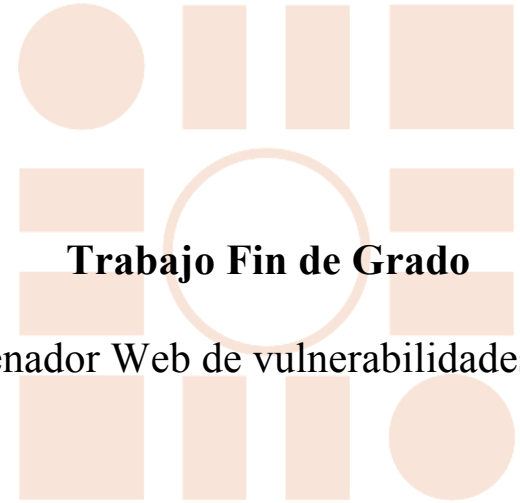
Universidad de Alcalá
Escuela Politécnica Superior

GRADO EN SISTEMAS DE INFORMACIÓN



Trabajo Fin de Grado

Entrenador Web de vulnerabilidades SQL

Autor: Rocío Recuero Santaella

Tutor/es: D. Manuel Sánchez Rubio

2015

Entrenador Web de Vulnerabilidades SQL

Trabajo Fin de Grado

Rocío Recuero Santaella
rrecuerosantaella@gmail.com

Tutor: Manuel Sánchez Rubio

Resumen

El principal objetivo de este proyecto es el desarrollo de una aplicación que permita evidenciar la existencia de vulnerabilidades de seguridad intrínsecas a los sistemas Web, centrado específicamente en aquellas de tipo *Inyección SQL*.

La aplicación desarrollada contiene vulnerabilidades diseñadas específicamente para un propósito docente, es decir, los usuarios dispondrán de distintas funcionalidades donde poder verificar diferentes niveles y tipos de vulnerabilidad.

Para responder a las diferentes necesidades del aprendizaje, en la aplicación se incluye un "modo seguro" donde los usuarios se enfrenten al desafío de una aplicación diseñada teniendo en cuenta ciertos mínimos aspectos de *ciberseguridad*.

Abstract

This Project main purpose is develop an application able to highlight all those security vulnerabilities intrinsic to Web Systems, focused mainly on those concerning to SQL Injection.

Some designing specific vulnerabilities are part of the developed application responding a docent objective, this means, users will be allowed to check different weakness levels and types in each kind of functionalities.

In order to respond to several training needs, application has included a "safety mode" which confronts users to the challenge of an application that has been designed taking certain minimal aspects of *cyber security* in account.

Palabras clave

Web, SQL Injection, Vulnerabilidades, Ciberseguridad.

2015

Resumen extendido

En el presente Trabajo Fin de Grado (TFG), se recogen todos los aspectos necesarios para el desarrollo de una aplicación Web cuyas funcionalidades sean la gestión básica de usuarios y noticias, permitiendo además monitorizar ataques de seguridad, con diferentes niveles de vulnerabilidad, para medir posteriormente la eficacia de los ataques, según su naturaleza, y origen.

La aplicación a diseñar, debe ser una aplicación web, es decir, un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador o explorador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (*HTTP*) están estandarizados y no han de ser creados por el programador de aplicaciones.

Desde el punto de vista del usuario, este tipo de aplicaciones consisten en un enorme conjunto de documentos llamados páginas, cada una de las cuales pueden contener vínculos o enlaces con otras páginas relacionadas a modo de gran repositorio de información. En un servicio Web los clientes demandan hipertextos a los servidores por medio de direcciones bajo el esquema URL (*Universal Resource Locator*) el cual permite localizar recursos en la red, incluyendo en la red de Internet.

Más concretamente, la aplicación debe ser una Web intuitiva y fácil de usar para cualquier usuario, pero cuyo objetivo principal es garantizar la existencia de vulnerabilidades fácilmente detectables de tipo *SQL Injection,* de tal manera que permitiese a los usuarios entrenar este tipo de ataques en un entorno seguro y controlado.

La aplicación se desarrolla en lenguaje Java, puesto que éste permite el desarrollo de aplicaciones en red, distribuidas y concurrentes independientes de la plataforma, ya que el código fuente del programa se compila una única vez generando el *bytecode* asociado y éste es interpretado en tiempo de ejecución, necesitando únicamente para su correcto funcionamiento una máquina virtual Java.

Por esta característica de ser independiente de la plataforma, puede ser instalada y ejecutada en cualquier servidor de aplicaciones que sea y soporte un contenedor de Servlets y JSP, como son *Apache-Tomcat, WebSphere, Weblogic, GlassFish, etc.*, aunque en la presente memoria se describe la instalación utilizando Apache-Tomcat.

Una vez instalada en el servidor, una aplicación Web será accesible a través de cualquier navegador de la red. Se ha comprobado su correcto funcionamiento para los navegadores de internet más utilizados actualmente: *Mozilla, Internet Explorer, Google Chrome y Safari*.

La aplicación se compone de dos partes principales: la base de datos y la aplicación propiamente dicha.

La Base de Datos se ha implementado con el gestor de base de datos relacionales *MySQL*. Se ha optado por esta tecnología, ya que es una herramienta potente con licenciamiento *GNU*, ampliamente utilizada en entornos profesionales y de la administración pública, y además, porque posee la credencial de tener millones de descargas en internet.

Lo realmente destacable de la aplicación es que ha sido desarrollada, utilizando el patrón MVC. Mediante el cual se consiguen aislar procesos de obtención de los datos a mostrar al usuario, de la forma en la que estos son presentados. El hecho de separar los distintos códigos (*Java Servlets, JSP, JavaScript, CSS*…) permite un desarrollo y diseño más cómodo y organizado, además de posibilitar un mantenimiento y desarrollo posterior de la aplicación mucho más efectivo y sencillo.

El uso del framework *Struts* provee la infraestructura básica para la implementación del patrón MVC, proporciona la integración con el modelo, la lógica de negocio se implementa basándose en clases predefinidas por el framework y la construcción de la interfaz está soportada por la utilización de un conjunto de *tags* predefinidos, buscando evitar el uso de *scriptlets* (código Java incluido en el JSP dentro de etiquetas <% %>).

Comúnmente se dice que Java es un lenguaje de programación seguro, fue diseñado para solucionar ciertas vulnerabilidades relacionadas con el acceso a memoria presentes en otros lenguajes como *C++*.

Hay ciertas características específicas del lenguaje que contribuyen de forma extraordinaria a este objetivo. No obstante, cualquier sistema de información conlleva exposición a un riesgo.

Ciertamente, cualquier elemento que muestre información puede conllevar un riesgo. Antes de la aparición de las TIC toda la información se guardaba en papel en almacenes y grandes sistemas de archivadores. Los sistemas informáticos permiten la digitalización de todo este volumen de información, ganando además de espacio físico, velocidad en el acceso a la información. Pero con ello aparecen toda una serie de implicaciones derivadas del uso de estas facilidades, ya que el sistema de información debe seguir garantizando la seguridad de la información.

Se entiende por seguridad de la información al conjunto de métodos o medidas técnicas, organizativas y legales cuyo objetivo es garantizar la confidencialidad, la integridad y la disponibilidad del sistema de información tratando otras propiedades como la responsabilidad, la autenticidad y la fiabilidad.

Toda organización debe tener en cuenta un Sistema de Gestión de la Seguridad de la información, el cual se encarga de proteger y cuidar la información frente a las amenazas a las que cualquier sistema está expuesto.

Se puede definir como amenaza a todo elemento o acción capaz de atentar contra la seguridad de la información, provocando un daño (material o inmaterial).

Las amenazas surgen a partir de la existencia de vulnerabilidades, es decir que una amenaza sólo puede existir si existe una vulnerabilidad que pueda ser aprovechada, e independientemente de que se comprometa o no la seguridad de un sistema de información.

La presencia de una amenaza es una advertencia de que puede ser inminente el daño a algún activo de la información, o bien es un indicador de que el daño se está produciendo o ya se ha producido.

La mayor parte de los ataques a los sistemas informáticos son provocados, intencionadamente o no, por las personas que en general persiguen conseguir un nivel de privilegio en el sistema que les permita realizar acciones no autorizadas sobre el mismo.

En un sistema informático hay que proteger todos los recursos que forman parte del sistema, esto es:

- Hardware, donde se incluyen todos los elementos físicos del sistema informático.
- Software, donde se encuentran todos los programas que se ejecutan sobre el hardware.
- Datos, aquellos que comprenden la información que procesa el software haciendo uso del hardware.
- Otros (consumibles, personas, infraestructuras, etc.).

De entre todos los activos, el más crítico son los datos, ya que la dificultad o imposibilidad de reponerlos conllevaría una pérdida de tiempo y dinero.

Sin llegar a entrar en detalle sobre todos los tipos de personas que pueden constituir una amenaza para el sistema, se pueden mencionar los curiosos, intrusos remunerados, crackers, terroristas, ex-empleados e incluso el personal de la propia organización.

De forma general las amenazas pueden clasificarse en 2 grandes grupos: amenazas físicas y amenazas lógicas, pudiendo ser materializadas por personas, programas específicos o catástrofes naturales (inundación, incendio, fallo eléctrico, explosión, etc.).

El objetivo de esta aplicación web es la creación de un entorno seguro y apropiado para la realización de pruebas para la verificación de la existencia de vulnerabilidades, en concreto el estudio se ha centrado en aquellas de *SQL Injection*, cuyo origen radica en el incorrecto chequeo y/o filtrado de las variables utilizadas en un programa que contiene, o genera, código SQL.

Las *inyecciones SQL* se han convertido en un problema muy común en sitios web que cuentan con base de datos. La vulnerabilidad es fácilmente detectada y fácilmente explotada, y como tal, cualquier sitio es propenso a ser objeto de un intento de ataque de este tipo.

Este tipo de intrusión normalmente es de carácter malicioso, dañino o espía y por tanto son un problema de seguridad informática, y para poder prevenirlo y evitarlo debe ser tomado en cuenta por el programador de la aplicación.

La intrusión ocurre durante la ejecución del programa vulnerable, exponiendo cualquier dato de la base de datos para ser leído o modificado por un usuario malintencionado. Los ataques por *SQL Injection* permiten a los atacantes suplantar identidad, alterar datos existentes, causar problemas de repudio como anular transacciones o cambiar balances, permiten la revelación de todos los datos del sistema, destruirlos o si no volverlos inasequibles, incluso convertirse en administradores del servidor de base de datos.

Las vulnerabilidades son difíciles de gestionar. Se descubren decenas día a día, y clasificarlas es una tarea compleja.

La mejor forma de prevenir es sin duda el conocimiento, comprender cómo y porque es posible este tipo de debilidad en nuestras aplicaciones. Todo dato enviado a nuestra aplicación por un usuario, ya sea este humano o electrónico, es susceptible de contener código SQL que podría modificar el comportamiento esperado de nuestra aplicación. Por lo tanto, cualquier información que nuestra aplicación esté esperando desde fuera, debe tomarse como potencialmente peligrosa.

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR

Universidad
de Alcalá