FACULTAD DE TECNOLOGIAS DE INFORMACION Y COMUNICACION

ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

Tarea 2

Tipos de ataques, Inyección SQL y XSS

BSI – 090-2 PROGRAMACION III

Profesor

Jorge Vásquez

Alumnos

Carlos Gonzalez Segura

San José, 25 Mayo 2018

INJECTION

Un ataque por inyección SQL consiste en la inserción o "inyección" de una consulta SQL por medio de los datos de entrada desde el cliente hacia la aplicación. Un ataque por inyección SQL exitoso puede leer información sensible desde la base de datos, modificar la información (Insert/ Update/ Delete), ejecutar operaciones de administración sobre la base de datos (tal como parar la base de datos), recuperar el contenido de un determinado archivo presente sobre el sistema de archivos del DBMS y en algunos casos emitir comandos al sistema operativo. Los ataques por inyección SQL son un tipo de ataque de inyección, en el cual los comandos SQL son insertados en la entrada de datos con la finalidad de efectuar la ejecución de comandos SQL predefinidos.

Los ataques por inyección SQL permiten a los atacantes suplantar identidad, alterar datos existentes, causar problemas de repudio como anular transacciones o cambiar balances, permite la revelación de todos los datos en el sistema, destruir los datos o si no volverlos inasequibles, y convertirse en administradores del servidor de base de datos.

La inyección SQL es muy común con aplicaciones PHP y ASP debido a la prevalencia de interfaces funcionales obsoletas. Debido a la naturaleza de las interfaces programáticas disponibles, las aplicaciones J2EE y ASP.NET tienen menor probabilidad de ser fácilmente atacadas por una inyección SQL.

La gravedad de una inyección SQL está limitada por la habilidad e imaginación del atacante, y en menor medida a las contramedidas, como por ejemplo las conexiones con bajo privilegio al servidor de bases de datos, entre otras. En general, se considera a la inyección SQL de alto impacto.

XSS

Los ataques por secuencias de comandos entre páginas web (o XSS) son ataques dirigidos a las páginas web con ciertas fallas al momento de programarlas, es decir, estas fallas ocurre al momento de no verificar o 'filtrar' la información que ingresa el usuario común, ya sea mediante un formulario o en una URL o de algún otra forma. Este tipo de ataque obliga a la página web mostrar el texto ingresado tal cual se envió, por lo tanto al momento de mostrarlo lo hará como sólo de una forma y es en código HTML.
El código "inyectado" en una página web vulnerable se considera malintencionado o código malicioso.

El código malicioso utilizado en este tipo de ataque está compuesto por cadena de datos: scripts completos contenidos en enlaces o ejecutados desde formularios vulnerables.

La forma de aprovechar esta vulnerabilidad es sabiendo de Javascript, ya que es el lenguaje de programación web útil, es decir, también se puede ingresar código html, pero no serviría

de mucho, por las limitaciones que tiene, por no ser un lenguaje de programación sino de maquetado.

Javascript es el lenguaje que se encarga de hacer más dinámica la página, aunque ahora con HTML5 su uso se disminuyo en ciertas cosas, pero aun se sigue usando mucho por su utilidad al hacer dinámicas las paginas web.

Clasificación de los ataques Cross Site Scripting

1.- XSS persistente o directo

Este tipo de ataque consiste en inyectar (o embeber) codigo HTML en sitio que lo permita por medio de etiquetas <script> o <iframe>. Es el tipo más grave ya que el código se queda implantado en la web de manera interna y se ejecuta una y otra vez al abrir la página web hasta no ser "reseteada" la página o eliminado ese código en concreto del sitio.
Local. Es una de las variantes del XSS directo, uno de sus objetivos consiste en explotar las vulnerabilidades del mismo código fuente o página web. Esas vulnerabilidades son resultado del uso indebido del DOM (Modelo de Objetos del Documento, es un conjunto estandarizado de objetos para representar páginas web) con JavaScript, lo cual permite abrir otra página web con código malicioso JavaScript incrustado, afectando el código de la primera página en el sistema local.

2.- XSS reflejado o indirecto

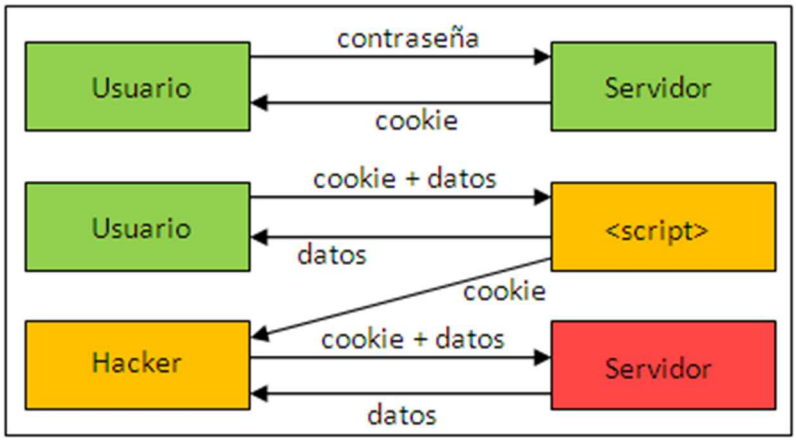
Es el tipo de ataque XSS más habitual y consiste solo en editar los valores que pasan mediante URL, ingresando el código malicioso haciendo que se ejecute en dicho sitio. En otras palabras, sucede cuando se envía un mensaje o ruta en una URL, una cookie o en la cabecera HTTP.

Métodos de inyección de código utilizado en los ataques.

A la hora de lanzar un ataque de este tipo, los atacantes pueden utilizar varios tipos de inyección de código distinto. Veamos a continuación cuáles son los más utilizados.
1.- Inyección en un formulario
Se trata del ataque más sencillo. Consiste en inyectar código en un formulario que después al enviarlo al servidor, será incluido en el código fuente de alguna página. Una vez insertado en el código fuente, cada vez que se cargue la página se ejecutará el código insertado en ella.

2.- Inyección por medio de elementos
En este tipo de sistema de inyección de código se utiliza cualquier elemento que viaje entre el navegador y la aplicación, como pueden ser los atributos usados en las etiquetas HTML utilizadas en el diseño de la página.

3.- Inyección por medio de recursos
Aparte de los elementos en la url y los formularios, hay otras formas en la que se puede actuar como son las cabeceras HTTP. Estas cabeceras son mensajes con los que se comunican el navegador y el servidor.
Aquí entran en juego las cookies, las sesiones, campo referer…



Operación de un ataque XSS

Maneras de evitar los ataques

Por suerte, prevenir estos ataques es sencillo, y una vez que aprendamos cómo hacerlo es algo que siempre deberíamos tener presente para evitarlos.
A continuación veremos algunos consejos prácticos que podemos llevar a cabo para proteger nuestras aplicaciones web mediante la validación de entrada de datos, asegurándonos que cada uno de los campos que utilicemos no contengan código que pueda afectar nuestra aplicación o a nuestros usuarios.

1.- Limitar los caracteres de entrada
Lo primero que debemos hacer es limitar los caracteres que un usuario puede introducir en los campos de texto. Por ejemplo, si tenemos un campo para introducir el nombre del usuario, no vamos a dejarlo abierto para que se puedan introducir un número indefinido de caracteres, sino que lo vamos a limitar por ejemplo a 20 o 30 caracteres. Para limitar el número de caracteres, podemos utilizar la variable “maxlength” que nos proporciona el estándar HTML.
2.- Sanear los datosCuando hablamos de sanear los datos, nos estamos refiriendo a quedarnos únicamente con la información que nos interesa, eliminando las etiquetas HTML que pueden ser incluidas en una caja de texto. Por ejemplo, si estamos almacenando el

nombre de una persona, de poco nos sirve que el usuario lo introduzca en negrita, ya que lo único que nos interesa es su nombre.
Para lograr esta limpieza, podemos utilizar la función “strip_tags”. Por ejemplo:
$comentario = strip_tags($_POST[‘comentario’]);

3.- Escapar los datos

Para proteger los datos y mostrarlos tal y como el usuario los introdujo, deberíamos “escapar” los datos al presentarlos al usuario. Es decir, caracteres que deben ser representados por entidades HTML si se desea preservar su significado (por ejemplo las comillas dobles hay que transformarlas en " que es como se representa en HTML). Con esto evitamos que el navegador lo ejecute y evalúe el código.
Para lograr esto, podemos utilizar la función “htmlspecialchars”. Por ejemplo:
echo “Mostrando resultados de: ”.htmlspecialchars($_GET['busqueda’]);
Pero aparte de tomar medidas de seguridad para impedir ataques XSS en nuestros sitios, debemos evitar almacenar información relevante de nuestros visitantes en las Cookies, y además encriptar las sesiones de usuario para que en caso de que las capturen no puedan acceder a la información almacenada en ella.

Bibliografía

- Creative Commons Attribution-ShareAlike (2012). **Welcome to OWASP. Descargado de:** https://www.owasp.org/index.php/Inyecci%C3%B3n_SQL.
- Copyright © 2018 Un estudiante de Sistemas. (2010). Un estudiante de Sistemas. Descargado de: http://insecuritytime.blogspot.com/2015/10/tipos-de-ataque-xss-que-es-y-como.html