# CTF学习笔记——sql注入(2)

原创

Obs_cure 于 2020-08-16 17:53:12 发布 146 收藏

文章标签： 网络安全

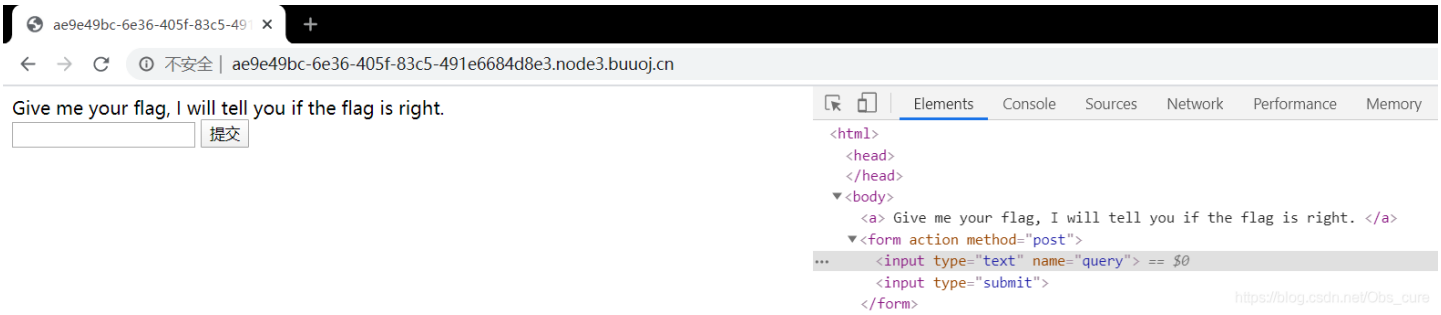


本文链接：https://blog.csdn.net/Obs_cure/article/details/108023807



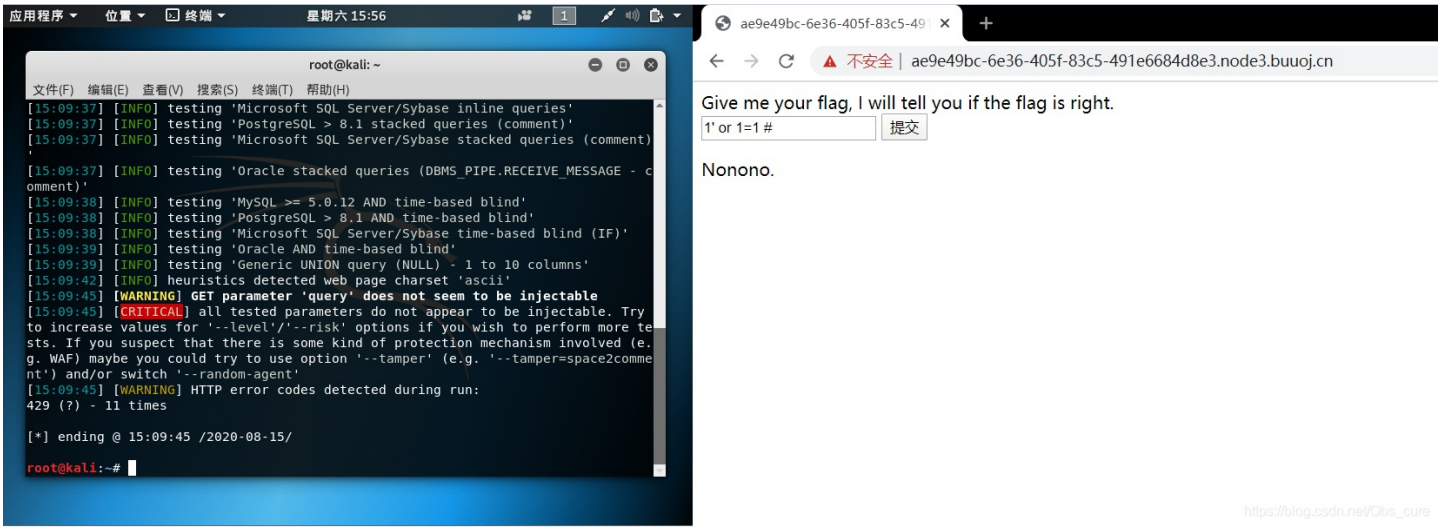
## 一、[SUCTF 2019]EasySQL

### 1.题目



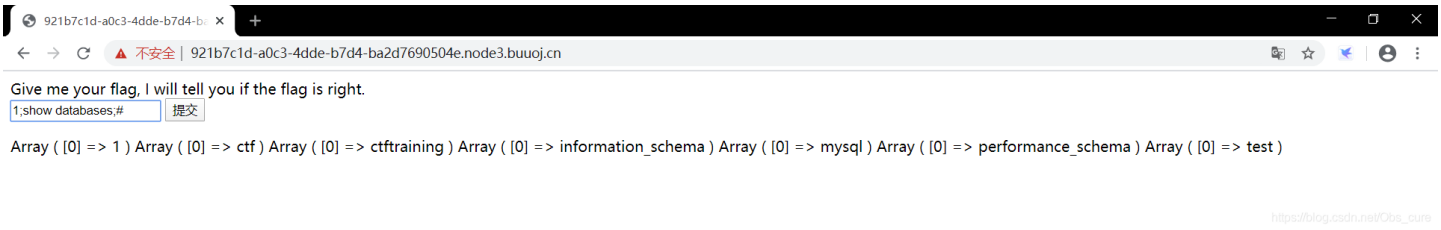

### 2.解题步骤

看标题就知道是关于sql注入的题目。老规矩，先跑一下sqlmap，再用1' or 1=1 #试试。




Nonono. 不死心再试一下别的语句呢？

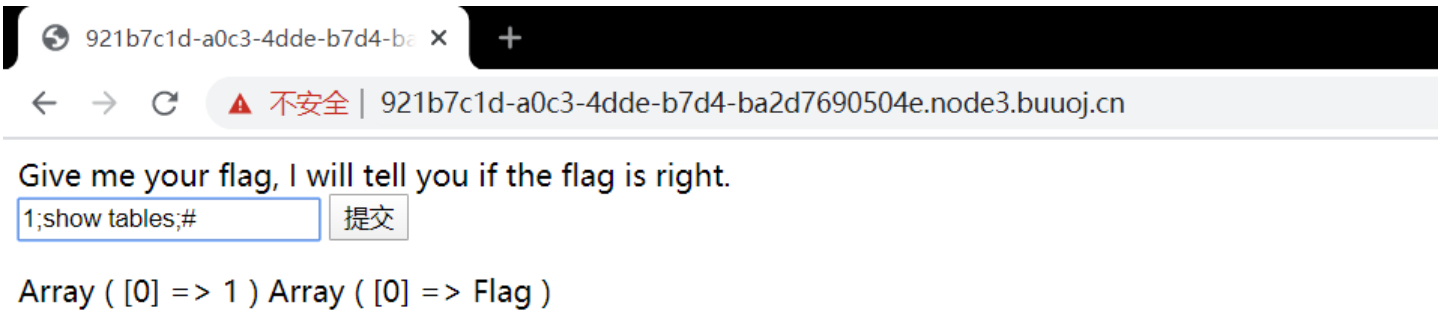
```
1;show databases;#
```





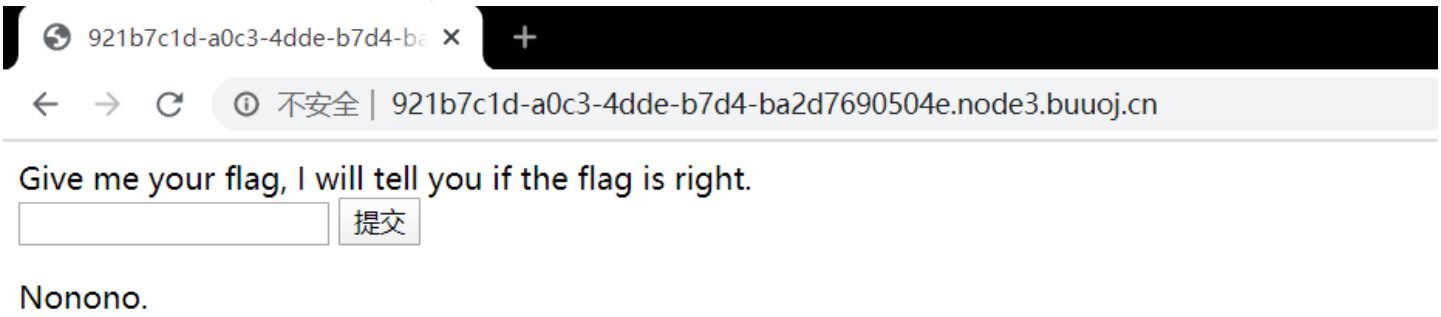

成功的把数据库爆出来了，说明存在堆叠注入的漏洞。(这里有个疑惑的点是上个博客的堆叠注入是1';用于闭合，这里不加单引号就可以，加上单引号就不好使了，不明白为什么 ...可能是后台语句中屏蔽了单引号或者没有这个闭合吧)
继续爆表或者其他信息试试呢

```
1;show tables;#
```






爆出了表名，非常明显的有一个flag表。尝试了上篇博客的注入手法，毫不意外的失败了。应该是过滤了某些关键字失效的（比如ascii码转义）。于是开始搜索writeup。






嗯？师傅们在跑注入漏洞的时候会直接跑字典检查关键字有没有被过滤？隔壁小孩馋哭了~看了几篇博客发现师傅们也没有头绪，这题内置的sql语句为：

```
sql="select".post['query']."||flag from Flag";
```

先贴师傅的writeup，这里直接照搬了EasySQL_qq_43619533的博客-CSDN博客_easysql

解法1

输入的内容为`*,1`

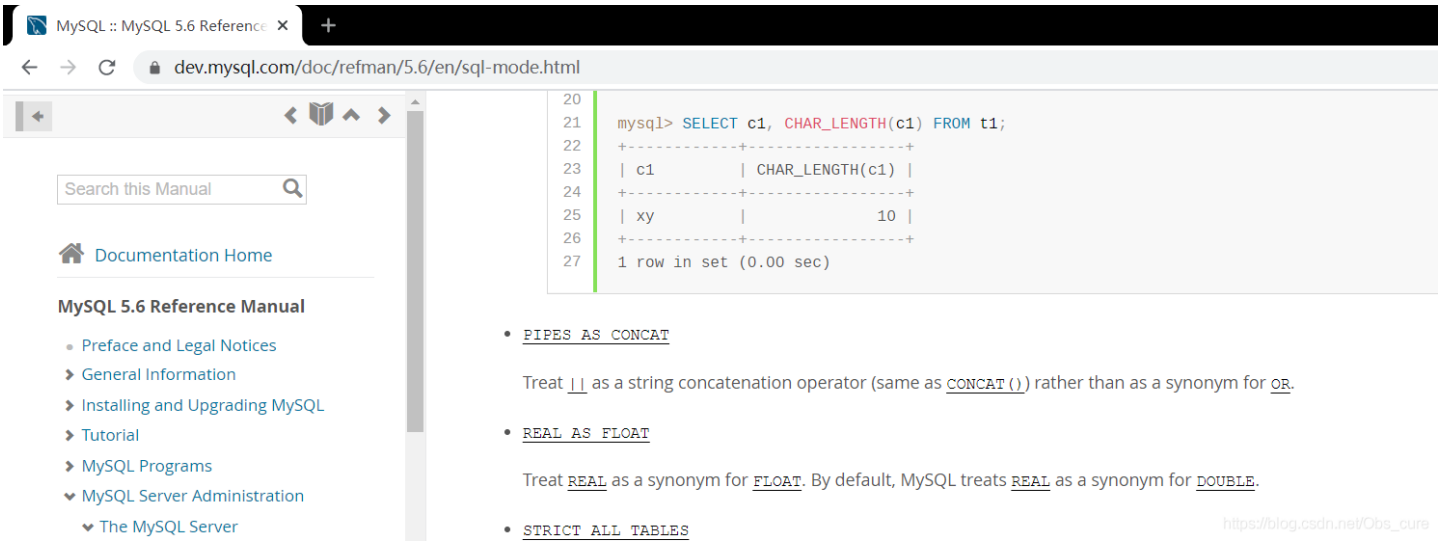
如果`$post['query']`的数据为`*,1`，sql语句就变成了`select *,1||flag from Flag`，也就是`select *,1 from Flag`，也就是直接查询出了Flag表中的所有内容

解法2

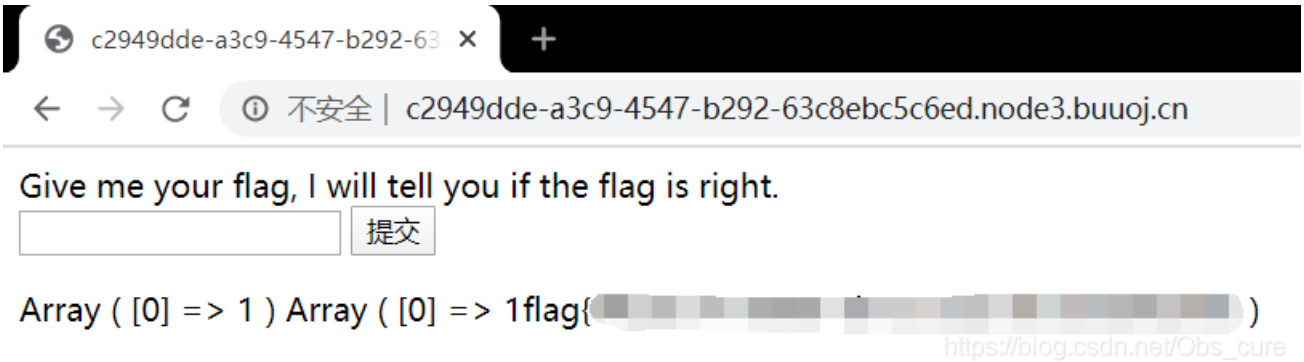
输入的内容为`1;set sql_mode=pipes_as_concat;select 1`

其中`set sql_mode=pipes_as_concat;`的作用为将||的作用由or变为拼接字符串，执行的语句分别为select 1和set `sql_mode=pipes_as_concat`和`select 1||flag from Flag`，读出flag

这里贴一个pipes_as_concat模式的作用，在官方手册上是这么写的：将||视为字符串连接操作符(与CONCAT()相同)，而不是OR的同义词。




妙啊~佩服找到这个漏洞的人，绕过了flag被过滤的情况。得到了flag




## 3.总结

1.居然可以用字典跑被过滤的关键字，比手工注方便多了，要去找个字典再尝试一下这道题

2.||转concat()这个应该算是个需要记住的漏洞，虽然还不明白师傅怎么判断出来后端的语句中含有||的…

3.sql语句还是好难…要系统的学一下sql语句

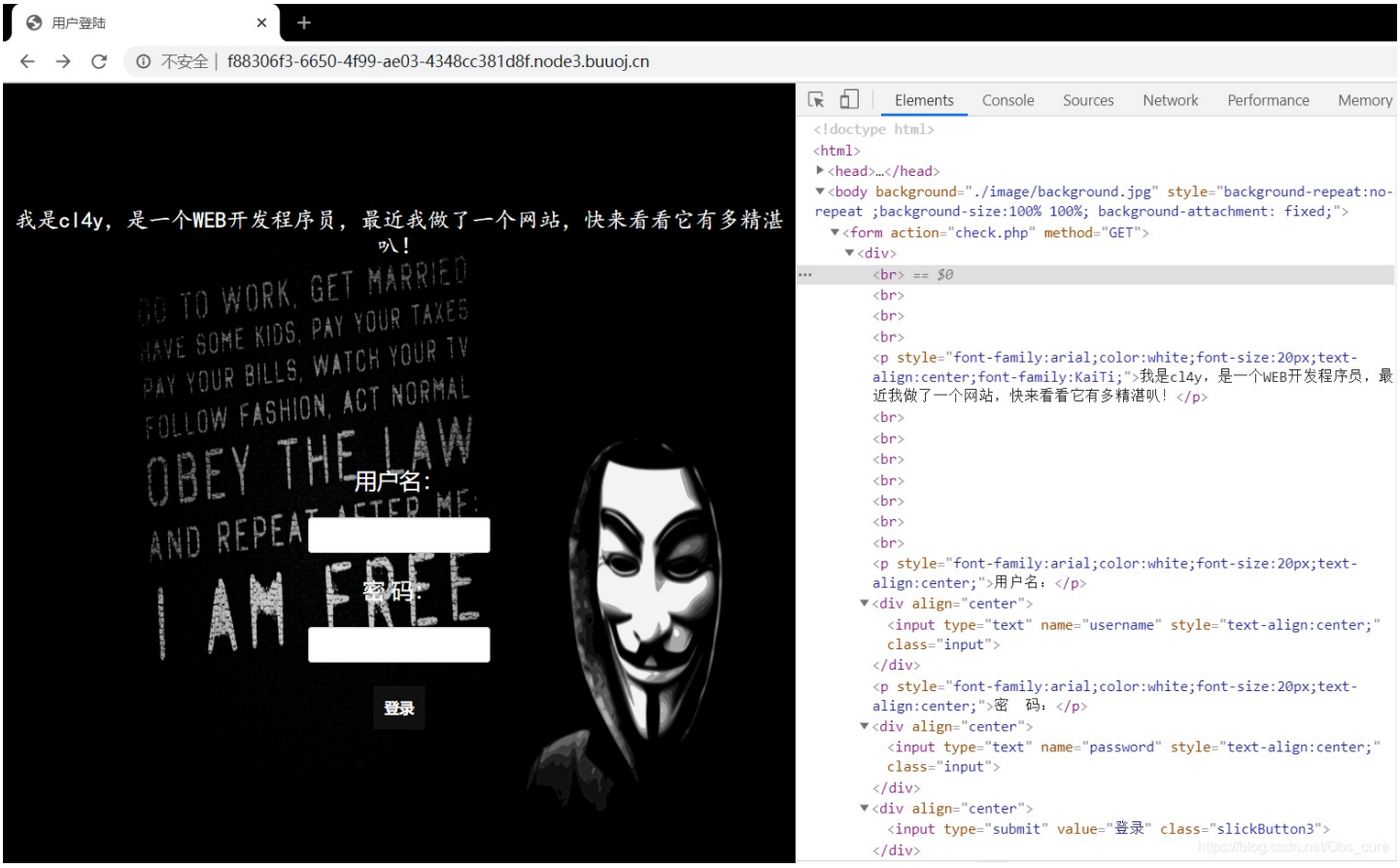
4。目前只学会了堆叠注入的一种漏洞，在看writeup的过程中发现还有联合查询等方式，有时间可以整理一个漏洞查询语句集。

## 4.参考资料

- SUCTF_2019_部分复现_Lethe's Blog-CSDN博客_suctf
- [SUCTF 2019]EasySQL - 育良书记 - 博客园
- [SUCTF 2019]EasySQL_qq_43619533的博客-CSDN博客_easysql
- 刷题记录：[SUCTF 2019]EasySQL - MustaphaMond - 博客园
- MySQL :: MySQL 5.6 Reference Manual :: 5.1.10 Server SQL Modes

## 二、[极客大挑战 2019]EasySQL

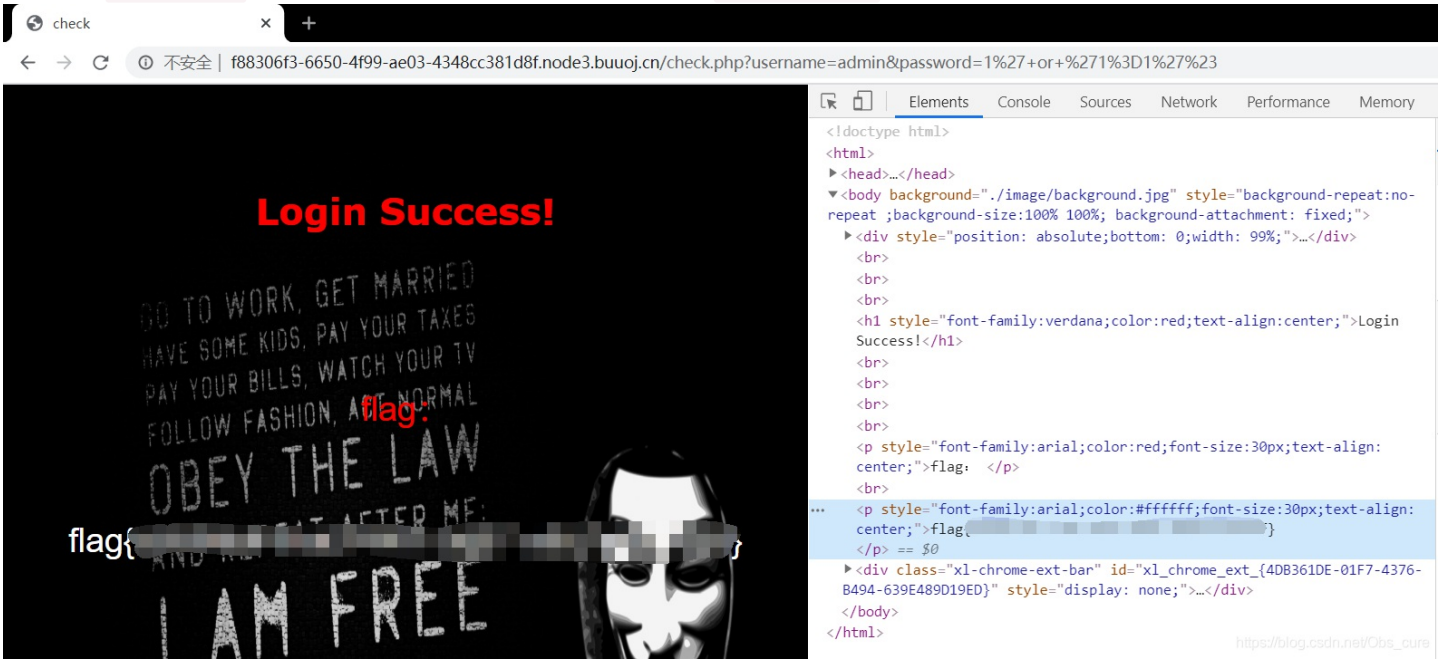
### 1.题目




### 2.解题步骤

用的是get方法！先试试sql注入语句测试吧`1’ or 1=1 #`


额，这里懵了一下，还没做过这样的题，看了一眼writeup，需要构造一个万能密码，也就是说需要判断密码恒真就行了，所以密码输入`1‘ or 1=1#`是对的，但是有些变量需要加引号...所以`1‘ or ’1=1‘#`就可以了...




## 3.总结

1.单引号真是个神奇的东西，下次注入的时候会排列组合尝试的...
2.这种登陆方式的网站注入原来只是判断的密码，涨姿势了