DISEÑO DE METODOLOGIA PARA VERIFICAR LA SEGURIDAD EN APLICACIONES WEB CONTRA INYECCIONES SQL.

Ivan Camilo Gómez
Código 1400214

UNIVERSIDAD MILITAR NUEVA GRANADA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA EN TELECOMUNICACIONES
BOGOTÁ, D. C.
NOVIEMBRE 2011

DISEÑO DE METODOLOGIA PARA VERIFICAR LA SEGURIDAD EN APLICACIONES WEB CONTRA INYECCIONES SQL.

Ivan Camilo Gómez González

Trabajo final de monografía presentado como requisito para optar por el título de Ingeniero en Telecomunicaciones

Director
Ing. Edward Guillen

UNIVERSIDAD MILITAR NUEVA GRANADA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA EN TELECOMUNICACIONES
BOGOTÁ, D. C.
MAYO 2012

NOTA DE ACEPTACIÓN

_____

_____

_____

_____

Presidente del jurado

_____

Jurado

_____

Jurado

BOGOTÁ, D. C., MAYO 2012

DEDICATORIA

A *mis padres y su enorme paciencia;*

A *mi Esposa que se ha convertido en mi ancla y mi vida*

Ivan Camilo

AGRADECIMIENTOS

A dios que me ha dado la fortaleza para sobrepasar todos los obstáculos que se han presentado en mi vida y el cual me va a llevar a grandes caminos.

A mis padres por su paciencia en este proceso de crecimiento profesional y por su apoyo incondicional en todas las situaciones de mi vida.

A mi esposa Erika por la comprensión y por el apoyo que siempre me presto en el desarrollo de este proceso.

A la Universidad Militar Nueva Granada la cual me todos sus recursos tecnológicos, organizacionales, humanos y logísticos para poder realizar este trabajo, especialmente a mi director del proyecto el ingeniero Edward Paul Guillen y a mi asesora Yaneth Cárdenas por sus constantes consejos y recomendaciones.

TABLA DE CONTENIDO

1. INTRODUCCIÓN

Actualmente el creciente uso masivo de la tecnología y de Internet ha facilitado al mundo, el uso de los diferentes servicios como consulta de páginas estáticas, consulta de páginas dinámicas, reservas en línea, compra de productos, asesorías online, entre otros.

Puesto que algunos de estos servicios manejan información confidencial, es de vital importancia manejar dicha información con sumo cuidado, para evitar la pérdida, la manipulación, o que accedan a estos activos.

Debido a que los diferentes servicios mencionados anteriormente, generalmente están automatizados mediante herramientas como las bases de datos, es de vital importancia asegurar estas herramientas para que en lo posible las intrusiones y manipulación sean nulas.

Este trabajo consiste en realizar una metodología que nos permita revisar y aumentar la seguridad en las aplicaciones que específicamente manejen bases de datos multidimensionales.

1.1 TITULO

“DISEÑO DE METODOLOGIA PARA VERIFICAR LA SEGURIDAD EN APLICACIONES WEB CONTRA INYECCIONES SQL. ”

1.2 PLANTEAMIENTO DEL PROBLEMA

En algunas aplicaciones en Internet, las cuales están conectadas con bases de datos se encuentran inconsistencias en la información que está allí almacenada, como por ejemplo: la información no se encuentra completa, la

información ha sido modificada, la información ha sido borrada totalmente, estas inconsistencias son debido a modificaciones realizadas por personas o programas internos o ajenos a la organización, que no tienen los privilegios para realizar este tipo de alteraciones o modificaciones

Actualmente según Owasp, en su artículo “El Top diez de los riesgos de las aplicaciones”, nos dice que la principal vulnerabilidad en las aplicaciones web son las inyecciones. Se denominan así puesto que el agresor envía o inyecta cadenas de texto las cuales manipulan la sintaxis de los diferentes gestores de bases de datos existentes, como por ejemplo, PostgreSQL, Firebird,MySQL, Fox Pro, IBM Informix, Paradox, Microsoft SQL, entre otros. Este tipo de fallas generan consecuencias como: consultas no autorizadas de información, modificación de la información, borrado de la información o que el atacante tome posesión total de la administración de la base de datos.

Por lo anterior se ve la necesidad de crear un diseño metodológico para verificar la seguridad en aplicaciones web contra inyecciones SQL, que nos permita actuar de una manera dinámica en la protección de los recursos informáticos.

1.3 OBJETIVOS

Las metas trazadas para llevar a cabo el proyecto son definidas como:

1.3.1 OBJETIVO GENERAL

Diseñar una metodología para el análisis de vulnerabilidades de acuerdo a parámetros de comparación utilizados en ataques denominados inyecciones por código SQL en aplicaciones web que manejen bases de datos multidimensionales.

1.3.2 OBJETIVOS ESPECÍFICOS

- Realizar una documentación de metodologías ya existentes para la verificación de vulnerabilidades en aplicaciones web contra inyecciones SQL.
- Realizar una descripción de tres sistemas gestores de bases de datos para utilizarlos como parámetros de comparación en la metodología propuesta.

- Realizar una descripción de tres lenguajes de programación en los cuales están escritas las aplicaciones web para tomarlo como otro parámetro de comparación.
- Comparar tres tipos diferentes de metodologías existentes para la verificación de las vulnerabilidades web, específicamente en los ataques denominados inyección SQL.
- Diseñar de acuerdo a los parámetros establecidos la metodología para la verificación de vulnerabilidades en aplicaciones web contra los ataques denominados inyección SQL.
- Realizar la documentación pertinente de la metodología y de los resultados obtenidos.

1.3.3 ANTECEDENTES

- En el año 2002 en Estados Unidos, Chirs Anley presento un artículo titulado ¨Inyección Avanzada de SQL en Servidores con Aplicaciones SQL¨, en este documento se mira en detalle la técnica de inyección sql aplicada a la plataforma Microsoft Internet Information Server/Active Server Pages/SQL server.

- En el año 2005 OWASP, Open Web Appication Security Project, presenta en su pagina de Internet, www.owasp.org, un articulo titulado ¨ **UNA GUIA PARA CONSTRUIR APLICACIONES Y SERVICIOS WEB SEGUROS.**¨ en la cual se muestra como evitar las vulnerabilidades de las aplicaciones para las inyecciones SQL.

- En el año 2008, Carlos Tori libera un articulo titulado ¨**Hacking Ético**¨ en el cual en uno de sus capítulos habla sobre la seguridad en aplicaciones Web y de las diferentes técnicas de inyección SQL conocidas hasta ese momento

- En el año 2008 en Madrid, España presento una tesis de grado titulada ¨ **PERTRECHAMIENTO DE ATAQUES DE INYECCIÓN SQL CIEGA Y EXTRACCIÓN DE CONTENIDO EN SISTEMAS SQL SERVER MEDIANTE EXFILTRACIÓN DE DATOS VÍA DNS Y ENCAMINAMIENTO SOBRE LA ZONA DESMILITARIZADA.¨,** este trabajo nos habla de la de lo peligroso y de que manera se puede ejecutar una inyección SQL ciega.

- En el año 2010 OWASP, Open Web Application Security Project, presenta en su pagina de Internet, www.owasp.org, un articulo titulado ¨ **GUIDE TO SQL INJECTION¨,** en este articulo se encuentra un detallado análisis de cuales son los diferentes tipos de vulnerabilidades en las diferentes tipos de inyección SQL.

2. MARCO TEÓRICO

Para Ubicarse en el área de conocimiento del trabajo se hace el siguiente marco el cual esta dividió en marco teórico conceptual y marco teórico referencial.

2.1 MARCO TEÓRICO CONCEPTUAL"

En esta sección se desarrollarán los conceptos básicos relacionados con las aplicaciones Web e inyección SQL.

2.1.1 APLICACIONES WEB

Las aplicaciones según la Real Academia de la Lengua Española son: "4. f. Inform. Programa preparado para una utilización específica, como el pago de nóminas, formación de un banco de términos léxicos, etc." y Web es: "1. f. Inform. Red informática.", en Ingeniería de software se denomina Aplicación Web al software que el usuario utiliza, accediendo a un servidor Web por medio de navegadores.

Estas aplicaciones generalmente son sistemas complejos que involucran varias capas que estas compuestas por múltiples componentes como, servidores Web, servidores de aplicaciones, bases de datos entre otros, todos estos elementos trabajando en conjunto permiten el correcto funcionamiento de la aplicación.

Aunque las variaciones que se pueden encontrar son multiples generalmente una aplicación web está estructurada en tres capas: La primera capa es el navegador, como por ejemplo: Mozilla Firefox, Google Chrome Internet Explorer, entre otros, la segunda capa es un motor que utiliza una tecnología de web dinámica, como por ejemplo: PHP, ASP.NET, Phyton, ROR, entre otros y por último se encuentra la tercera capa que es una base de datos.

Algunas de las ventajas de las aplicaciones Web son: Ahorra tiempo, es decir se pueden realizar tareas básicas sin necesidad de instalar software

adicional, no hay problemas de compatibilidad es decir no hay diferencia si se trabaja con un computador Linux o Windows o Mac, la portabilidad nos permite ingresar desde un portátil o un computador de escritorio.

2.1.2 METODOLOGÍA DE SEGURIDAD

Una metodología de seguridad consiste en la ejecución de determinados pasos a seguir, con el fin de determinar la mayor cantidad de amenazas que puedan afectar a una organización, evaluar cuales pueden ser las vulnerabilidades con su respectivo nivel de riesgo y sus posibles efectos en las diferentes áreas de la organización.

Actualmente existen una gran cantidad de estándares, normas y procedimientos públicos y privados que permitan generar una defensa fuerte contra ataques informáticos, una detección temprana de una intrusión y medidas de contingencia de rápido desarrollo, permiten que la organización sea lo menos vulnerable posible ante cualquier ataque.

2.1.3 SQL

Sql es un lenguaje declarativo de acceso a las bases de datos el cual permite ejecutar operaciones de algebra y calculo relacional sobre estas, permitiendo realizar consultas y modificaciones sobre la información, el sql esta normalizado, es decir nos permite trabajar en cualquier tipo de lenguaje como por ejemplo asp o php y adicionalmente se puede combinar con cualquier tipo de base de datos Ms Access, Sql Server, Mysql entre otras.

Aunque este lenguaje este normalizado, no quiere decir que sea totalmente igual para todos los tipos de lenguajes y de bases de datos, es por esto que algunas funciones no sirven en todos los lenguajes, sin embargo la potencia y la versatilidad que tiene es enorme.

Las bases de datos están compuestas por tablas, las cuales contienen una gran variedad de diferentes tipos de información, por esto se generalizan los tipos de datos que maneja de la siguiente manera.

Alfanuméricos	Contienen cifras y letras. Longitud limitada (255 caracteres)
Numéricos	Varios tipos principalmente, enteros (sin decimales) y reales (con decimales).
Booleanos	Verdadero y falso (Sí o No)
Fechas	Almacenan fechas permitiendo ordenar los registros por fechas o calcular los días entre una fecha y otra.
Memos	Son campos alfanuméricos de longitud ilimitada.
Autoincrementables	Campos numéricos enteros que incrementan en uno su valor para cada registro incorporado. Sirven como identificadores de un registro

Tabla 1. Tipos de datos Sql

Las sentencias que se manejan en sql son diversas, a continuación un listado de las mas utilizadas.

SELECT	Recupera datos de la bd
INSERT	Añade datos a la bd
DELETE	Borra datos de la bd
UPDATE	Modifica datos de la bd
CREATE TABLE	Adiciona una tabla a la bd
DROP TABLE	Suprime una table de la bd
ALTER TABLE	Modifica la estructura de la tabla
CREATE INDEX	Crea indice para las colomnas
DROP INDEX	Suprime los indices de las colomnas

CREATE SYNONYM	Crea alias para la table
DROP SYNONYM	Suprime los alias de las tablas.
GRANT	Da privilegios a un usuario
REVOKE	Quita los privilegios de un usuario
COMMIT	Finaliza la transacción actual
ROLLBACK	Aborta la transacción actual.

Tabla 2. Sentencias Sql

Con este tipo de sentencias se manipulan los datos y se obtiene la información que se necesita de la base de datos.

2.2 MARCO TEÓRICO REFERENCIAL

El marco teórico referencial se enfoca hacia los conceptos necesarios para el desarrollo del trabajo.

2.2.1 SISTEMAS GESTORES DE BASES DE DATOS

Los sistemas gestores de bases de datos conocidos como SGBD o DBMS del ingles Database Management system son programas muy específicos que permiten la conexión entre las bases de datos y las aplicaciones que las utilizan, estos programas tienen como objetivo manejar de manera clara y sencilla los datos que posteriormente serán requeridos por una persona, grupo u organización.

Los sistemas gestores de bases de datos buscan tener ciertas características que permitan una gran potencia y flexibilida, estas caracteristicas son:

- Abstracción de la información: Evitar al usuario detalles de almacenamiento que y de manejo de los datos, puesto que no es relevante para los mismo.
- Consistencia: Permite que los datos sean congruentes con la realidad y que si existe redundancia, los datos repetidos sean actualizados de forma simultanea.

- Independencia: Hace referencia a la capacidad de modificar el esquema lógico o físico in necesidad de realizar cambios en las aplicaciones que las utilizan.
- Seguridad: Hace referencia a que se debe tener la capacidad de otorgar diversas categorías para que los diferentes tipos de usuarios accedan a la información sin comprometer la confidencialidad, disponibilidad e integridad de los datos
- Tiempo de respuesta: Se desea minimizar los tiempos en que los datos son almacenados y la información presentada al usuario.
- Control de redundancia: Se desea evitar la duplicidad de datos para que nos se presenten inconsistencia ni perdida de espacio en disco.

Existen diferentes sistemas gestores de bases de datos, los cuales se pueden clasificar por su tipo de licenciamiento, algunos de estos son

Sistemas gestores de Bases de Datos Libres:

PostgrsSQL

Firebird

SQLite

DB2 Express-c

Apache Derby

Maria DB

MySQL

Drizzle

Sistemas Gestores de Bases de Datos No libres

Advantage Database

dBase

FileMaker

FoxPro

gsBase

IBM DB2

IBM Informix

Interbase

Microsoft Access
Microsoft SQL Server
Oracle
Paradox

Sistemas Gestores de Bases de Datos No libres y Gratuitos

Microsoft SQL Server Compact Edition Basic
Sybase ASE
Oracle Express edition 10
DB2 Express-C

- **POSTGRES SQL**

El Postgres Sql es un sistema gestor de base de datos relacional orientada a objetos Y Libre, el desarrollo de Postgres no está dirigido por alguien específico o por una empresa, puesto que es código abierto está dirigido por el PGDG (Postgres Sql Global Development Group).

La evolución de este Sistema gestor empieza en 1982 con un proyecto que inicialmente se llamó 'Ingres' que fue el primer intento de implementar un motor de base de datos relacional, en 1985 después de la primera experiencia comercial de este producto se empezó a trabajar en el POST Ingres o Postgres el cual pretendía resolver los problemas encontrados en la primera versión comercial, para 1988 salio la primera versión utilizable. Al pasar de los años se ha venido evolucionando y depurando el código resolviendo problemas y bugs que estaban presentes. Actualmente el Proyecto Postgres SQL continua haciendo lanzamientos con la contribución de los empresas aportantes, proveedores y programadores de código abierto que desean mejorar las prestaciones de este Sistema.

Las características más relevantes son las siguientes:

Implementación del estándar SQL92/SQL99.
Licencia BSD.
Por su arquitectura se escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
Soporte para triggers y procedimientos en el servidor.
Incorpora estructura de datos array.
Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.

Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
Se pueden realizar varias operaciones al mismo tiempo sobre la misma tabla sin necesidad de bloquearla. Postgres es un gestor que fácilmente puede competir con muchos gestores comerciales, permite un manejo de usuarios y de base de datos con una gestión fácil, la velocidad de respuesta es se mantiene de bases de datos pequeñas a bases de datos grandes, lo cual habla muy bien de su desempeño

- **MYSQL**

Este es un Sistema Gestor de Bases de Datos relacional, multihilo y multiusuario, el cual tiene un esquema de licenciamiento dual, es decir, se ofrece bajo licencia GNU GPL para cualquier uso compatible con este, pero para empresas que quieran incorporarlo en productos privados se debe adquirir una licencia específica, puesto que su desarrollo es patrocinado por una empresa privada que posee Copyright de la mayor

parte del código. Adicional a las licencias, la empresa ofrece soporte y servicios adicionales. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

Las características principales son:

Excelente velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
Consume muy pocos recursos ya sea de CPU y de memoria.
Licencia GPL y también posee una licencia comercial para aquellas empresas que deseen incluirlo en sus aplicaciones privadas.
Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
Soporta hasta 64 índices por tabla

Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
Tiene soporte para transacciones permite agrupar transacciones.

Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.

Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.

Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.

Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro (Barrapunto.com) y de buscadores de aplicaciones (Freshmeat.net).

- **ORACLE**

Es un sistema gestor de base de datos Relacional orientado a objetos desarrollado por Oracle Corporation. Las principales características son:

Oracle es uno de los motores de base de datos relacional más usado a nivel mundial.
Permite el manejo de triggers y procedimientos almacenados, posee la característica de integridad
Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
El software del servidor puede ejecutarse en multitud de sistemas operativos.
Existe incluso una versión personal para Windows 9x, lo cual es un punto a favor para los desarrolladores que se llevan trabajo a casa.
¿Qué hay de los objetos de Oracle? Este sistema ha comenzado a evolucionar en esta dirección, añadiendo tipos de clases, referencias, tablas anidadas, matrices y otras estructuras de datos complejas.
Desafortunadamente, la implementación actual de las mismas no ofrece una ventaja clara en eficiencia, como sería de esperar, y sí provocan la incompatibilidad de los diseños que aprovechan las nuevas características con otras bases de datos.
Oracle es la base de datos con mas orientación hacía INTERNET
Un aceptable soporte

El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son muy caras.

2.2..2 LENGUAJES DE PROGRAMACIÓN

- **PHP**

Es un lenguaje de programación para ambiente web, esta orientado principalmente para crear aplicaciones para internet o que se visualizan por el navegador, php es uno de los lenguajes mas utilizados para el acceso a las bases de datos como mysql, Oracle y postgre Sql, su licencia es GPL y es OPEN SOURCE.

Como se ejecuta PHP

Inicialmente un cliente web realiza una petición al servidor la cual llama a un archivo con extensión PHP, inmediatamente esta petición es interpretada por PHP y se devuelve al servidor para que se envié la respuesta en formato HTML. Por eso el código fuente de php no es visible.

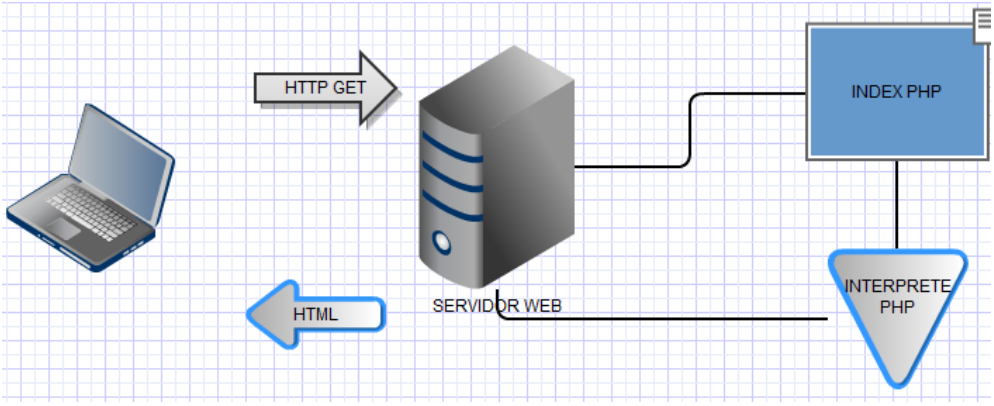


Figura 1. Ejecución PHP

Adicionalmente en PHP se tienen ciertas características como:

No se tiene un tipo de datos definido, es decir no se tiene que inicializar las variables como por ejemplo integer o double entre otros, no hay declaración de variables, todas empiezan con el signo $. Adicionalmente este lenguaje puede ser embebido en el lenguaje HTML.

Se caracteriza por ser un lenguaje muy rápido.
Soporta en cierta medida la orientación a objeto. Clases y herencia.
Es un lenguaje multiplataforma: Linux, Windows, entre otros.
Capacidad de conexión con la mayoría de los manejadores de base de datos: MysSQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
Capacidad de expandir su potencial utilizando módulos.
Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

- **ASP**

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitio web dinámicos. ASP significa en inglés (Active Server Pages), fue liberado por Microsoft en 1996. Para poder visualizar las páginas web desarrolladas bajo este lenguaje es necesario tener instalado Internet Information Server (IIS).
ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es VBScript, nativo de Microsoft. ASP se puede hacer también en Perl and Jscript (no JavaScript). El código ASP puede ser insertado junto con el código HTML. Los archivos cuentan con la extensión (asp).

Este es un lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia de mercado denominada .NET.

Algunas características de ASP son:

Código desorganizado.
Se necesita escribir mucho código para realizar funciones sencillas.
Tecnología propietaria.

Por otro lado el ASP.NET fue desarrollado para resolver las limitantes que brindaba tu antecesor ASP. Creado para desarrollar web sencillas o grandes aplicaciones. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. Los archivos cuentan con la extensión (aspx). Para su funcionamiento de las páginas se necesita tener instalado IIS con el Framework .Net. Microsft Windows 2003 incluye este framework, solo se necesitará instalarlo en versiones anteriores.
Algunas de las características de asp.net son:
Completamente orientado a objetos.
Controles de usuario y personalizados.
División entre la capa de aplicación o diseño y el código.
Facilita el mantenimiento de grandes aplicaciones.
Incremento de velocidad de respuesta del servidor.

- **JSP**

Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor.

JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

Algunas de las características de este lenguaje son las siguientes:
Código separado de la lógica del programa.
Las páginas son compiladas en la primera petición.
Permite separar la parte dinámica de la estática en las páginas web.
Los archivos se encuentran con la extensión (jsp).
El código JSP puede ser incrustado en código HTML.
Ejecución rápida del servlets.
Crear páginas del lado del servidor.
Multiplataforma.
Código bien estructurado.
Integridad con los módulos de Java.
La parte dinámica está escrita en Java.
Permite la utilización se servlets.

2.2.3 METODOLOGÍA OSSTMM

OSSTM, Open Source Security Methodology es una metodología libre que desarrollo el ISECOM, Institute for Security and Open Methodologies, la cual consiste en un paso a paso para llevar a cabo mediciones y pruebas de seguridad. Se encuentra dividida en seis secciones, que entre todas permiten realizar un completo análisis del sistema.

Estas son:

- Seguridad de la Información
- Seguridad de los Procesos
- Seguridad en las tecnologías de Internet
- Seguridad en las Comunicaciones

- Seguridad Inalámbrica
- Seguridad Física

En cada sección se especifican una serie de módulos a ser evaluados, teniendo en cuenta si aplica o no, cada uno de ellos al tema en cuestión, el resultado de la observación de todos ellos es lo que nos permitirá tener el panorama de seguridad.

El ISECOM en su sección de Seguridad en las tecnologías de internet especifica una serie de análisis y de pruebas que nos permiten analizar detalladamente nuestro sistema, adicionalmente si el ítem en cuestión no existe se debe colocar en un documento formal que dicho ítem no es aplicable.

- Logística y Controles
- Sondeo de Red
- Identificación de los Servicios de Sistemas
- Búsqueda de Información Competitiva
- Revisión de Privacidad
- Obtención de Documentos
- Búsqueda y Verificación de Vulnerabilidades
- Testeo de Aplicaciones de Internet
- Enrutamiento
- Testeo de Sistemas Confiados
- Testeo de Control de Acceso
- Testeo de Sistema de Detección de Intrusos
- Testeo de Medidas de Contingencia
- Descifrado de Contraseña
- Testeo de Denegación de Servicios
- Evaluación de Políticas de Seguridad

Uno de los puntos de la sección se Seguridad de las tecnologías de internet es el Testeo de Aplicaciones, aquí en este punto se maneja un gráfico muy claro de cuáles son los diferentes tipos de test que se deben ejecutar.

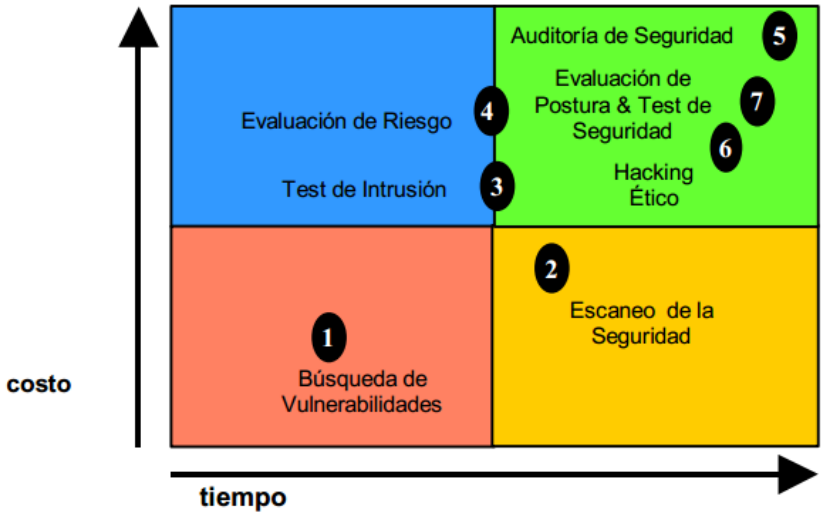


Figura 2. Test de Aplicaciones

1. Búsqueda de Vulnerabilidades: Hace referencia generalmente a las comprobaciones automáticas de un sistema o sistemas dentro de una red.

2. Escaneo de la Seguridad: Hace referencia a las búsquedas de vulnerabilidades que incluyen verificaciones manuales de falsos positivos, identificación de los puntos débiles de la red y análisis profesional individualizado.

3. Test de Intrusión: Hace referencia a los proyectos orientados a objetivos en los cuales dicho objetivo es obtener un trofeo, que incluye ganar acceso privilegiado con medios pre-condicionales.

4. Evaluación de Riesgo: Hace referencia a los análisis de seguridad a través de entrevistas e investigación de nivel medio que incluye la justificación negocios, las justificaciones legales y las justificaciones específicas de la industria.

5. Auditoría de Seguridad: hace referencia a la inspección manual con privilegios administrativos del sistema operativo y de los programas de aplicación del sistema o sistemas dentro de una red o redes.

6. Hacking Ético: Hace referencia a los tests de intrusión en los cuales el objetivo es obtener escalamiento de permisos en la red dentro del tiempo predeterminado de duración del proyecto.

7. Test de Seguridad y Evaluación de Postura, es una evaluación de riesgo

con orientación de proyecto de los sistemas y redes, a través de la aplicación de análisis profesional mediante escaneos de seguridad donde la intrusión se usa generalmente para confirmar los falsos positivos y los falsos negativos dentro del tiempo permitido de duración del proyecto.

2.2.4 ITIL

Itil consiste en una librería de las mejores prácticas para servicios de tecnologías de información (TI) de alta calidad. Lo que principalmente busca es resumir un extenso conjunto de normas y procedimientos de gestión ideados con el fin de que las organizaciones puedan tener calidad y eficiencia en los diferentes procesos de TI.

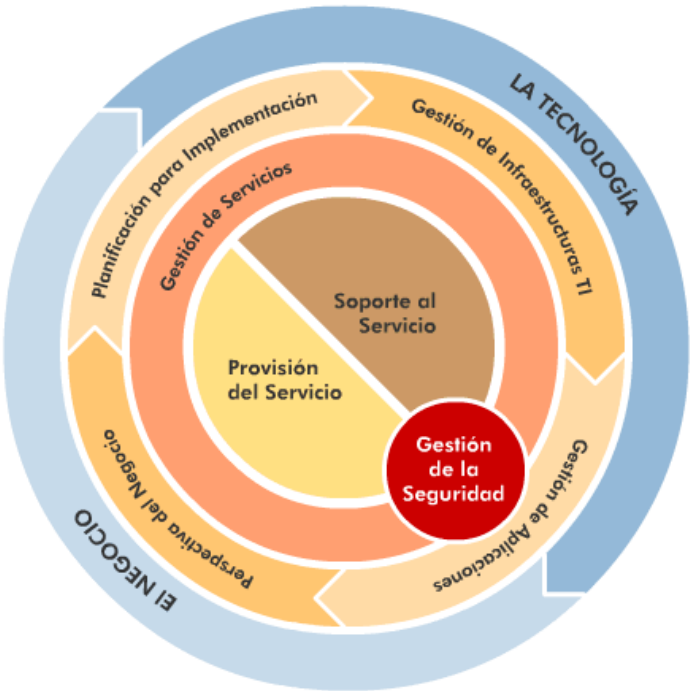
ITIL provee un acercamiento sistemático y profesional a la administración de las TI. Mediante la adopción de las prácticas resumidas por ITIL, se puede obtener en la organización:

- Disminución de costos
- Mejora en la productividad

- Mejora de las habilidades y experiencia del personal de la organización
- Mejora en la prestación de servicios a terceros
- Ejecución de procesos e implementaciones según estándares internacionales.

Estos beneficios se pueden tener gracias a un sencillo, pero efectivo modelo de trabajo propuesto por varias fuentes, que consta de un ciclo de cuatro fases, conocido por su sigla en inglés como el PDCA (Plan, Do, Check, Act), que para su traducción se podría tomar como el ciclo de la Planificación, Implementación, Monitorización y Revisión de los diferentes procesos que componen el proceso de Proveer y dar Soporte a los Servicios, como lo ilustrados por la Figura 3.

Figura 3. El modelo ITIL[1]



[1] Imagen e en ITIL. http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/que_es_ITIL/que_es_ITIL.php

2.2.5 OWASP

Owasp es una comunidad libre dedicada a permitir que las organizaciones desarrollen, conciban, operen y mantengan las diferentes aplicaciones de manera segura alrededor del mundo.

El 1 de diciembre de 2001 Owasp salió a la luz pública por medio de su página web, con algunos principios característicos que son: Sin ánimo de lucro, libre y abierta, no seguirá ningún interés de uso comercial y se compromete a manejar su código de ética, el cual puede ser encontrado en la página web oficial.

Owasp tiene una serie de publicaciones que abarcan muchos temas de seguridad informática. Entre ellas se encuentra varios artículos referentes al tema en cuestión.

Uno de ellos es `SQL INJECTION`, en este artículo se tiene una descripción de que es un ataque de inyección Sql y algunos posibles problemas que se presentan con este tipo de ataques.

3. INGENIERÍA DEL PROYECTO

Para el desarrollo de este trabajo, se han tenido en cuenta una serie de metodologías ya existentes, lenguajes y sistemas gestores de bases de datos, basado en ellos se genera una metodología que busca identificar de gran manera los peligros que sufren las aplicaciones por causa específicamente de las inyecciones sql.

3.1 DESARROLLO DEL PROYECTO

Para el desarrollo del trabajo se documentaron previamente diferentes Sistemas Gestores de Bases de Datos, se documentaron diferentes lenguajes de programación y se realizó una comparación de los mismos.

3.1.1 COMPARACIÓN DE TRES DIFERENTES SISTEMAS GESTORES DE BASES DE DATOS

Se determinaron diferentes puntos en los sistemas gestores de bases de datos que van a ser comparados y analizados en la siguiente tabla.

Descripción	Mysql	Oracle	Postgresql
Arquitectura	Arquitectura cliente servidor de 2 capas. Capa de motor de almacenamiento y capa superior	Motor de almacenamiento unificado con 3 componentes	Cliente Servidor
Desempeño	Es relativo al ambiente en donde se desarrolla		
Multiprocesador	Se pensaba que Oracle permitía un mejor manejo de multiprocesadores, pero en las últimas versiones Mysql ha avanzado mucho en el tema		
Procedimientos almacenados	si	si	si
Disparadores	si	si	si
Replicación	si	si	si

Sub consultas	si no en vistas	si	si
Indexado	si	si	si

Tabla 3. Comparación SGBDS

3.1.2 COMPARACIÓN DE DIFERENTES LENGUAJES DE PROGRAMACIÓN.

En la siguiente tabla tenemos una comparación general de algunas de las características de los lenguajes de programación explicados anteriormente.

DESCRIPCION	JSP	PHP	ASP	ASP.NET
Fácil aprendizaje	NO	SI	SI	SI
Multiplataforma	SI	SI	NO	NO
Conexión con bases de datos	SI	SI	SI	SI
Velocidad de Acceso	DEPENDE DE LA EXPERIENCIA DEL PROGRAMADOR			
Pago	no	no	si/no	si/no
Soporte de clases	si	si	si	si

Tabla 4. Comparación Lenguajes

La evolución de estos diferentes lenguajes buscando mejorar el desempeño de los mismos. Hace que este cuadro comparativo este en constante evolución. En el análisis de los ítems mencionados se tienen algunos puntos para resaltar.

Fácil aprendizaje: La arquitectura de jsp es un poco más complicada que la php, asp y asp.net, pero con el continuo crecimiento de la implementación de este tipo de lenguaje en el día a día se ha aumentado la cantidad de información y documentación que se tiene del mismo, esto facilita el aprendizaje.

Multiplataforma: Asp y Asp.net son nativos de plataformas Microsoft. PHP desde la versión 5 se puede unir con ambientes IIS (microsoft).

Velocidad de acceso: El desempeño de estos lenguajes es parecido, no existe uno que sobresalga de manera notoria con relación a su desempeño, pero este puede ser afectado por la falta de pericia al programar y de manejo erróneo del código.

3.1.3 DESARROLLO DE LA METODOLOGÍA.

Esta metodología planteada se basa en el ciclo de mejora continua, el cual será explicado a continuación.

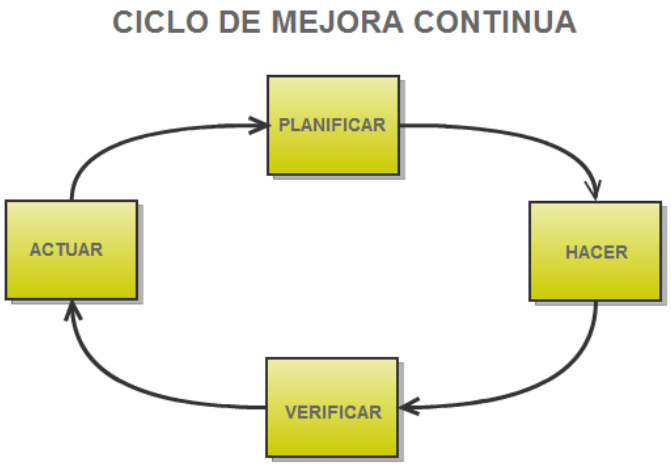


Figura 4. Ciclo mejora continua

El primer paso a seguir es la **PLANIFICACIÓN**

Esta primera etapa esta dividida en varios pasos a seguir.

- Identificación: este paso hace referencia a que se debe analizar y saber cuales de las diferentes aplicaciones que puede tener una organización quieren ser protegidas.
- Recopilación de datos: esta paso hace referencia a que se debe tener una documentación de todas las partes que integran la aplicación, como por ejemplo lenguaje en la que esta hecha, modulos que las

componen, tipo de servidores en donde esta montada la aplicación, posibles fallos que se pueden encontrar.

- Definición de los parámetros a mejorar: este paso hace referencia a que se debe definir y tener muy claro después del identificación y de la recopilación de datos cuales son loa parámetros que deseamos mejorar.
- Definir los niveles de mejora: se deben identificar los diferentes niveles de mejorar que se van a manejar para nuestro propósito.

El segundo paso es el **HACER**

Para esta etapa se deben realizar los siguientes pasos:

- **Tests vulnerabilidades:** Para realizar este test de vulnerabilidades se deben utilizar herramientas que realicen un proceso automático de verificación, como por ejemplo SQL MAP BSQL HACKER, MSSQL HAXOR, PIPPER, entre otras

  Búsqueda de vulnerabilidades: Este paso hace referencia a realizar de manera manual la confirmación de las diferentes vulnerabilidades encontradas en el punto anterior y las verificaciones que se crean pertinentes.

  Test de intrusión: Este test consiste en realizar un intento de adquirir permisos, datos a modificaciones sobre nuestra base de datos

  Validación de datos de Entrada:

  Se debe asegurar que todas las entradas que pueden ser modificadas están propiamente validadas.

  Se debe asegurar que se realicen los respectivos chequeo de longitud en todas las entradas existentes. Por ejemplo si una página solicita un código de identificación de usuario y este código

no es mayor a 7 dígitos, se debe asegurar que la entrada por parte del usuario no exceda este límite.

Se debe validar que los datos ingresados no contengan ningún tipo de carácter especial que pueda modificar nuestro código base.

Se recomienda que la validación de datos ocurra en el lado del servidor.

SE debe examinar la ausencia de puertas traseras por las cuales los atacante puedan filtrarse:

Evaluación de riesgo: Hace referencia a clasificar cada uno de los riesgos que se obtuvieron después de realizar los pasos anteriores.

Implementación de cambios: Después de realizar la evaluación de riesgos se deben realizar los cambios pertinentes y que se ajusten a nuestro análisis inicial de niveles de mejora planteados.

El tercer paso es **VERIFICAR:**

En este paso se debe recopilar información sobre las diferentes vulnerabilidades encontradas en el punto anterior y se debe comparar contra los objetivos trazados en la fase inicial, a su vez se deben documentar las mejoras y sus resultados prácticos.

Se debe verificar que mejoras hacen falta por implementar.

El cuarto paso a seguir es **ACTUAR**:

Se debe realizar un análisis con base en la documentación anterior, este análisis nos debe llevar a tres pasos.

- Se debe determinar si las modificaciones que se hicieron son satisfactorias a las metas trazada y no se debe realizar el ciclo nuevamente
- Se debe determinar si las modificaciones que se hicieron no son lo suficientemente satisfactorias a las metas trazadas y se debe realizar nuevamente el ciclo de mejora continua
- Se debe plantear la pregunta de cómo se podría mejorar la seguridad en nuestra aplicación y documentar todo el proceso realizado.

4. CONCLUSIONES

Después de la realización de este proyecto se obtuvieron las siguientes conclusiones.

- Se debe concientizar a las organizaciones de las implicaciones y de los alcances que tienen los ataques de inyección SQL. Cuando se conoce y se evalúan los daños que pueden hacer, se decide actuar para combatir y protegerse ante este tipo de intrusiones.

- Las aplicaciones no se deben diseñar solo para cumplir un objetivo sino que a la vez se deben diseñar de tal manera que no se comprometa la seguridad de la información en la organización.

- Si una aplicación está en fase de ejecución no se debe dejar de lado las medidas para evaluar y clasificar los riesgos presentes, para así protegerla de las inyecciones SQL.

- Esta metodología es aplicable para diferentes lenguajes de programación y sistemas gestores de bases de datos.