SQL INYECTION

Mauricio Mijail De La QuintanaIllanes
Universidad Mayor de San Andrés
Facultad de Ciencias Puras y Naturales
Carrera de Informática
Análisis y Diseño de Sistemas de Información
maudelaquintana@gmail.com

RESUMEN
En este artículo se muestra que la inyecciones SQL son un ataque que consiste en la introducción de una consulta oculta SQL oculta en una petición del cliente a una página web. Nos muestra como clasificar estos ataques según la vulnerabilidad que aprovechan y finalmente como mitigarlos en base a listas blancas y negras.

Palabras Claves

Test de Penetracion, Ethical Hacking, Redes, Java, Paginas Web, Lenguaje Sql

1. INTRODUCCIÓN

De acuerdo con el proyecto Open Web Application Security (OWASP), los ataques de inyección están en la lista de las 10 principales vulnerabilidades web. Siendo que las inyecciones SQL responsables de una gran parte de esto. El uso de estos ataques SQL es todavía trivial ya que esta vulnerabilidad no es sólo aplicable a la web sino también puede ocurrir en aplicaciones de escritorio que usan software apoyado en un servidor. La capacidad de detección de estas vulnerabilidades depende de la complejidad de la aplicación en cuestión. La mayoría de las veces, las herramientas como antivirus o firewalls no pueden alertarnos sobre la brecha de seguridad que tenemos.

1.1 ¿QUÉ ES SQL?

El Lenguaje de Consulta Estructurado (SQL) se utiliza para consultar, operar y administrar los sistemas de bases de datos como Microsoft SQL Server, Oracle o MySQL. El uso general de SQL es consistente a través de todos los sistemas de bases de datos que lo apoyan, sin embargo, hay complejidades que son propias de cada sistema.

Los sistemas de bases de datos se utilizan comúnmente para proporcionar una funcionalidad de servidor para muchos tipos de aplicaciones web. En apoyo de aplicaciones web, los datos suministrados por el usuario se utilizan a menudo para crear dinámicamente las sentencias SQL que interactúan directamente con una base de datos.

1.2 ¿QUÉ ES UNA INYECCIÓN SQL?

Un ataque de inyección de SQL es un ataque que va dirigido a subvertir la intención original de la solicitud mediante la presentación suministrada por el atacante sentencias SQL directamente en la base de datos. Dependiendo de la aplicación web, y cómo procesa los datos suministrados por el atacante antes de la construcción de una sentencia SQL, un exitoso ataque de inyección SQL puede tener implicaciones de largo alcance, que van de derivación de autenticación para la divulgación de información para facilitar la distribución de código malicioso a los usuarios de la aplicación.

2. TIPOS DE INYECCION SQL

Hay una serie de tipos de inyección SQL clasificadas en base a las vulnerabilidades que aprovechan, estas son:

- Cadenas mal filtradas
- Incorrecta manipulación de tipo
- Evasion de Firma
- Blind SQL Injection

2.1 CADENAS MAL FITRADAS

Inyecciones de SQL se basan en cadenas mal filtrados son causados por la entrada del usuario que no se filtra para caracteres de escape. Esto significa que un usuario puede introducir una variable que se puede pasar como en una instrucción SQL, lo que resulta en la manipulación de entrada de base de datos por el usuario final.

2.2 INCORRECTA VERIFICACION DE TIPO

Una incorrecta verificación de tipo ocurre cuando una entrada no está marcada por limitaciones de tipo. Un ejemplo de esto sería un campo de ID que es numérico, pero que no hay filtrado en su lugar para comprobar que la entrada del usuario es numérico. is_numeric () siempre se debe utilizar cuando el tipo de campo que se supone explícitamente a ser un número.

2.3 EVASION DE FIRMA

Muchas de las inyecciones SQL serán bloquedaos por los sistemas de detección y prevención de intrusiones de intrusos utilizando las reglas de detección de firmas. Programas comunes que detectan las inyecciones SQL son mod_security para Apache y Snort. Estos programas no son a toda prueba y por lo tanto, las firmas pueden ser evadidas.

2.4 BLIND SQL INJECION

Estas inyecciones se conocen como inyecciones SQL a ciegas. Entre ellas se clasifican entre Inyecciones parcialmente ciegas y totalmente ciegas. Las Inyecciones parcialmente ciegas se pueden ver pequeños cambios en la página de resultados, por ejemplo, una inyección deficiente puede redirigir el atacante a la página principal, donde si tiene éxito devolverá una página en blanco.

Las inyecciones totalmente ciegas a diferencia de las inyecciones parcialmente ciegas no producen cambios en la página web. Esto sigue siendo sin embargo inyectable, aunque es difícil determinar si la inyección es en realidad lleva a cabo.

2.5 IMPACTOS DE UNA INJECTION SQL

Aunque los efectos de un ataque de inyección SQL éxito varían en función de la aplicación específica y la forma en que la aplicación procesa los datos suministrados por el usuario, inyección SQL en general se puede utilizar para llevar a cabo los siguientes tipos de ataque:

- Divulgación de información: Este ataque permite a un atacante obtener, ya sea directa o indirectamente, la información sensible en una base de datos.
- Comprometer la integridad de los datos: Este ataque consiste en la alteración de los contenidos de una base de datos. Un atacante podría utilizar este ataque para desfigurar una página web, o más probablemente para insertar contenido malicioso en páginas web.
- Disponibilidad de datos: Este ataque permite a un atacante borrar información con la intención de causar daño o eliminar la información del registro de auditoría o en una base de datos.
- Ejecución de comandos remotos: Uso de comandos a través de una base de datos puede permitir a un atacante poner en peligro el sistema operativo host. Estos ataques a menudo aprovechan un procedimiento existente, del sistema operativo para la ejecución del comando host.

3 DEFENSA CONTRA ATAQUES INJECTION SQL

Un ataque de inyección SQL se puede detectar y potencialmente bloqueada en dos localidades en un flujo de tráfico de aplicaciones: en la demanda y en la red.

3.1 LAS DEFENSAS DE APLICACIÓN

Hay varias formas en las que una aplicación puede defenderse contra los ataques de inyección SQL. Los enfoques primarios incluyen la validación de los datos suministrados por el usuario, en forma de lista blanca o lista negra, y la construcción de sentencias SQL de forma que los datos proporcionados por el usuario no pueden influir en la lógica de la declaración.

3.2 LISTAS BLANCAS Y NEGRAS

Dentro de una aplicación en sí, hay dos enfoques para la validación de entradas que puede defenderse contra los ataques de inyección SQL: listas negras y listas blancas. Con las listas negras, personajes maliciosos específicos, conocidos se eliminan o sustituyen en la entrada del usuario. Aunque este enfoque se implementa a menudo, en gran parte debido a la facilidad con la que se puede lograr, no es eficaz cuando se compara con listas blancas. Lista negra puede dejar de manejar adecuadamente la ofuscación complejo, que podría permitir a un atacante para subvertir filtros y potencialmente inyectar sentencias SQL. Un ejemplo del enfoque de listas negras es la parametrizacion de sentencias sql. A continuación se muestra un ejemplo en java

```
Connection con = (acquire Connection)
Statement st= con.createStatement();
ResultSet rset = st.executeQuery("SELECT * FROM
usuarios WHERE nombre = '" + nombreUsuario + "';");
```

Usando st.setString(1, nombreUsuario) se establece que la sentencia podrá ser solo un string igual al nombre de usuario cualquier otra cosa es descartada.

```
Connection con = (acquire Connection)
PreparedStatement st = con.prepareStatement("SELECT *
FROM usuarios WHERE nombre = ?");
st.setString(1, nombreUsuario);
ResultSet rset = st.executeQuery();
```

Alternativamente, listas blancas examina cada pedazo de la entrada del usuario con una lista de caracteres permitidos. Este enfoque es más eficaz para mitigar el riesgo de inyección de SQL, ya que es más restrictiva sobre el cual se permiten los tipos de entrada. Listas blancas bien implementado debe examinar cada pieza de datos suministrados por el usuario contra el formato de datos esperado. A continuación un ejemplo en el lenguaje java.

En este código se reemplazan las cadenas por otras que son invalidas para asi que cuando un intruso introduzca sentencias estas sean convertidas en caracteres ambiguos que serán rechazados

```
Connection con = (acquire Connection)
Statement st= con.createStatement();
ResultSet rset = st.executeQuery("SELECT * FROM
usuarios WHERE nombre = '" +
nombreUsuario.replace("\\", "\\\\").replace("'", "\\'")
+ "';");
```

Independientemente del enfoque, lo más probable es que sea necesario adaptar los caracteres permitidos por el tipo de campo de entrada o por la clase de campo de entrada (por ejemplo, texto o datos numéricos). La validación de la entrada y las funciones de desinfección que se utilizan para filtrar inyección SQL permite a las sentencias que puedan ser generalizados y se utiliza para filtrar los caracteres que pueden usar para una injection sql.

4. CONCLUSION

En la actualidad las páginas web están expuestas a las inyecciones sql que pueden ocasionar la filtración de información de la base de datos o la transmisión de contenido malicioso. Para esto un desarrollador web puede programar cientos controles a nivel de código en el lenguaje java que puede ser parametrizar las sentencias sql o remplazar las sentencias invalidas por otras validas, a fin de evitar cualquier intrusión.

5. REFERENCIAS

OWASP: SQL Injection
http://www.owasp.org/index.php/SQL_injection

How To: Protect From SQL Injection in ASP.NET
http://msdn.microsoft.com/en-us/library/ms998271.aspx

Ronaldo Gumerato *SQL Injection En Aplicaciones Web*

Justin Clarke *SQL Injection Attacks and Defense, Second Edition*