preparado para recibir y enviar datos desde y hacia un servidor web según este estándar. Normalmente se programan en C o en Perl, aunque se puede usar cualquier lenguaje de propósito general (Lujan, 2002)
Una definición de Aplicación Web según Shklar (2003) se puede citar como: por definición, se trata de algo más que sólo un sitio Web. Se trata de una aplicación cliente/servidor que utiliza un navegador web como su programa cliente, y lleva a cabo un servicio interactivo mediante la conexión con los servidores a través de Internet (o Intranet). Un sitio web simplemente proporciona el contenido de los archivos estáticos. Una aplicación web presenta contenido de forma dinámica a medida en base a parámetros de solicitud, seguimiento del comportamiento del usuario y consideraciones de seguridad.

2 PRINCIPIOS DE SEGURIDAD EN APLICACIONES WEB

La seguridad es importante en cualquier tipo de desarrollo de software, como así también en las Aplicaciones Web, siendo en este tipo de aplicaciones una necesidad más crítica, dado el gran número de usuarios que pueden acceder a las mismas, con distintos objetivos. Un ataque mal intencionado podría acarrear graves

consecuencias, que pueden ser pérdidas económicas o perdidas de confianza por parte de los usuarios, entre otros.

El intercambio de mensajes seguros entre dos entidades, implica que ambas sean seguras y que se eviten las escuchas y la modificación de los datos en tránsito. Además deben preservarse la privacidad y la seguridad de los recursos locales (Kappel, 2006).

La seguridad de las aplicaciones Web está basada en siete conceptos clave: Autenticación, Autorización, Confidencialidad, Integridad, Disponibilidad, Privacidad, No repudio.

2.1 *Autenticación*

Es el acto de verificar la identidad de alguien que puede ser una persona u otra aplicación invocando un servicio en nombre de un usuario humano. La autenticación con frecuencia se lleva a cabo a través de un mecanismo de login/password, o empleando técnicas biométricas como escaneo de huellas digitales, de iris, de retina, identificación de voz entre otras (Daswani, 2007). También la autenticación de clave pública está adquiriendo importancia.

2.2 *Autorización*

Es el acto de comprobar si un usuario tiene permiso para realizar alguna acción. La autorización puede depender de la identidad de los solicitantes y/o de atributos de éste. El uso de Listas de Control de Acceso (ACL) es una técnica muy extendida, para constituir asignaciones de privilegios de usuarios dentro de una organización. Como mínimo, una ACL es un conjunto de usuarios y un conjunto correspondiente de los recursos a los que se permite el acceso. Existen 3 modelos de ACL: Control de Acceso Obligatorio (MAC), control de acceso discrecional (DAC) y el control de acceso basado en roles (RBAC) el cual es especialmente adecuado para Aplicaciones Web escalables (Daswani, 2007).

2.3 *Confidencialidad*

Significa que los datos intercambiados entre un cliente y un proveedor no puedan ser leídos por un tercero (Kappel, 2006). La encriptación o cifrado es la base tecnológica del intercambio de mensajes confidenciales a través de canales de comunicación seguros tales como canales privados o una red privada virtual (VPN). La mayoría de las tecnologías de cifrado utiliza una clave para cifrar la comunicación. Una clave es una secuencia secreta de bits solo conocida por el emisor y el receptor del mensaje. Un algoritmo de cifrado tendrá como entrada la clave y el mensaje que se quiere transferir, se codifica el mensaje de una manera que es matemáticamente dependiente de la clave. El mensaje está codificado de tal manera que cuando un tercero vea la comunicación codificada, no será capaz de comprender su contenido. El destinatario puede utilizar la clave para descifrar el mensaje mediante el cálculo matemático inverso del algoritmo de cifrado (Daswani, 2007).

2.4 *Integridad*

Se refiere a que nadie pueda modificar la información intercambiada. El objetivo de la integridad del mensaje es asegurar que incluso si un tercero logra observar, no pueda tocar el contenido del mensaje. El propósito es al menos detectar posibles modificaciones de datos. Para asegurar la integridad se pueden emplear enfoques tales como Códigos de Autenticación de Mensajes (MAC) que están en función del mensaje y de una clave conocida por el emisor y el receptor de tal manera que incluso si un tercero es capaz de modificar los bytes de un mensaje, no será capaz de modificar apropiadamente la correspondiente MAC.

2.5 *Disponibilidad*

Es la característica, cualidad o condición de la información de encontrarse a disposición de quienes deben acceder a ella, ya sean personas, procesos o aplicaciones (Daswani, 2007). Garantizar la disponibilidad de las aplicaciones web es de importancia económica, ya que el tiempo de inactividad de un servicio típicamente implica pérdidas económicas. Un sistema disponible es el que puede responder a las peticiones de sus usuarios en un plazo razonable. Un atacante interesado en la reducción de la disponibilidad de un sistema, típicamente lanza un ataque de denegación de servicio (DoS). Si por ejemplo, un sitio web de tienda de libros en línea se ejecuta en un único servidor web, y un atacante transmite datos al servidor web para hacer que se caiga, daría lugar a un ataque de denegación de servicio en el que los clientes legítimos serían incapaces de hacer compras hasta que el servidor web se inicie de nuevo.

2.6 *Privacidad*

La privacidad exige la manipulación confiable de los datos, tales como información personal, datos de contacto o números de tarjetas de crédito, pero también los archivos almacenados en el sistema

de archivos local. Estos datos no deben ser accesibles a terceros no autorizados que podrían abusar de ellos para el robo de identidad.

2.7 *No Repudio*

El objetivo del no repudio es asegurar la innegabilidad de una transacción por cualquiera de las partes involucradas (Daswani, 2007). Para lograr esto se puede utilizar un tercero de confianza e imparcial con quien interactuarán las partes y ayudará a realizar transacciones no repudiables. En general, los protocolos de no repudio en el ámbito de la seguridad se utilizan para garantizar que ambas partes no puedan negar que interactuaban entre sí, para ello se generan varios conjuntos de pruebas, tales como recibos que se pueden firmar digitalmente para probar que la transacción tuvo lugar.

3 CLASIFICACIÓN DE AMENAZAS A LA SEGURIDAD EN APLICACIONES WEB

Durante el proceso de programación de aplicaciones más seguras hay que ser capaz de anticipar las amenazas o riesgos a los que puede ser vulnerable dicha aplicación.
Una Clasificación de Amenazas a la Seguridad Web es un gran aporte que permite para clasificar los puntos débiles y los ataques que pueden llevar a comprometer seriamente una aplicación web, sus datos o sus usuarios y es de excepcional valor para desarrolladores de aplicaciones, profesionales de la seguridad, fabricantes de software o cualquier otro interesado en la seguridad web, ya que se recopilan y desglosan los tipos conocidos de ataque que han presentado una amenaza a aplicaciones o sitios web en el pasado.
Para la confección de este trabajo, se revisó la clasificación de riesgos y amenazas realizada por tres importantes Organizaciones: Web Application Security Consortium (WASC), Microsoft y Open Web Application Security Project (OWASP), haciendo mayor énfasis en los riesgos o amenazas más relevantes determinadas por OWASP.
Con el objeto de evitar confusiones entre los conceptos utilizados en adelante, se definen brevemente cada uno de ellos (WASC. Threat-Classification-Glossary):
Amenaza: una violación potencial de la seguridad.
Impacto: consecuencias para la organización o el medio ambiente cuando un ataque es realizado, o la debilidad está presente.
Ataque: un conjunto bien definido de acciones que, si tiene éxito, podría resultar en un daño a un bien o un funcionamiento indeseable. Es la forma en la que se aprovecha una vulnerabilidad de seguridad.
Vulnerabilidad: es una ocurrencia de una o varias debilidades dentro del software, en el que la misma puede ser utilizada por una de las partes para modificar el software o acceder a datos, interrumpir la adecuada ejecución, o realizar acciones incorrectas que no se hayan concedido expresamente.
Debilidad: es un tipo de error en el software que, en condiciones adecuadas, podría contribuir a la introducción de vulnerabilidades dentro de éste.

3.1 *Clasificación de Amenazas de WASC*

WASC es una organización sin fines de lucro formada por un grupo internacional de expertos, profesionales de la industria y representantes de organizaciones que producen software libre basándose en estándares de buenas prácticas de seguridad para la World Wide Web.
Como una comunidad activa, WASC facilita el intercambio de ideas y organiza varios proyectos de la industria y constantemente libera información técnica, artículos escritos, pautas de seguridad y documentación de utilidad para empresas, instituciones educativas, gobiernos, etc.
La Clasificación de Amenazas de WASC (WASC Threat Classification) es un esfuerzo de cooperación que sirve principalmente como una guía de referencia para cada ataque y/o debilidades, incluyendo ejemplos de cada tema, así como material de referencia útil. A continuación se enumeran los ataques y debilidades más significativos que componen esta clasificación.
Ataques: Inyección SQL, Suplantación de contenido, Cross-Site Scripting, Denegación de Servicio, Tráfico ilícito de solicitud y respuesta http, Abuso de redireccionamineto URL, entre otros.
Debilidades: Aplicación mal configurada, Revelación de información, Autenticación y autorización insuficiente, Expiración de sesión insuficiente, entre otros.

3.2 *Clasificación de Amenazas según la Corporación Microsoft*

Microsoft Corporation® es una empresa multinacional, dedicada al sector del software que desarrolla, fabrica, licencia y produce software y equipos electrónicos.
Por otra parte, Microsoft pone a disposición de sus usuarios, la MSDN Library (Biblioteca de desarrollo de Microsoft), la cual es una fuente de información esencial para desarrolladores que

usan productos de Microsoft e incluye procedimientos y documentación de referencia, código de ejemplo, artículos técnicos, entre otros.
Con el objeto de identificar las áreas en las que una aplicación web es más vulnerable y luego elegir las herramientas adecuadas y aplicar el mejor diseño para protegerla, la comunidad de Microsoft ha desarrollado un modelo para clasificar las amenazas de seguridad informática denominado STRIDE (MSDN Library) (Uncover Security Design Flaws Using the STRIDE Approach) (Howard, 2002), acrónimo de Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (suplantación de identidad, manipulación, repudio, divulgación de información, denegación de servicio y elevación de privilegios).

3.3 *Clasificación de Amenazas según OWASP*

El Proyecto Abierto de Seguridad en Aplicaciones Web (OWASP) es una comunidad abierta dedicada a habilitar a las organizaciones para desarrollar, comprar y mantener aplicaciones confiables. Esta comunidad trabaja para crear artículos de libre disponibilidad, metodologías, documentación, herramientas y tecnologías.
Las herramientas y documentos del proyecto OWASP están organizados en tres categorías: Protección, Detección y Ciclo de Vida. En las dos primeras categorías, los estándares y herramientas se utilizan para proteger o encontrar fallas relacionadas con la seguridad en el diseño y la implementación. La categoría Ciclo de Vida, tiende a añadir seguridad en las actividades relacionadas con el SDLC.
La clasificación de riesgos o amenazas que ofrece OWASP se distribuye mediante el Proyecto Top 10 (Category: OWASP Top Ten Project) (The ten most Critical Web Application Security Risk), el cual representa una lista concisa y enfocada sobre los Diez Riesgos más Críticos sobre Seguridad en Aplicaciones. Los miembros del proyecto incluyen expertos en seguridad de todo el mundo que han compartido y consensuado su experiencia para producir esta lista.
La última versión del proyecto Top 10 correspondiente al año 2010, entrega información genérica para cada uno de los riesgos, sobre probabilidad e impacto técnico a través de un esquema de clasificación simple, que se basa en la Metodología de Calificación de Riesgo OWASP (OWASP Risk Rating Methodology).
OWASP define que los riesgos a los que está expuesta una aplicación, son las diferentes rutas a las que puede acceder potencialmente el atacante a través de la aplicación para causar daño en la organización.
Para determinar el riesgo de cada organización, se puede evaluar la probabilidad asociada con cada agente de amenaza, vector de ataque y debilidad de seguridad y combinarla con una estimación del impacto técnico y de negocios en la organización. Juntos, estos factores determinan el riesgo total. Ver Fig. 1

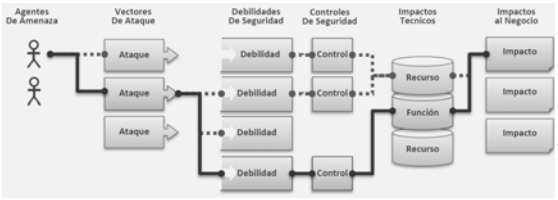

Figura 1. Diferentes rutas de acceso no autorizado a una Aplicación.

De los diez riesgos tratados en el proyecto Top 10 de OWASP (ver Fig. 2), se expondrá en este trabajo, una descripción, nivel de impacto, forma de prevención y escenario de ejemplo para cinco riesgos (A1, A2, A3, A4), mientras que para los restantes se dejará un breve descripción.

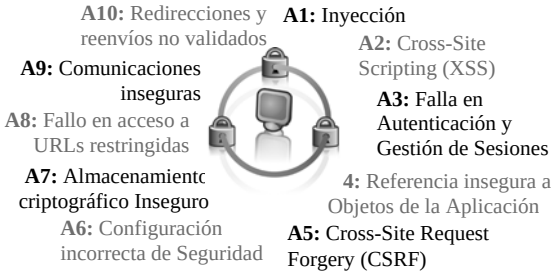

Figura 2. Los diez Riesgos del Proyecto Top 10 de OWASP

3.3.1 Riesgo A1: Inyección

Las fallas de inyección ocurren cuando se envían datos no confiables a un intérprete de comandos como parte de una sentencia o consulta tales como SQL, LDAP, OS, XPAth, para que ejecute comandos no deseados o acceder a datos no autorizados.
Nivel de Impacto: grave. La inyección puede causar la corrupción total o parcial de datos de una Base de Datos, acceso a cuentas restringidas, denegación de acceso a la aplicación, entre otros.
Formas de Prevención:

- Utilizar una interfaz que soporte variables que obliguen a un determinado formato (por ejemplo sentencias preparadas), lo cual permite la distinción entre código fuente y datos.
- Escapar cuidadosamente los caracteres especiales o adicionalmente, realizar siempre una validación de entrada tipo lista blanca a

todos los datos suministrados por el usuario y minimizar los privilegios de acceso a la B.D.

Escenario de Ejemplo: considere que la aplicación utiliza los datos no confiables ingresados por el atacante para la construcción de la siguiente sentencia vulnerable SQL:

"SELECT * FROM cuentas WHERE IDCliente='" + request.getParameter("id") +"'";

En este caso el atacante modifica el parámetro id para enviar: 'or '1' = '1, resultando la sentencia:

"SELECT * FROM cuentas WHERE IDCliente='" OR 1=1--'"

Esto cambia el significado de la consulta para devolver todos los registros de la base de datos de la Tabla Cuentas, en lugar de sólo el registro del cliente al que va dirigido.

3.3.2 Riesgo A2: Secuencia de Comandos en Sitios Cruzados (XSS: Cross-Site Scripting)

Los ataques XSS son un tipo de problema de inyección y ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, entre otros.

Existen tres tipos conocidos de fallas de XSS: almacenados, reflejados, y XSS basado en DOM (OWASP. Cross-site Scripting (XSS)).

Nivel de Impacto: Moderado. El atacante puede ejecutar scripts en el navegador de la víctima, apoderarse de las sesiones de usuario, insertar contenido malicioso, redirigir a usuarios, etc.

Formas de Prevención:

- No incluir en la página de salida los datos de entrada suministrados por el usuario.
- Escapar correctamente los datos no confiables basados en el contexto HTML (body, attribute, JavaScript, CSS, o URL).
- Realizar una validación tipo lista blanca de los datos de entrada.
- Considerar el empleo de la Política de Seguridad de Contenido definida por el W3C (W3C. Content Security Policy 1.0).

Escenario de Ejemplo:

Suponga que una aplicación utiliza datos no confiables en la construcción del siguiente código HTML sin validación o escape:

"<input name='creditcard' type='text' value='"+request.getParameter("CC") + "'>";

El atacante modifica el parámetro CC en su navegador a:

"><script>document.location='http://www.attacker.com/cgi-bin/cookie.cgi?foo='+document.cookie</script>"

Esto provoca que el identificador de la sesión de la víctima sea enviado al sitio web del atacante, permitiéndole a éste apoderarse de la sesión actual del usuario.

3.3.3 Riesgo A3: Falla en Autenticación y Gestión de Sesiones

Al ser HTTP un protocolo sin estado, se debe recurrir a un manejo de sesiones donde las credenciales de usuarios tienen que transportarse con todas las solicitudes que realice el mismo.

Los esquemas de autenticación y gestión de sesiones desarrollados en forma personaliza, con frecuencia tienen defectos en áreas como: cierre de sesión, administración de contraseñas, pregunta secreta, actualización de cuentas, etc.

Nivel de Impacto: grave. Las fallas de este tipo pueden permitir que el atacante haga cualquier cosa que la víctima pudiera hacer, apoderándose de cuentas y datos de usuarios confidenciales.

Formas de Prevención:

- La autenticación debe ser simple, centralizada y estandarizada.
- Asegurarse de que identificadores de sesión y credenciales son transportados mediante un puerto seguro como SSL.
- Cumplir con los requisitos de gestión de autenticación y sesión definidos en el Proyecto OWASP de estándares para verificación de seguridad en Aplicaciones (ASVS) (Category: OWASP Application Security Verification Standard Project), áreas V2 (Autenticación) y V3 (Manejo de Sesiones).
- Verificar que al cerrar la sesión, se destruya realmente la misma.

Escenario de Ejemplo: suponga que en una aplicación web, los tiempos de espera de sesiones no fueron ajustados correctamente y un usuario utiliza un equipo público para acceder al sitio. Luego, en lugar de seleccionar cerrar sesión, el usuario simplemente cierra la pestaña del navegador y se aleja. El atacante utiliza el mismo navegador hora más tarde, y resulta que el usuario anterior está autenticado todavía, pudiendo acceder a todos sus datos confidenciales.

3.3.4 Riesgo A4: Referencia directa insegura a Objetos

Una referencia directa ocurre cuando un desarrollador expone una referencia a un objeto interno de la aplicación, como un archivo, un directorio o claves de base de datos. Sin una comprobación del control de acceso o de otro tipo de protección, los atacantes pueden manipular estas referencias para acceder a datos no autorizados. Nivel de Impacto: moderado.

Formas de Prevención:

- Utilizar referencias de objeto indirectas por cada usuario o sesión. Para implementar esto, se puede reemplazar cada referencia de objeto, con un valor de mapeo temporal (por ejemplo, 1, 2, 3).
- Comprobar que el valor del parámetro tiene el formato correcto.
- Verificar que el usuario está autorizado a acceder al objeto de destino y que el modo de acceso solicitado está permitido para el objeto de destino (lectura, escritura, eliminación).

Escenario de Ejemplo: considere que una aplicación expone la siguiente URL: https://www.onlinebank.com/user?id=6065,
donde el atacante advierte que el ID del usuario es 6065. Luego, modifica este ID por un número cercano: ?id=6066, entonces el atacante puede ver la información de otro usuario diferente.

3.3.5 Riesgo A5: Falsificación de Peticiones en Sitios Cruzados (CSRF: Cross-Site Request Forgery)

CSRF es un ataque que obliga al navegador web de una víctima autenticada a enviar una petición HTTP falsificada, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima para generar pedidos que la aplicación vulnerable piensa son peticiones legítimas.
Nivel de Impacto: moderado. El atacante puede en nombre de la víctima, recuperar o modificar información de su cuenta, iniciar transacciones como transferir fondos entre cuentas, o cualquier otra función proporcionada por el sitio web.

3.3.6 Riesgo A6: Configuración incorrecta de Seguridad

Una buena seguridad requiere tener definida e implementada una configuración segura para la aplicación, framework, servidor de aplicación, servidor web, base de datos y plataforma. Todos estos ajustes se deben definir, implementar y mantener ya que por lo general no son seguras por defecto. Esto incluye mantener todo el software actualizado, incluidas las librerías de código utilizadas por la aplicación.
Nivel de Impacto: moderado. Una falla en la configuración de la seguridad suele dar a los atacantes acceso no autorizado a algunos datos del sistema o funcionalidad.

3.3.7 Riesgo A7: Almacenamiento criptográfico Inseguro

Muchas aplicaciones web no protegen adecuadamente los datos sensibles, tales como tarjetas de crédito, datos financieros, números de documento o credenciales de autenticación, con el cifrado o hash apropiado. También puede haber una falta de identificación de todos los datos sensibles y a su vez, de todos los sitios en los que estos datos sensibles son almacenados (bases de datos, archivos, directorios, backups, etc).
Nivel de Impacto: Grave. Se ven comprometidos todos los datos sensibles que se deberían haber sido cifrado, ya que el atacante puede acceder a ellos y/o modificarlos.

3.3.8 Riesgo A8: Falla de restricción de acceso a URL

Muchas aplicaciones web verifican los privilegios de acceso a URLs antes de generar enlaces o botones protegidos. Sin embargo, las aplicaciones necesitan realizar controles similares cada vez que estas páginas son accedidas, o los atacantes podrán falsificar URLs para acceder a estas páginas igualmente.
Nivel de Impacto: moderado. Estos defectos permiten a los atacantes el acceso no autorizado al sistema. Las funciones administrativas son los objetivos principales de este tipo de ataques.

3.3.9 Riesgo A9: Protección Insuficiente en la Capa de Transporte

Las aplicaciones frecuentemente fallan al autenticar, cifrar, y proteger la confidencialidad e integridad de tráfico de red sensible. Cuando esto ocurre, es debido a la utilización de algoritmos débiles, certificados expirados, inválidos, o sencillamente no utilizados correctamente.
Nivel de Impacto: moderado. Estos problemas exponen datos de usuarios y pueden derivar en un robo de cuenta. Si una cuenta de administrador es comprometida, la aplicación entera podría estar expuesta.

3.3.10 Riesgo A10: Re-direccionamientos y Re-envíos No validados

Las aplicaciones web frecuentemente redirigen y reenvían a los usuarios hacia otras páginas o sitios web, y utilizan datos no confiables o sin validar, para determinar la página de destino.
Nivel de Impacto: moderado. Estas redirecciones pueden intentar instalar código malicioso o engañar a las víctimas para que revelen contraseñas u otra información sensible.

4 COMPARACION DE LOS MODELOS DE CLASIFICACION DE AMENAZAS

De las clasificaciones de amenazas revisadas, desarrolladas por las organizaciones OWASP, OWASC y Microsoft, podemos decir que en

términos generales, todas cubren el espectro total en lo que a seguridad en las aplicaciones web se refiere. Microsoft realiza una clasificación más genérica en comparación con los otros dos modelos, siguiéndole la clasificación de riesgos de OWASP. Se observa que el modelo de amenazas de la organización OWASC es la más detallada de los tres modelos analizados.

Por otra parte, OWASP realiza su clasificación de acuerdo al término riesgo, el cual lo define como: las diferentes rutas a las que puede acceder potencialmente el atacante a través de la aplicación para causar daño en la organización y para determinarlo, se toman en cuenta los ataques que se puedan producir, las debilidades de seguridad en la aplicación y el impacto o consecuencias que pudieran surgir.

OWASC por su parte, diferencia claramente su clasificación de amenazas en los ataques y debilidades que pueden dar lugar a comprometer seriamente una aplicación web, sus datos o sus usuarios.

Por último, cada ítem de la clasificación de las amenazas de seguridad informática desarrollada por la corporación Microsoft (STRIDE), se corresponde uno a uno con los principios de Seguridad en Aplicaciones web, es decir, que para evitar cada amenaza, se debe aplicar el principio de seguridad correspondiente.

En la tabla 1 se puede apreciar una aproximación de correspondencia entre los riesgos de OWASP y las amenazas analizadas por OWASC y Microsoft.

Tabla 1. Comparación de clasificación de amenazas de OWASP contra OWASC y Microsoft

Riegos de OWASP	Amenazas de OWASC	Amenazas de Microsoft
Inyección	Inyección SSI, SQL, XPath, XML, XQuery, LDAP, de comandos de mail, de byte nulo	Manipulación de datos, Divulgación de información, Repudio
Cross-Site Scripting (XSS)	Cross-Site Scripting, Denegación de Servicio. Debilidades: Manipulación de entrada inapropiado	Manipulación de datos, Denegación de servicio, Repudio
Perdida de Autenti-cación y Gestión de Sesiones	Fuerza Bruta, Abuso de Funcionalidad. Debilidades: Autenticación y Autorización insuficiente, Recuperación de la contraseña y Expiración de sesión insuficiente	Suplantación de identidad, Elevación de privilegios, Divulgación de información Elevación de privilegios, Repudio
Referencia directa insegura a Objetos	Desbordamiento de buffer, de Enteros, Formateo de Strings. Debilidades: Manipulación de salida inapropiado, Fuga de información	Suplantación de contenido, Divulgación de información
Cross-Site Request Forgery (CSRF)	Cross-Site Request Forgery. Debilidad: Manipulación de entrada inapropiado	Manipulación de datos
Configura-ción de Seguridad defectuosa	Abuso de Funcionalidad, de SOAP Array, Denegación de Servicio, Fijación de sesión. Debilidades: Configuración errónea del servidor y de aplicación, Permisos incorrectos Sistema de Archivos	Suplantación de identidad, Manipulación de datos, Denegación de servicio, Elevación de privilegios, Divulgación de información, Elevación de privilegios, Repudio
Almace-namiento criptográfi-co Inseguro	Abuso de Funcionalidad	Suplantación de identidad, Manipulación de datos, Elevación de privilegios, Divulgación de información, Repudio
Falla de restricción de acceso a URLs	Denegación de Servicio, Acceso a directorios, Localización de recursos predecibles, Explotación de atributos XML, Expansión de entidades XML. Debilidad: Manipulación de entrada inapropiado	Denegación de Servicio, Divulgación de información, Repudio
Protección insuficiente en capa transporte	Denegación de Servicio, Inclusión de archivos remotos (RFI). Debilidad: Protección de la capa de transporte insuficiente	Elevación de privilegios, Divulgación de información
Redireccio-nes y reenvíos no validados	Tráfico ilícito de la solicitud y respuesta http, División de la solicitud y respuesta http, Abuso de redireccionamineto URL. Debilidades: Manipulación de entrada y salida inapropiado	Manipulación de datos, Divulgación de información

5 CONCLUSIONES

En el desarrollo de este trabajo hemos analizado distintas clasificaciones de categorías de las amenazas o riesgos que pueden afectar las Aplicaciones Web, según distintas Organizaciones dedicadas a tal fin. Hemos destacado la importancia de conocer los problemas de vulnerabilidades, ataques y amenazas que las podrían afectar, para poder aplicar las técnicas que mejoren su seguridad.

Los diferentes aspectos de la seguridad son causados por la complejidad de las aplicaciones Web, especialmente por el hecho que están potencialmente al alcance de miles de usuarios a través de Internet y que la comunicación avanza a través de canales de acceso público (inseguros).

De las clasificaciones de amenazas revisadas provenientes de distintas Organizaciones, se observa que en general se distinguen los mismos riesgos o amenazas, pero se diferencian en la forma de agrupar los mismos, siendo en algunos casos más o menos específicos.

Aunque no existe un esquema de seguridad que cubra todos los riesgos, se debe estar preparado y trabajar constantemente ante cualquier nueva amenaza, en pos de una Aplicación Web segura.

6 REFERENCIAS

Lujan, M.S., *Programación de Aplicaciones Web: Historia, Principios Básicos y Clientes Web*, Club Universitario, España, 2002.

Shklar, L. & R. Rosen, *Web Application Architecture. Principles, protocols and practices*, John Wiley & Sons, England, 2003

Kappel, G. & B. Pröll & S. Reich & W. Retschitzegger, *Web Engineering. The Discipline of Systematic Development of Web Applications*, John Wiley & Sons, Germany, 2006

Daswani, N. & C. Kern & A. Kesavan, *Foundations of Security: What Every Programmer Needs to Know*, Apress, United States of America, 2007

Web Application Security Consortium, http://www.webappsec.org/, 1 de Julio de 2013.

MSDN Library. Información general sobre las amenazas para la seguridad de las aplicaciones web, http://msdn.microsoft.com/es-es/library/f13d73y6%28v=vs.100%29.aspx, 1 de Julio de 2013.

Open Web Application Security Project, https://www.owasp.org, 1 de Julio de 2013.

Web Application Security Consortium. Threat-Classification-Glossary, http://projects.webappsec.org/w/page/13246980/Threat-Classification-Glossary, 1 de Julio de 2013.

Web Application Security Consortium. WASC Threat Classification. Version 2.00, https://files.pbworks.com/download/57CMSh64fh/webappsec/13247059/WASC-TC-v2_0.pdf, 1 de Julio de 2013.

MSDN Magazine. Uncover Security Design Flaws Using the STRIDE Approach, http://msdn.microsoft.com/en-gb/magazine/cc163519.aspx, 1 de Julio de 2013.

Howard, M. & D. Leblanc, *Writing Secure Code*, Microsoft Press, United States of America, 2002

Open Web Application Security Project. Category:OWASP Top Ten Project, https://www.owasp.org/index.php?title=Category:OWASP_Top_Ten_Project&setlang=es, 1 de Julio de 2013.

The OWASP Fundation, *OWASP Top 10 – 2010. The ten most Critical Web Application Security Risk*, Licencia: Creative Commons Attribution ShareAlike 3.0, 2010

Open Web Application Security Project. OWASP Risk Rating Methodology, https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology, 1 de Julio de 2013.

Open Web Application Security Project. Category: OWASP Enterprise Security API, www.owasp.org/index.php/ESAPI, 1 de Julio de 2013.

Open Web Application Security Project. Cross-site Scripting (XSS), https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29, 1 de Julio de 2013.

Open Web Application Security Project. DOM Based XSS, https://www.owasp.org/index.php/DOM_Based_XSS, 1 de Julio de 2013.

World Wide Web Consortium (W3C). Content Security Policy 1.0, http://www.w3.org/TR/CSP/, 1 de Julio de 2013.

Open Web Application Security Project. Category: OWASP Application Security Verification Standard Project, https://www.owasp.org/index.php/ASVS, 1 de Julio de 2013.

Open Web Application Security Project. Category: OWASP CSRFGuard Project, https://www.owasp.org/index.php/CSRFGuard, 1 de Julio de 2013.