Efectividad de OWASP Para Proteger Applicaciones Web Contra Inyección de SQL

Manuel Lopez Arredondo

*OWASP Guadalajara*

manuel.lopez@owasp.org

**Keywords**: vulnerabilidades, aplicaciones web, seguridad, inyección SQL, OWASP

# ABSTRACT

# El presente documento muestra un estudio del nivel de riesgo al que se encuentran expuestas todo tipo de instituciones, empresas privadas y gobiernos por medio de sus aplicaciones web, ya sea sitios internos (intranet) o externos (internet), mediante la explotación de las vulnerabilidades de inyección de SQL, así como del nivel de efectividad de las técnicas de OWASP (Open Web Application Security Project) para mitigar ésas vulnerabilidades.

Por medio de la ejecución de pruebas de penetración enfocadas a explotar las vulnerabilidades mencionadas, y con base a una evaluación de las prácticas de desarrollo seguro de software de la organización objetivo, tendremos un panorama inicial del estado de la seguridad de la aplicación, y obtendremos el nivel de madurez de las prácticas de desarrollo de software seguro de la organización; para después, implementar las técnicas de OWASP, realizar las mismas evaluaciones, e identificar si éstas técnicas son efectivas mediante la comparación de resultados obtenidos. Lo anterior con la finalidad de ayudar a los profesionales de Seguridad de la Información, Desarrolladores de Software, y Organizaciones en general, a tomar decisiones más informadas para mitigar riesgos de integridad y confidencialidad de la información en aplicaciones web.

# INTRODUCCIÓN

En un estudio del SANS Institute se menciona que el 60% de los ataques en Internet se hacen contra aplicaciones Web (Scholt, Balzarotti, & Kirda, 2012); en consecuencia, cualquier organización está expuesta al robo de datos e incluso a problemas de integridad de los mismos, llevando a la organización a enfrentar riesgos regulatorios, de reputación y legales. En el caso de México, un ejemplo de un riesgo legal es el no cumplimiento de la Ley Federal de Protección de Datos en Posesión de Particulares (LFPDPPP), que describe como motivo de incumpliendo con la regulación, el hecho de “Vulnerar la seguridad de bases de datos, locales, programas o equipos, cuando resulte imputable al responsable”; las sanciones van desde 100 hasta 320,000 días de salario mínimo (Diario Oficial de la Federación, 2010).

Los ataques de Inyección de SQL son realizados al enviar código malicioso sobre aquellos campos que se encuentran a la vista del usuario; vía la sintaxis del código enviado, se puede explotar al intérprete de base datos. Por ejemplo, existen combinaciones de caracteres que pueden tener un significado especial para el manejador de la base de datos ya que estos caracteres son parte del lenguaje del manejador de base de datos. Los siguientes caracteres son un ejemplo de estos caracteres especiales:

* ‘ o “ (caracteres que indican cadenas de texto)
* # (línea simple comentada)
* /\* … \*/ (varias líneas comentadas)
* + (suma o concatencación)
* % (indicador para varios caracteres)

Al enviar a la aplicación una combinación de los caracteres arriba mostrados, un atacante puede tener acceso para consultar o manipular la información, al enviar comandos de SQL.

El Proyecto Abierto de Seguridad en Aplicaciones Web (OWASP por sus siglas en inglés) provee algunas soluciones para fortalecer la seguridad en aplicaciones web y mitigar el riesgo de que la inyección de SQL sea encontrada y explotada.

El propósito de éste estudio de caso es trabajar en la implementación de las técnicas OWASP en organizaciones mexicanas realizando las siguientes tareas:

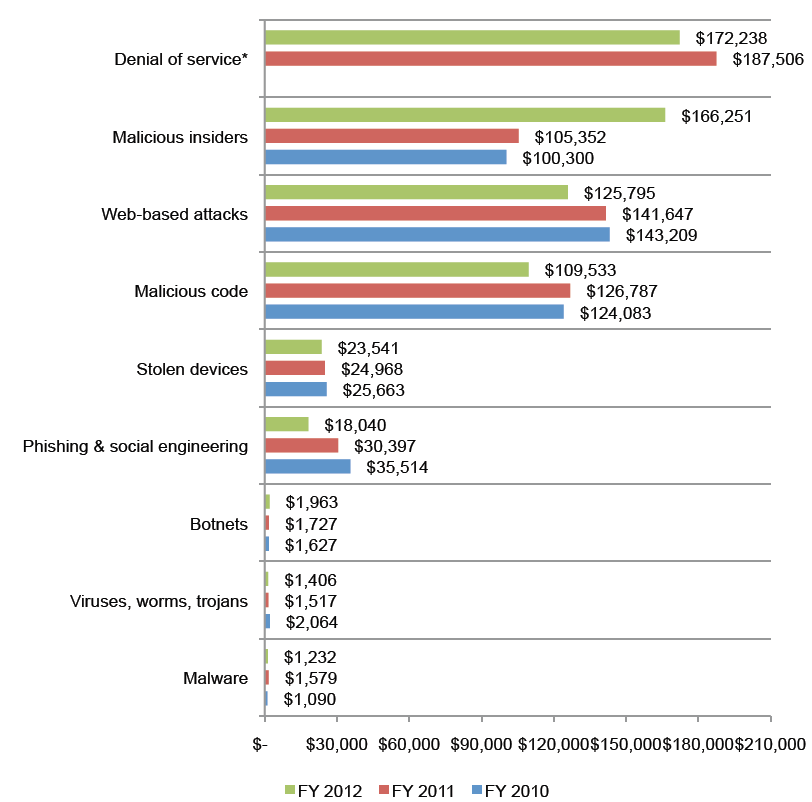
Capacitar a las organizaciones en la implementación de Técnicas OWASP ya sea en línea o en sitio.

* Identificar las aplicaciones dentro del alcance por cada organización.
* Implementar las técnicas de OWASP en las aplicaciones dentro del alcance.
* Medir los resultados mediante:
  + Indicadores clave de riesgo (KRI por sus siglas en inglés): Número y nivel de riesgo de las vulnerabilidades encontradas por un ejercicio de prueba de penetración, antes y después de la implementación de las técnicas OWASP.
  + Evaluación del nivel de madurez de la organización para el desarrollo seguro de software, antes y después de la implementación de las técnicas OWASP.

1. ANTECEDENTES

Actualmente, el costo total de un ataque cibernético va desde $3,252,912 USD hasta $8,933,510 USD (Ponemon Institute, 2012). La siguiente figura muestra el costo anual promedio por ataque donde podemos ver que los “Web-based attacks” se encuentran en la tercer categoría de ataque con mayores costos para las compañías.

Figura 1 Costo anual promedio de un ciber ataque (Ponemon Institute, 2012)

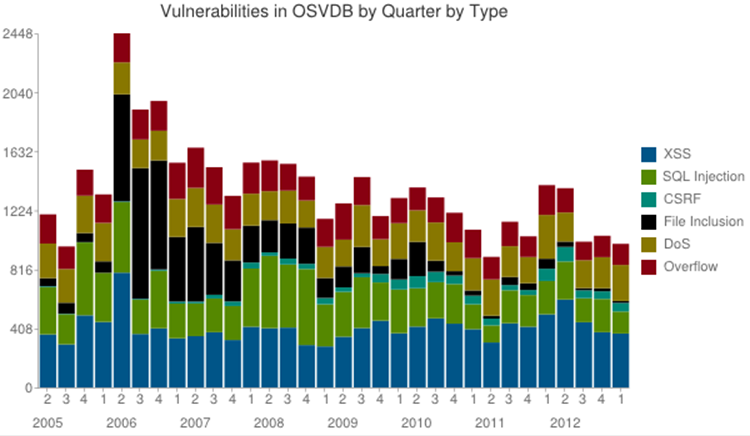


El estudio contempló entre otro tipo de ataques las inyecciones de SQL dentro de la categoría “Web-based attacks”.

Las vulnerabilidades en las aplicaciones web siguen siendo populares debido a la falta de atención por parte de las organizaciones y desarrolladores de software para evitarlos (Hewlett-Packard, 2012). Una de las vulnerabilidades más comunes encontrada durante 2012 fue la de inyección de SQL.

La siguiente figura ilustra las vulnerabilidades reportadas por trimestre desde 2005 hasta 2012 demostrando que las vulnerabilidades de inyección de SQL han sido una de las más constantes a lo largo de los años (The Open Source Vulnerability Database, 2012).

Figura 2 Vulnerabilidades por trimestre 2005-2012 (The Open Source Vulnerability Database, 2012)



Sin embargo, el universo de los ataques a aplicaciones web es grande y no sería posible evaluar todos ellos como alcance de éste estudio de caso. Por lo anterior, nos enfocaremos en el ataque más común y con más nivel de riesgo en aplicaciones web: la inyección de SQL (Williams & Wichers, 2013).

No resulta claro cuándo fue la primera vez que un ataque de SQL se presentó, sin embargo el crédito se ha dado a Rain Forest Puppy quien en 1998 hizo público cómo funcionan estos ataques al describirlos en su artículo (Puppy, 1998). Desde entonces y como hemos analizado anteriormente, ha sido uno de los ataques más constantes, provocando pérdida de información personal y daños reputacionales a organizaciones como lo señala (Clarke, 2012) en diversos ejemplos que proporciona en su estudio.

## La Fundación OWASP

La fundación OWASP surgió en 2001 debido a la necesidad de contar con un estándar de la industria de la seguridad para desarrollar una metodología para hacer pruebas de seguridad en aplicaciones web. Sus miembros desde un inicio y hasta la fecha hacen aportaciones voluntarias ya sea con fondos económicos, desarrollando proyectos, software y documentación en nombre de la fundación, así como en la difusión del material mencionado.

OWASP es una fundación no lucrativa cuyo principal propósito es proporcionar a la industria de la seguridad información material que ayude a las organizaciones y profesionales a tomar decisiones informadas cuando haya que aplicar controles de seguridad en aplicaciones web.

El material desarrollado por la fundación es libre y el código es abierto, de tal forma que puede ser obtenido por cualquier persona e incluso puede ser mejorado dependiendo de las necesidades de cada organización. El software generado por OWASP se puede distribuir bajo la licencia de software libre.

OWASP está influenciado por otros estándares internacionales de seguridad de la información como COBIT e ISO27001; por lo tanto “*Es perfectamente correcto mezclar y combinar controles de COBIT y de ISO 17799 y casi cualquier otro estándar de seguridad de la información; rara vez se encuentran en desacuerdo en los detalles*” (van der Stock, 2005).

## Objetivos de la Investigación

* + 1. *Objetivos Generales*

Evaluar la efectividad de la aplicación de técnicas de OWASP para la protección contra ataques de inyección de SQL

* + 1. *Objetivos Específicos*

Los objetivos particulares de éste estudio son:

* Ejecutar ataques de Inyección de SQL a objetivos específicos.
* Obtener las métricas de riesgo tales como el número total de vulnerabilidades identificadas así como su clasificación de riesgo con base a OWASP.
* Mostrar el nivel de madurez, en el desarrollo de software seguro de la organización objetivo, con base a OWASP.
* Implementar técnicas de OWASP, a las aplicaciones web objetivo y a la organización objetivo.
* Una vez que las técnicas OWASP han sido implementados, ejecutar las mismas técnicas y script de ataques a los mismos objetivos.
* Obtener las métricas de riesgo, y el nivel de madurez; para evaluar si el número de vulnerabilidades y su respectivo nivel de riesgo, así como el nivel de madurez de la organización, varió con base al ejercicio inicial.
* Presentar los resultados en conferencias y revistas especializadas.

## METODOLOGÍA

El tipo de investigación será Descriptiva (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 1991); ya que trataremos primero de describir e identificar cómo se realizan los ataques de inyección de SQL; posteriormente describir los controles OWASP que se pretenden implementar. Por último, y con base a las variables de investigación, medir la efectividad de los controles OWASP para mitigar los ataques de inyección de SQL.

Por otro lado, el diseño del experimento será cuasi experimental (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 1991); lo anterior debido a que no nos será posible tener una aplicación web con la misma arquitectura que otra, y tal vez las tecnologías utilizadas para el desarrollo de las aplicaciones, así como la infraestructura en la que se encuentran, será diferente. Del mismo modo, ninguna organización es igual a otra, aun estando dentro de una misma industria.

El diseño del experimento será de preprueba-postprueba y grupos intactos (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 1991). Éste diseño es el que más se apega a la validación de las hipótesis planteadas; ya que nos permitirá hacer las pruebas de penetración a las aplicaciones objetivo, y a evaluar el nivel de madurez de la organización, antes de la aplicación de técnicas OWASP; obtener resultados, ejecutar de nuevo las pruebas de penetración, evaluar de nuevo el nivel de madurez de la organización, y de ésta manera comparar los resultados obtenidos antes y después de la implementación de técnicas OWASP.

Desde el punto de vista técnico, estaremos realizando las evaluaciones de las vulnerabilidades de inyección de SQL con base al nivel cuatro de evaluación de (OWASP, 2009), cuyo alcance es todo el código que fue modificado o desarrollado para crear la aplicación incluyendo librerías y servicios invocados; en términos de la aplicación web, esto implica evaluar toda la aplicación.

Los requerimientos de verificación los estaremos acotando al nivel V5 (Input Validation Verification Requirements) ya que es el que se apega a la verificación de inyección de SQL; los puntos de verificación V5 correspondientes a un nivel de evaluación cuatro se detallan en la siguiente tabla (OWASP, 2009):

Tabla 1 OWASP ASVS Requerimientos de Validación de Captura (V5) (OWASP, 2009)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Verification Requirement** | **Level 1A** | **Level 1B** | **Level 2A** | **Level 2B** | **Level 3** | **Level 4** |
| 1. Verify that the runtime environment is not susceptible to buffer overflows, or that security controls prevent buffer overflows. | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 1. Verify that a positive validation pattern is defined and applied to all input. | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 1. Verify that all input validation failures result in input rejection or input sanitization. | ✓ |  | ✓ | ✓ | ✓ | ✓ |
| 1. Verify that a character set, such as UTF-8, is specified for all sources of input. |  |  | ✓ | ✓ | ✓ | ✓ |
| 1. Verify that all input validation is performed on the server side. |  |  | ✓ | ✓ | ✓ | ✓ |
| 1. Verify that a single input validation control is used by the application for each type of data that is accepted. |  |  |  | ✓ | ✓ | ✓ |
| 1. Verify that all input validation failures are logged. |  |  |  | ✓ | ✓ | ✓ |
| 1. Verify that all input data is canonicalized for all downstream decoders or interpreters prior to validation. |  |  |  |  | ✓ | ✓ |
| 1. Verify that all input validation controls are not affected by any malicious code. |  |  |  |  |  | ✓ |

Utilizaremos diversas herramientas para automatizar la identificación y explotación de la vulnerabilidad de inyección de SQL; dado el costo de las soluciones comerciales, nos enfocaremos a utilizar soluciones de software libre como SQLiX, SQLmap, SQL Ninja, ZAP.

Por otro lado, evaluaremos las prácticas de desarrollo de software de la organización con base al cuestionario de (Chandra & Deleersnyder, 2012), definiremos el nivel de madurez descrito por SAMM y entregaremos las recomendaciones a la organización.

Se espera que la organización trabaje en esos puntos, para después de un tiempo acordado con la organización objetivo, regresar y aplicar el mismo cuestionario, las mismas pruebas a la aplicación, verificar evidencias de la implementación de las recomendaciones, y revaluar el nivel de madurez de (Chandra & Deleersnyder, 2012).

# RESULTADOS

# DISCUSIÓN

# CONCLUSIÓN

# RECONOCIMIENTOS

# REFERENCIAS

Chandra, P., & Deleersnyder, S. (2012). *OWASP Software Assurance Maturity Model.* Creative Commons (CC) Attribution-Share Alike.

Clarke, J. (2012). *SQL Injection Attacks and Defense, 2nd Edition.* Syngress.

Diario Oficial de la Federación. (2010). DECRETO por el que se expide la Ley Federal de Protección de Datos Personales en Posesión de los Particulares. *Diario Oficial de la Federación*, 11-12.

Harris, S. (2008). *All in One CISSP Exam Guide.* McGraw-Hill.

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (1991). *Metodología de la investigación.* Naucalpan de Juárez, Edo. de México: McGRAW - HILL INTERAMERICANA DE MÉXICO, S.A. de C.V.

Hewlett-Packard. (2012). *HP 2012 Cyber Risk Report.*

OWASP. (2009). *OWASP Application Security Verification Standard (ASVS).* Creative Commons (CC) Attribution Share-Alike.

Ponemon Institute. (2012). *2012 Cost of Cyber Crime Study:.* Ponemon Institute LLC.

Puppy, R. F. (24 de Diciembre de 1998). *Phrack.* Recuperado el Abril de 2013, de NT Web Technology Vulnerabilities: http://www.phrack.com/issues.html?issue=54&id=8#article

Scholt, T., Balzarotti, D., & Kirda, E. (2012). *Have things changed now? An Empirical Study on Input Validation Vulnerabilities in Web Applications.*

Tatum, M. (Enero de 2013). *What Is a Cyberattack?* Recuperado el Marzo de 2013, de Wise Geek: http://www.wisegeek.com/what-is-a-cyberattack.htm

The Open Source Vulnerability Database. (2012). *The Open Source Vulnerability Database*. Recuperado el Marzo 2013 de 2013, de The Open Source Vulnerability Database: http://www.osvdb.org

van der Stock, A. (2005). *Una Guía para Construir Aplicaciones y Servicios Web Seguros.* Free Software Foundation.

Williams, J., & Wichers, D. (2013). *OWASP Top Ten - 2013.* Creative Commons (CC) Attribution Share-Alike.

# BIOGRAFÍA DEL AUTOR

Manuel López es un entusiasta de la Seguridad de la Información con 10 años de experiencia trabajando en diversas instituciones transnacionales. Es co-líder y co-fundador del capítulo OWASP Guadalajara. Comentarios pueden ser enviados a (manuel punto lopez en owasp punto org).