MyBatis框架中常见的SQL注入

先知 / 2018-05-19 10:26:00 / 浏览数 2660 安全技术 漏洞分析 顶(0) 踩(0)

作者：c0d3p1ut0s | 授权发布

0x00 MyBatis概述&背景

MyBatis
是支持定制化SQL、存储过程以及高级映射的优秀的持久层框架。由于它非常灵活，非常轻量级，受到广大开发者的欢迎，各个大厂也用得比较多。MyBatis框架介绍相关的

写到一半发现有些概念要在前面说清楚一下，不然容易晕。

- MySQL：指MySQL服务器。
- MyBatis：指MyBatis框架。
- JDBC：是Java用来规范数据库连接的接口。
- MySQL Connector/J：MySQL提供的、符合JDBC的、用来供java程序连接MySQL数据库的jar包。俗称：MySQL数据库驱动。

0x01 MyBatis的SQL注入

MyBatis支持两种参数符号，一种是#，另一种是$。
使用参数符号#的句子：

```
<select id="selectPerson" parameterType="int" resultType="hashmap">
 SELECT * FROM PERSON WHERE ID = #{id}
</select>
```

MyBatis会创建一个预编译语句，生成的代码类似于

```
// Similar JDBC code, NOT MyBatis…
String selectPerson = "SELECT * FROM PERSON WHERE ID=?";
PreparedStatement ps = conn.prepareStatement(selectPerson);
ps.setInt(1,id);
```

参数会在SQL语句中用占位符”?”来标识，然后使用prepareStatement来预编译这个SQL语句。

但是你以为这个SQL语句真的被MySQL数据库预编译了吗？naive！其实在默认情况下，MySQL
Connector/J只不过是把selectPerson做了一下转义，前后加了双引号，拼接到SQL语句里面，然后再交给MySQL执行罢了，更多的细节可以看这里
https://c0d3p1ut0s.github.io/%E7%AE%80%E5%8D%95%E8%AF%B4%E8%AF%B4MySQL-Prepared-Statement/

另一种使用参数符号$时，MyBatis直接用字符串拼接把参数和SQL语句拼接在一起，然后执行。众所周知，这种情况非常危险，极容易产生SQL注入漏洞。

在使用MyBatis框架时，有以下场景极易产生SQL注入。

SQL语句中的一些部分，例如order
by字段、表名等，是无法使用预编译语句的。这种场景极易产生SQL注入。推荐开发在Java层面做映射，设置一个字段/表名数组，仅允许用户传入索引值。这样保证传入

like参数注入。使用如下SQL语句可防止SQL注入

like concat('%',#{title}, '%')

in之后参数的SQL注入。使用如下SQL语句可防止SQL注入

```
id in
<foreach collection="ids" item="item" open="("separator="," close=")">
#{item}
</foreach>
```

0x02 x-generator的SQL注入

为了提高开发效率，一些generator工具被开发出来，generator是一个从数据库结构
自动生成实体类、Mapper接口以及对应的XML文件的工具。常见的generator有mybatis-generator，renren-generator等。

mybatis-generator是mybatis官方的一款generator。在mybatis-generator自动生成的SQL语句中，order
by使用的是$，也就是简单的字符串拼接，这种情况下极易产生SQL注入。需要开发者特别注意。

不过，mybatis-generator产生的like语句和in语句全部都是用的参数符号#，都是非常安全的实现。

上一篇：利用Java反射和类加载机制绕过J... 下一篇：深入分析Google YOLO点击...
1. 0 条回复
   - 动动手指，沙发就是你的了！