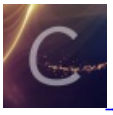
ISCC2018 Misc WriteUp

原创

lacoucou 于 2018-05-25 23:37:46 发布 1035 收藏
分类专栏： ctf 文章标签： ISCC 2018 WriteUp

本文链接：https://blog.csdn.net/lacoucou/article/details/80243051



1.What is that? [分值:50]

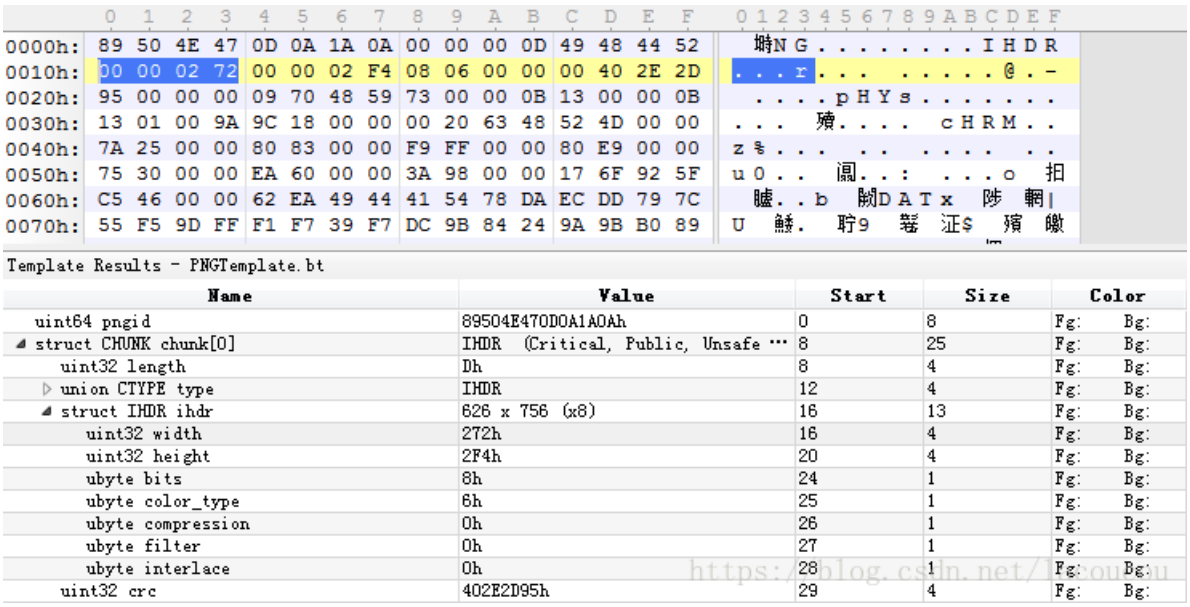
题目描述：

```
Where is the FLAG?
文件下载地址：
http://iscc2018.isclab.org.cn:4000/static/uploads/e8b1b391b0fec74623d43950fb95458a/ISCC-MISC05.rar
```

附件中为一张图片。png格式：

用010editor未发现有什么附加数据。其他的文件信息也没发现什么线索。

上传到谷歌识图显示分辨率太大，分辨率太大，分辨率太大。文件右键信息是625*500.这明显不大，用010editor把分辨率改大。

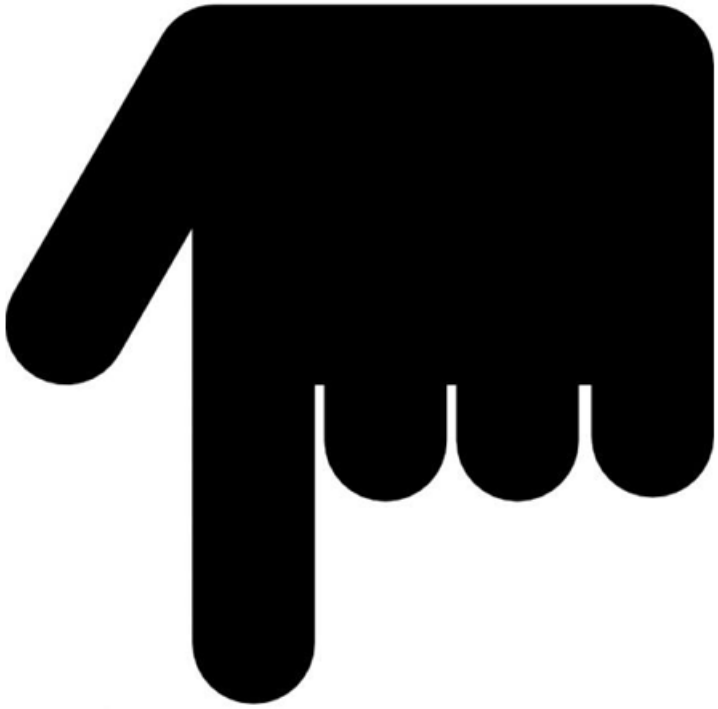


	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000h:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52
0010h:	00	00	02	72	00	00	02	F4	08	06	00	00	00	40	2E	2D
0020h:	95	00	00	00	09	70	48	59	73	00	00	0B	13	00	00	0B
0030h:	13	01	00	9A	9C	18	00	00	00	20	63	48	52	4D	00	00
0040h:	7A	25	00	00	80	83	00	00	F9	FF	00	00	80	E9	00	00
0050h:	75	30	00	00	EA	60	00	00	3A	98	00	00	17	6F	92	5F
0060h:	C5	46	00	00	62	EA	49	44	41	54	78	DA	EC	DD	79	7C
0070h:	55	F5	9D	FF	F1	F7	39	F7	DC	9B	84	24	9A	9B	B0	89

Template Results - PNGTemplate.bt

Name	Value	Start	Size	Color
uint64 pngid	89504E470D0A1A0Ah	0	8	Fg: Bg:
struct CHUNK chunk[0]	IHDR (Critical, Public, Unsafe ...	8	25	Fg: Bg:
uint32 length	Dh	8	4	Fg: Bg:
union CTYPE type	IHDR	12	4	Fg: Bg:
struct IHDR ihdr	626 x 756 (x8)	16	13	Fg: Bg:
uint32 width	272h	16	4	Fg: Bg:
uint32 height	2F4h	20	4	Fg: Bg:
ubyte bits	8h	24	1	Fg: Bg:
ubyte color_type	6h	25	1	Fg: Bg:
ubyte compression	0h	26	1	Fg: Bg:
ubyte filter	0h	27	1	Fg: Bg:
ubyte interlace	0h	28	1	Fg: Bg:
uint32 crc	402E2D95h	29	4	Fg: Bg:

就看到flag了。



Flag={_Welcome_To_ISCC_2018_}

2.秘密电报 [分值:50]

题目描述：

```
秘密电报：
知识就是力量 ABAAAABABBABAAAABABAAABAAABAAABAABAAAABAAAABA
```

搜一下就知道是培根密码了。网上找了一段代码：

```
# coding:utf8

import re

alphabet = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x'

first_cipher = ["aaaaa","aaaab","aaaba","aaabb","aabaa","aabab","aabba","aabbb","abaaa","abaab","ababa","ab

second_cipher = ["aaaaa","aaaab","aaaba","aaabb","aabaa","aabab","aabba","aabbb","abaaa","abaaa","abaab","a

def encode():
    upper_flag = False # 用于判断输入是否为大写
    string = raw_input("please input string to encode:\n")
    if string.isupper():
        upper_flag = True
        string = string.lower()
    e_string1 = ""
    e_string2 = ""
    for index in string:
```

```
    for index in string:
        for i in range(0,26):
            if index == alphabet[i]:
                e_string1 += first_cipher[i]
                e_string2 += second_cipher[i]
                break
    if upper_flag:
        e_string1 = e_string1.upper()
        e_string2 = e_string2.upper()
    print "first encode method result is:\n"+e_string1
    print "second encode method result is:\n"+e_string2
    return


def decode():
    upper_flag = False  # 用于判断输入是否为大写
    e_string = raw_input("please input string to decode:\n")
    if e_string.isupper():
        upper_flag = True
        e_string = e_string.lower()
    e_array = re.findall(".{5}",e_string)
    d_string1 = ""
    d_string2 = ""
    for index in e_array:
        for i in range(0,26):
            if index == first_cipher[i]:
                d_string1 += alphabet[i]
            if index == second_cipher[i]:
                d_string2 += alphabet[i]
    if upper_flag:
        d_string1 = d_string1.upper()
        d_string2 = d_string2.upper()
    print "first decode method result is:\n"+d_string1
    print "second decode method result is:\n"+d_string2
    return


if __name__ == '__main__':
    print "\t\tcoding by qux"
    while True:
        print "\t*******Bacon Encode_Decode System*******"
        print "input should be only lowercase or uppercase,cipher just include a,b(or A,B)"
        print "1.encode\n2.decode\n3.exit"
        s_number = raw_input("please input number to choose\n")
        if s_number == "1":
            encode()
            raw_input()
        elif s_number == "2":
            decode()
            raw_input()
        elif s_number == "3":
            break
        else:
            continue
```

运行得到答案：

```
input should be only lowercase or uppercase,cipher just include a,b(or A,B)
1.encode
2.decode
3.exit
please input number to choose
2
please input string to decode:
ABAAAABABBABAAAABABAAABAAABAAABAABAAAABAAAABA
first decode method result is:
ILIKEISCC                              <---  这个是答案
second decode method result is:
IJMIJLEIJTCC
```

3.重重谍影 [分值:100]

题目描述:

```
重重谍影
这是一道脑洞题，简单的要命。层层迷雾之后就是答案，刹那便是永恒。南无阿弥陀佛。

Vm0wd2QyVkZOVWRXV0doVlYwZG9WVll3WkRSV2JGbDNXa1JTVjAxWGVGWlZNakExVjBaS2RHVkljRnBXVm5CUVZqQmtTMUl4VG5OaFJtUlh
```

这是一道脑洞相当大的题目，能做出来完全靠运气,靠某些网站。

下边的字符串是base64,解码之后会发现有%3d %0a这样的字符，猜测是urlencode.

所以同时用base64和urlencode解密。

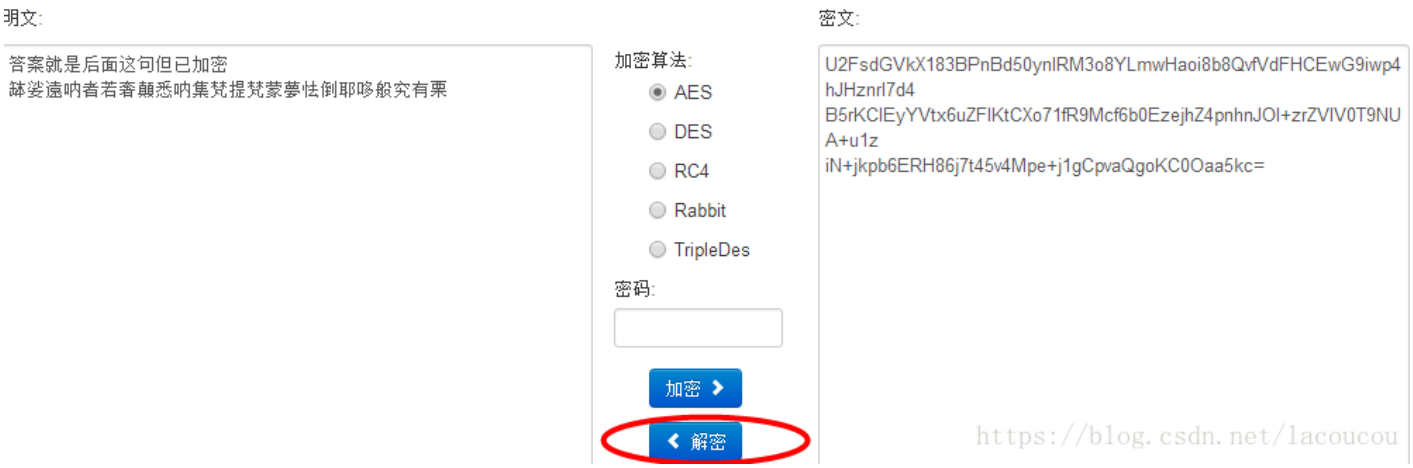
```
import base64
from urllib import unquote

str_text="Vm0wd2QyVkZOVWRXV0doVlYwZG9WVll3WkRSV2JGbDNXa1JTVjAxWGVGWlZNakExVjBaS2RHVkljRnBXVm5CUVZqQmtTMUl4V


str_xx=str_text
for i in range(0,100):
    str_xx=base64.b64decode(str_xx)
    str_xx=unquote(str_xx)
    print i,str_xx
```

运行10次左右就崩溃了。发现第七次是明文：

```
7 U2FsdGVkX183BPnBd50ynIRM3o8YLmwHaoi8b8QvfVdFHCEwG9iwp4hJHznrl7d4
B5rKClEyYVtx6uZFIKtCXo71fR9Mcf6b0EzejhZ4pnhnJOl+zrZVlV0T9NUA+u1z
iN+jkpb6ERH86j7t45v4Mpe+j1gCpvaQgoKC0Oaa5kc=
```

第一次做到这里就卡住了。。。。。后来佛系到答案。 上边的字符串是AES加密的字符串(请问怎么看出来的？）

在此网站http://tool.oschina.net/encrypt 解密，只能用这个网站哦，因为只有这个网站解密不要密码。



明文的第二句就是答案。要解开这个谜底需要另外一个佛系网站:http://www.keyfc.net/bbs/tools/tudoucode.aspx

解密方法：

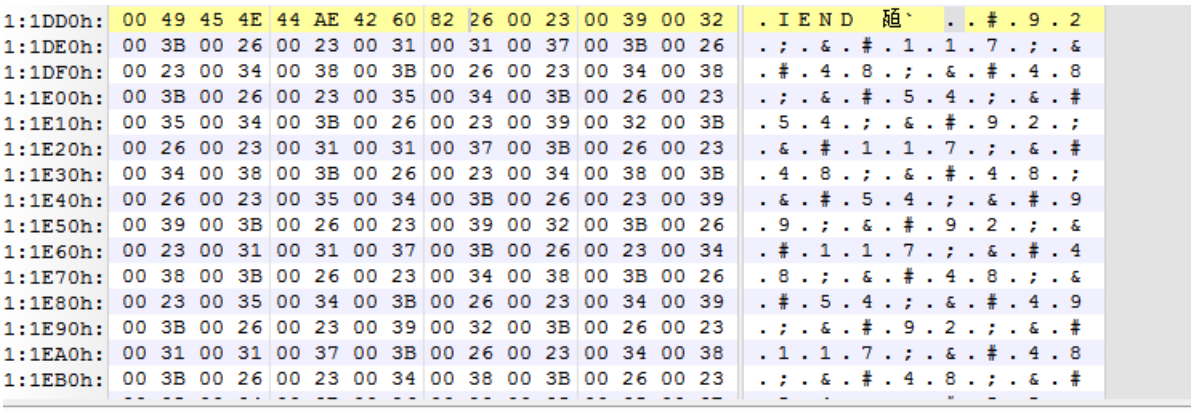
在密文前面添加“佛曰“ 然后点击按钮，解出来的就是答案。

4.有趣的ISCC [分值:100]

题目描述:

> 在ISCC的平台上，跟小伙伴们一起闯关，是不是很有趣啊！！！
> 猜猜我在图片中隐藏了什么？

一张图片，010editor发现结尾有数据隐藏：



```
1:1DD0h: 00 49 45 4E 44 AE 42 60 82 26 00 23 00 39 00 32   .IEND 砸` .&.#.9.2
1:1DE0h: 00 3B 00 26 00 23 00 31 00 31 00 37 00 3B 00 26   .;.&.#.1.1.7.;.&
1:1DF0h: 00 23 00 34 00 38 00 3B 00 26 00 23 00 34 00 38   .#.4.8.;.&.#.4.8
1:1E00h: 00 3B 00 26 00 23 00 35 00 34 00 3B 00 26 00 23   .;.&.#.5.4.;.&.#
1:1E10h: 00 35 00 34 00 3B 00 26 00 23 00 39 00 32 00 3B   .5.4.;.&.#.9.2.;
1:1E20h: 00 26 00 23 00 31 00 31 00 37 00 3B 00 26 00 23   .&.#.1.1.7.;.&.#
1:1E30h: 00 34 00 38 00 3B 00 26 00 23 00 34 00 38 00 3B   .4.8.;.&.#.4.8.;
1:1E40h: 00 26 00 23 00 35 00 34 00 3B 00 26 00 23 00 39   .&.#.5.4.;.&.#.9
1:1E50h: 00 39 00 3B 00 26 00 23 00 39 00 32 00 3B 00 26   .9.;.&.#.9.2.;.&
1:1E60h: 00 23 00 31 00 31 00 37 00 3B 00 26 00 23 00 34   .#.1.1.7.;.&.#.4
1:1E70h: 00 38 00 3B 00 26 00 23 00 34 00 38 00 3B 00 26   .8.;.&.#.4.8.;.&
1:1E80h: 00 23 00 35 00 34 00 3B 00 26 00 23 00 34 00 39   .#.5.4.;.&.#.4.9
1:1E90h: 00 3B 00 26 00 23 00 39 00 32 00 3B 00 26 00 23   .;.&.#.9.2.;.&.#
1:1EA0h: 00 31 00 31 00 37 00 3B 00 26 00 23 00 34 00 38   .1.1.7.;.&.#.4.8
1:1EB0h: 00 3B 00 26 00 23 00 34 00 38 00 3B 00 26 00 23   .;.&.#.4.8.;.&.#
```

Template Results - PNGTemplate.bt

Name	Value	Start	Size	Color
uint64 pngid	89504E470D0A1A0Ah	0	8	Fg: Bg:
struct CHUNK chunk[0]	IHDR (Critical, Public, Unsafe ···	8	25	Fg: Bg:
struct CHUNK chunk[1]	gAMA (Ancillary, Public, Unsafe···	33	16	Fg: Bg:
struct CHUNK chunk[2]	cHRM (Ancillary, Public, Unsafe···	49	44	Fg: Bg:
struct CHUNK chunk[3]	bKGD (Ancillary, Public, Unsafe···	93	18	Fg: Bg:
struct CHUNK chunk[4]	pHYs (Ancillary, Public, Safe t···	111	21	Fg: Bg:
struct CHUNK chunk[5]	tIME (Ancillary, Public, Unsafe···	132	19	Fg: Bg:
struct CHUNK chunk[6]	IDAT (Critical, Public, Unsafe ···	151	32780	Fg: Bg:
struct CHUNK chunk[7]	IDAT (Critical, Public, Unsafe ···	32931	32780	Fg: Bg:
struct CHUNK chunk[8]	IDAT (Critical, Public, Unsafe ···	65711	7063	Fg: Bg:
struct CHUNK chunk[9]	tEXt (Ancillary, Public, Safe t···	72774	49	Fg: Bg:
struct CHUNK chunk[10]	tEXt (Ancillary, Public, Safe t···	72823	49	Fg: Bg:
struct CHUNK chunk[11]	tEXt (Ancillary, Public, Safe t···	72872	90	Fg: Bg:
struct CHUNK chunk[12]	tEXt (Ancillary, Public, Safe t···	72962	36	Fg: Bg:
struct CHUNK chunk[13]	tEXt (Ancillary, Public, Safe t···	72998	37	Fg: Bg:
struct CHUNK chunk[14]	tEXt (Ancillary, Public, Safe t···	73035	36	Fg: Bg:
struct CHUNK chunk[15]	tEXt (Ancillary, Public, Safe t···	73071	37	Fg: Bg:
struct CHUNK chunk[16]	tEXt (Ancillary, Public, Safe t···	73108	27	Fg: Bg:
struct CHUNK chunk[17]	tEXt (Ancillary, Public, Safe t···	73135	30	Fg: Bg:
struct CHUNK chunk[18]	IEND (Critical, Public, Unsafe ···	73165	12	Fg: Bg:
struct CHUNK chunk[19]		73177	0	Fg: Bg:

提取出来大概是这样：

```
&#92;&#117;&#48;&#48;&#54;&#54;&#92;&#117;&#48;&#48;&#54;&#99;&#92;&#117;&#48;&#48;&#54;&#49;&#92;&#117;&#48;&#48;&#54;&#55;&#92;&#117;&#48;&#48;&#55;&#98;&#92;&#117;&#48;&#48;&#54;&#57;&#92;&#117;&#48;&#48;&#55;&#51;&#92;&#117;&#48;&#48;&#54;&#51;&#92;&#117;&#48;&#48;&#54;&#51;&#92;&#117;&#48;&#48;&#50;&#48;&#92;&#117;&#48;&#48;&#54;&#57;&#92;&#117;&#48;&#48;&#55;&#51;&#92;&#117;&#48;&#48;&#50;&#48;&#92;&#117;&#48;&#48;&#54;&#54;&#92;&#117;&#48;&#48;&#55;&#53;&#92;&#117;&#48;&#48;&#54;&#101;&#92;&#117;&#48;&#48;&#55;&#100;
```


粘贴到博客里就自动变unicode编码了。可在这里转换http://tools.jb51.net/transcoding/chinese2unicode

```
\u0066\u006c\u0061\u0067\u007b\u0069\u0073\u0063\u0063\u0020\u0069\u0073\u0020\u0066\u0075\u006e\u007d
```

unicode再转中文：

flag{iscc is fun}

5.Where is the FLAG [分值:100]

题目描述:

```
不只是Logo
```

等待更新.....

6.凯撒十三世[分值:150]

题目描述:

```
凯撒十三世在学会使用键盘后，向你扔了一串字符：“ebdgc697g95w3”，猜猜它吧。
```

根据描述应该是凯撒密码。13世估计指的是次数，键盘应该是键盘加密。

http://www.zjslove.com/3.decode/kaisa/index.html

```
第1次解密:ebdgc697g95w3
第2次解密:dacfb697f95v3
第3次解密:czbea697e95u3
第4次解密:byadz697d95t3
第5次解密:axzcy697c95s3
第6次解密:zwybx697b95r3
第7次解密:yvxaw697a95q3
第8次解密:xuwzv697z95p3
第9次解密:wtvyu697y95o3
第10次解密:vsuxt697x95n3
第11次解密:urtws697w95m3
第12次解密:tqsvr697v95l3
第13次解密:spruq697u95k3
第14次解密:roqtp697t95j3       flag:yougotme
第15次解密:qnpso697s95i3
第16次解密:pmorn697r95h3
第17次解密:olnqm697q95g3
第18次解密:nkmpl697p95f3
第19次解密:mjlok697o95e3    k
第20次解密:liknj697n95d3
第21次解密:khjmi697m95c3
第22次解密:jgilh697l95b3
第23次解密:ifhkg697k95a3
第24次解密:hegjf697j95z3
第25次解密:gdfie697i95y3
第26次解密:fcehd697h95x3
```

上边是第十四次，是因为第一次实际上是第0次，什么也没变。这里的键盘加密指的是

```
roqtp697t95j3
```

键盘上这些字母下方的字母。

7.一只猫的心思[分值:150]

题目描述:

`你能读懂它的心思吗？`

又是图片结尾有附加数据。提取之后是一个word文档，打开是一串佛系文字，同样在佛系网站解下：

文本前添加如是我闻：然后解密，一堆16进制字符串，打印出来是base64字符串，再解是base32的字符串。

```
import binascii
import base64

str1="523156615245644E536C564856544E565130354B553064524D6C524E546B4A56535655795645644F553052485754 4A4553553

de1=binascii.a2b_hex(str1)
print de1

de2=base64.b64decode(de1)

print de2

de3=base64.b32decode(de2)
print de3


de4=binascii.a2b_hex(de3)
print de4


de5=base64.b64decode(de4)
print de5


de6=base64.b32decode(de5)
print de6


de7=binascii.a2b_hex(de6)
print de7
```

来来回回7-8此 终于解出来一个短的：

F1a9_is_l5cc_ZOl8_G3TP01NT

然后就卡住了。。。。。。。。。。。。

原来 l5cc_ZOl8_G3TP01NT 这个是答案。还以为还有一层加密。

8.暴力XX不可取[分值:150]

题目描述:

A同学要去参加今年的ISCC。大赛在即，A同学准备了一批暴力破解工具，你感觉这个靠谱吗？
下载下来是一个zip包,题目提示暴力破解不可取。可能是zip伪加密。

https://blog.csdn.net/kajweb/article/details/76474476

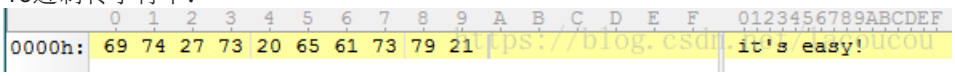
参考上边的链接，把偏移0x3f 改为08 解出来txt
vfppjrnerpbzvat

提交之后不对，应该还有加密，试了下凯撒解密：
http://www.zjslove.com/3.decode/kaisa/index.html
凯撒解了一下：
第1次解密:vfppjrnerpbzvat
第2次解密:ueooiqmdqoayuzs
第3次解密:tdnnhplcpnzxtyr
第4次解密:scmmgokbomywsxq
第5次解密:rbllfnjanlxvrwp
第6次解密:qakkemizmkwuqvo
第7次解密:pzjjdlhyljvtpun
第8次解密:oyiickgxkiusotm
第9次解密:nxhhbjfwjhtrnsl
第10次解密:mwggaievigsqmrk
第11次解密:lvffzhduhfrplqj
第12次解密:kueeygctgeqokpi
第13次解密:jtddxfbsfdpnjoh
第14次解密:isccwearecoming 这个
第15次解密:hrbbvdzqdbnlhmf
第16次解密:gqaaucypcamkgle
第17次解密:fpzztbxobzljfkd
第18次解密:eoyysawnaykiejc
第19次解密:dnxxrzvmzxjhdib
第20次解密:cmwwqyulywigcha
第21次解密:blvvpxtkxvhfbgz
第22次解密:akuuowsjwugeafy
第23次解密:zjttnvrivtfdzex
第24次解密:yissmuqhusecydw
第25次解密:xhrrltpgtrdbxcv
第26次解密:wgqqksofsqcawbu

9.数字密文 [分值:50]

题目描述:

这里有个很简单的flag，藏在下面这串数字里，猜猜吧！69742773206561737921
16进制转字符串：



4.有趣的ISCC [分值:100]

题目描述:

4.有趣的ISCC [分值:100]

题目描述:

4.有趣的ISCC [分值:100]

题目描述:

4.有趣的ISCC [分值:100]

题目描述: