接口sql漏洞注入测试

# 1：什么是sql漏洞注入攻击？

Sql漏洞注入是一种将sql代码添加到参数中，传递到服务端解析并执行的一种手段。

开发在接受入参的时候，未对参数进行过滤处理，直接将参数拼接到Sql语句当中，执行并达到一种预想之外的行为，称之为Sql漏洞攻击。

## 2：sql漏洞是如何产生的？

1：开发人员未对接受到的入参进行过滤校验。

2：攻击人员通过接口发送给服务器的输入参数构造可执行的Sql代码，（可把sql代码加入post请求，get请求，http信息头，cookie中）

3：服务端数据库未做相应的安全配置。

## 3：如何预防sql注入？

作为QA人员，了解sql漏洞注入是一项必备的技能，可以从源头上堵住Sql的漏洞，提升产品的安全性。

1：严格检查输入变量的类型和格式。

对于整型参数加判断条件不能为空，参数类型必须为数字

对于字符串参数可以使用正则表达式过滤

2：过滤和转义字符

对字符串变量比如接受到的username参数进行变量前的转义，例如对’、”、\等特殊字符进行转义

3：利用mysql的预编译机制

把sql语句的模板（变量采用占位符进行占位）发送给mysql服务器，mysql服务器对sql语句的模板进行编译，编译之后根据语句的优化分析对相应的索引进行优化，在最终绑定参数时把相应的参数传送给mysql服务器，直接进行执行，节省了sql查询时间，以及mysql服务器的资源，达到一次编译、多次执行的目的，除此之外，还可以防止SQL注入。

## 4:如何通过接口进行Sql漏洞测试？

接口入参的时候，有数值型参数及字符串参数，文件类型（file）等几种数据类型入参，而接口有get型和写入型两种接口类型，无论是get型还是写入型接口，我们在调用接口的时候都是对数据库的查询，写入操作。在调用接口的时候参数是不受任何限制的可以通过参数伪造Sql代码，传递到服务端查看是否做了参数变量前的校验是否与预期结果一致。

**1):接口sql漏洞测试用例设计之数字注入？**

例如：玩吧有一个 查询用户身份信息的接口，接口地址为 /v3/user/userinfo入参为uid,和username,当我们调用这个接口的时候，相当于调用一个查询语句。

开发的代码是这样写的，当我们调接口请求的时候，开发get到我们请求的参数，直接把参数变量直接拼接在sql语句中了，如下所示：

string sql= " select \* from appUser where uid=' " +uid+ " ' and username = ' " +username+ " ' ";

例如：查询小波的身份信息（uid为68047730,username为yangjianbo）

正常的请求数据为

data={“url”:”/v3/user/userinfo”,”params”:’{“uid”:”68047730”,”username”:”yangjianbo”}’}

这时服务端接受到的参数，sql拼接语句为：

string sql= " select \* from appUser where uid=' 68047730 ' and username = 'yangjianbo";

这样就可以查询出关于小波的玩吧用户信息了

但是呢？接口是可以随意入参的，我们想怎么入参就怎么入参，只要你高兴就可以^\_^

那我们就把username这个参数入一个' or '1' = '1这个参数试试看~~

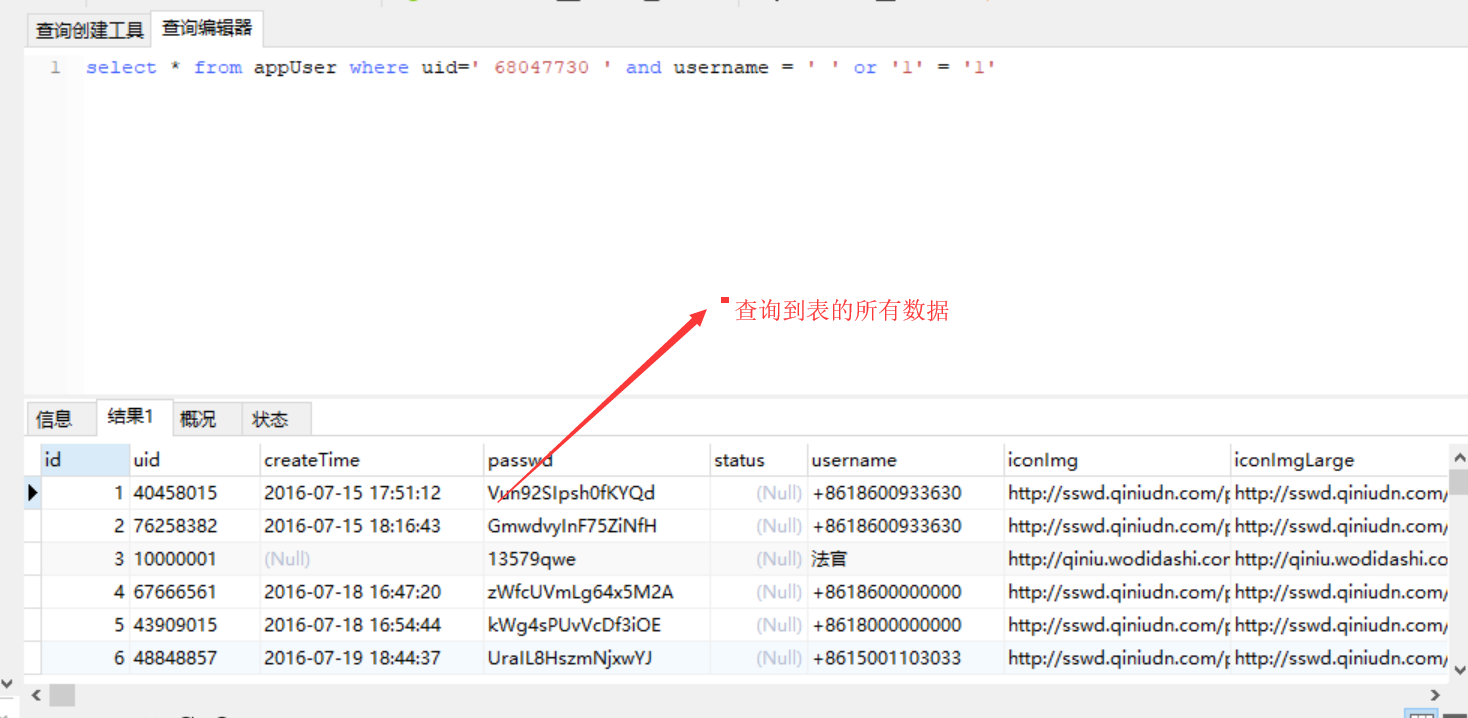
入参就变为：

data={“url”:”/v3/user/userinfo”,”params”:’{“uid”:”68047730”,”username”:” ' or '1' = '1”}’}

这时服务端接受到的参数，sql拼接语句为：

string sql= " select \* from appUser where uid=' 68047730 ' and username = ' ' or '1' = '1' ;

我们来分析这条Sql语句，'1' = '1'恒成立永远都是true,那么where条件判断就无效，相当于select \* from appUser 查询出了appUser表里的所有数据，这时一个相对危险的！！获取到了咱们玩吧所有用户数据。



## 2）接口sql漏洞测试用例设计之字符串注入？

Come on~~例如玩吧有个登陆接口 接口地址为/v3/user/login，参数为uid,passwd两个参数。

我们在调用接口请求的时候，例如小波的username为123456,passwd为123456

正常请求数据为：

data={“url”:” /v3/user/login”,”params”:’{“username”:”yangjianbo”,”passwd”:”123456”}’}

这时服务端接受到请求后，对接收到的uid，passwd两个参数进行查询校验,sql拼接语句为：

string sql= " select \* from appUser where username =' " + username + " ' and passwd = ' " +passwd+ " ' ";

接收到参数拼接后的Sql语句为：

string sql= " select \* from appUser where username =' yangjianbo' and passwd = '123456 ';

查询到这条记录后小波就可以正常登陆啦~~

我们观察到 username,passwd是个string类型的参数，字符串注入就是把参数变成sql语句带有注释的字符串 屏蔽后面的条件判断语句。

Mysql有两种注释方法

1: ‘#’方法注释，’#’后的所有语句都当成注释来处理了

这样我们就可以进行随意入参啦~~username入参我们可设计为yangjianbo'#,然后进行入参调接口吧~~

这时Data为:

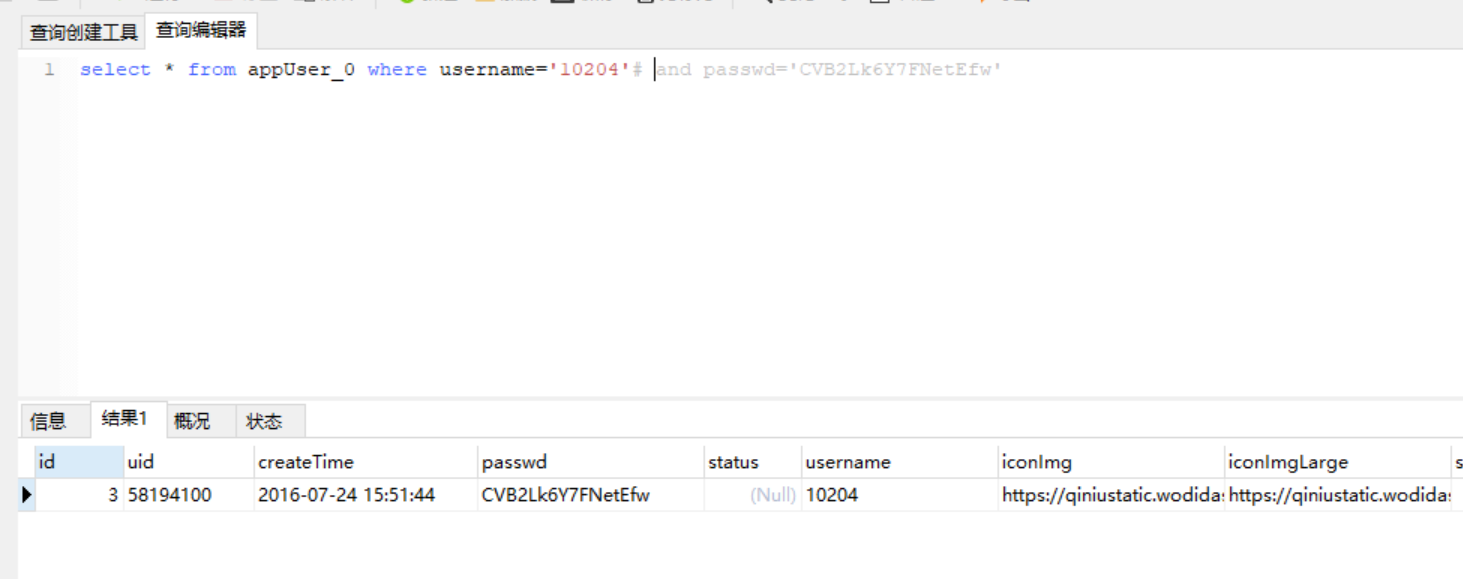
data={“url”:”/v3/user/login”,”params”:’{“username”:” yangjianbo'#”,”passwd”:”123456”}’}

服务端接收到入参，拼接sql后的语句为：

string sql= " select \* from appUser where username =' yangjianbo'#' and passwd = '123456 ';

我们来分析这条语句：

单引号已经闭合username左边的单引号，#号后面的sql已经注释了，那么只要知道username有这么个用户，密码随便输入，那么是不是就可以免密码登陆了啊,这是不是相当危险呢？



2：’-- ’方法注释，--后面有个空格，--后面的语句都会被注释

同理，username设计为yangjianbo'-- ,然后进行入参调用接口吧~~

这时data数据为：

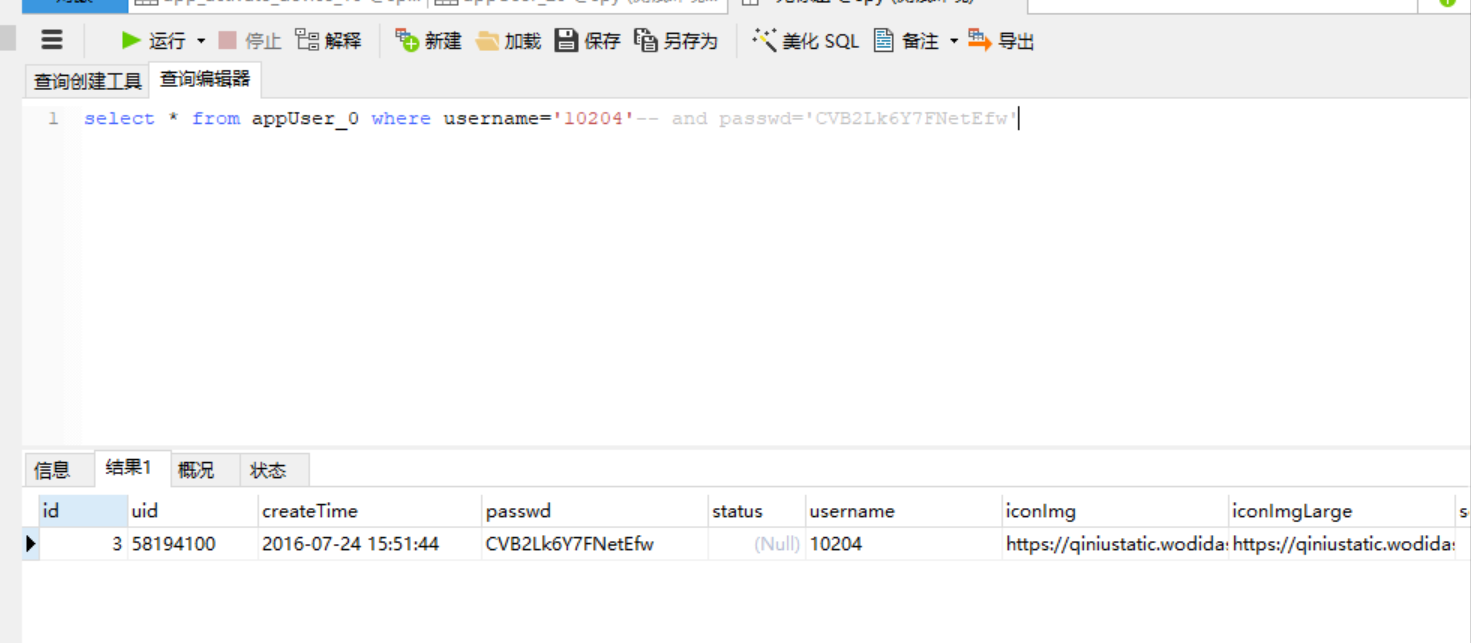
data={“url”:”/v3/user/login”,”params”:’{“username”:” yangjianbo'-- ”,”passwd”:”123456”}’}

服务端接收到入参，拼接sql后的语句为：

string sql= " select \* from appUser where username =' yangjianbo'--' and passwd = '123456 ';

我们来分析这条语句：

单引号已经闭合username左边的单引号，’-- ’后面的Sql语句已经被注释，也是可以随便输入密码都可以登陆~~这样是不是很危险呢？



好了~~到此接口接口测试用例设计之sql漏洞注入已经讲解完了~~

玩吧 QA 杨建波