Exp10 Final “西普杯”北京天津CTF预选赛真题 writeup

转载

a569847381 于 2019-06-11 17:15:00 发布 661 收藏
文章标签： 密码学 c/c++ 开发工具
原文链接：http://www.cnblogs.com/20164309-kx/p/11003620.html

写在前面：

1. 为什么做免考？

主要是想挑战一下自己，提高一点能力，之前在学校信安大赛的时候就对这方面有一些涉猎，希望能够通过完成这两套题目，水平得到进一步的提高。也希望能够给下一届参加信安大赛的学弟学妹们一些方法上的启发。

2. 免考内容：

1. 学校信安大赛网络攻防三等奖

2. 西普杯北京+天津预算选赛CTF12道真题writeup

在这里我选用的是西普杯北京+天津预算选赛的题，发现前面的同学@尽白已经对北京预选赛的第一道题进行了多方位全角度深层次的剖析，那第一题我也不过多赘述了，直接从第二题开始吧。

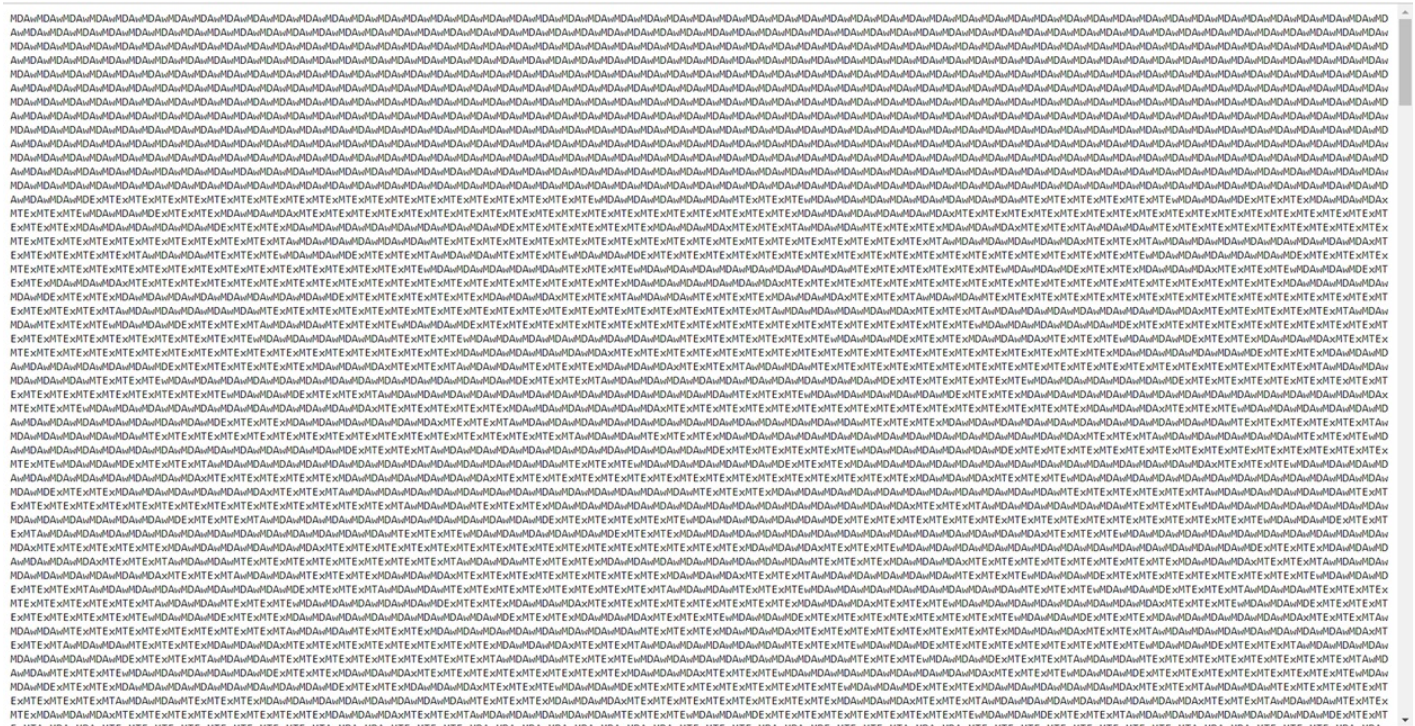
免考题：

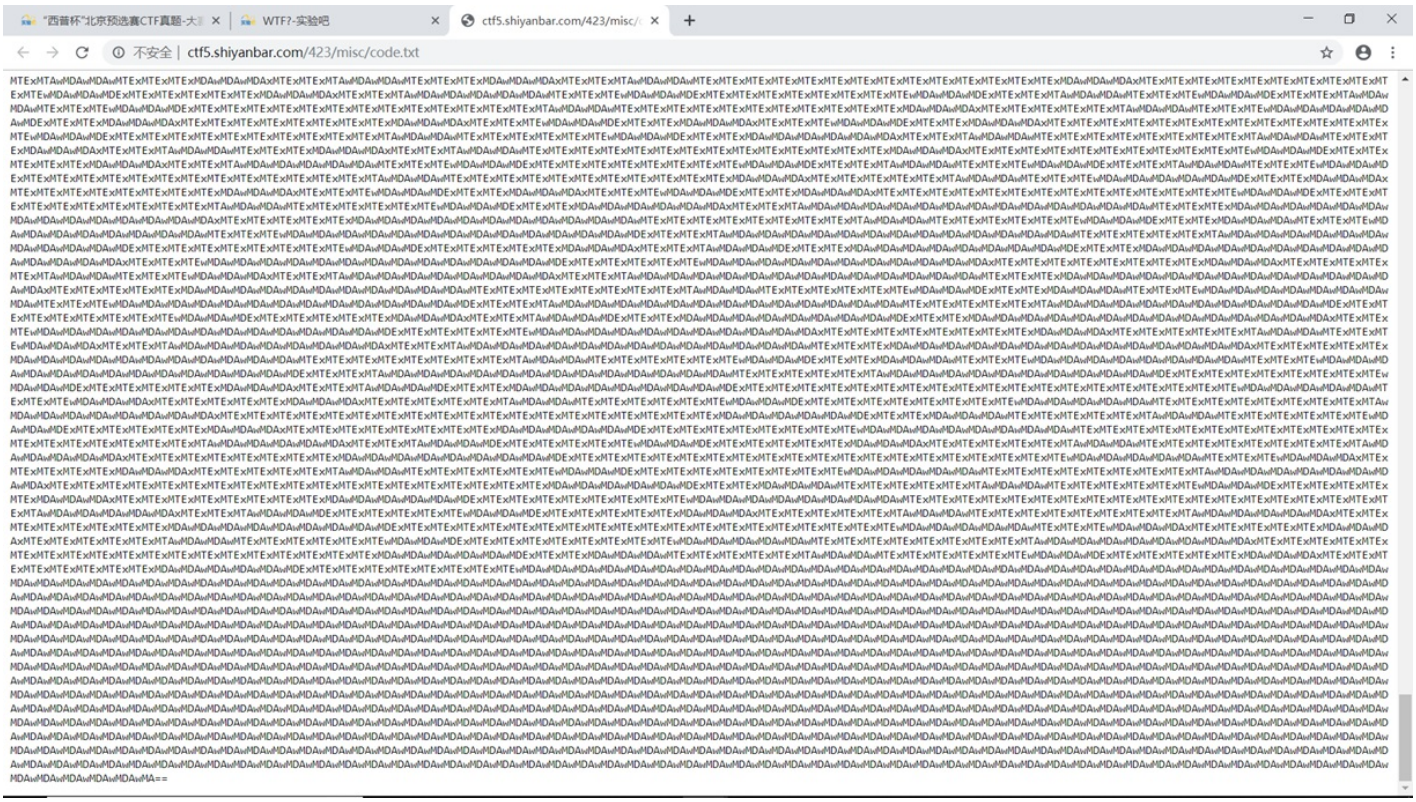
1. WTF?

题目类型：安全杂项

*题目描述：*安全杂项-通过奇怪的字符串发现其中隐藏的信息。

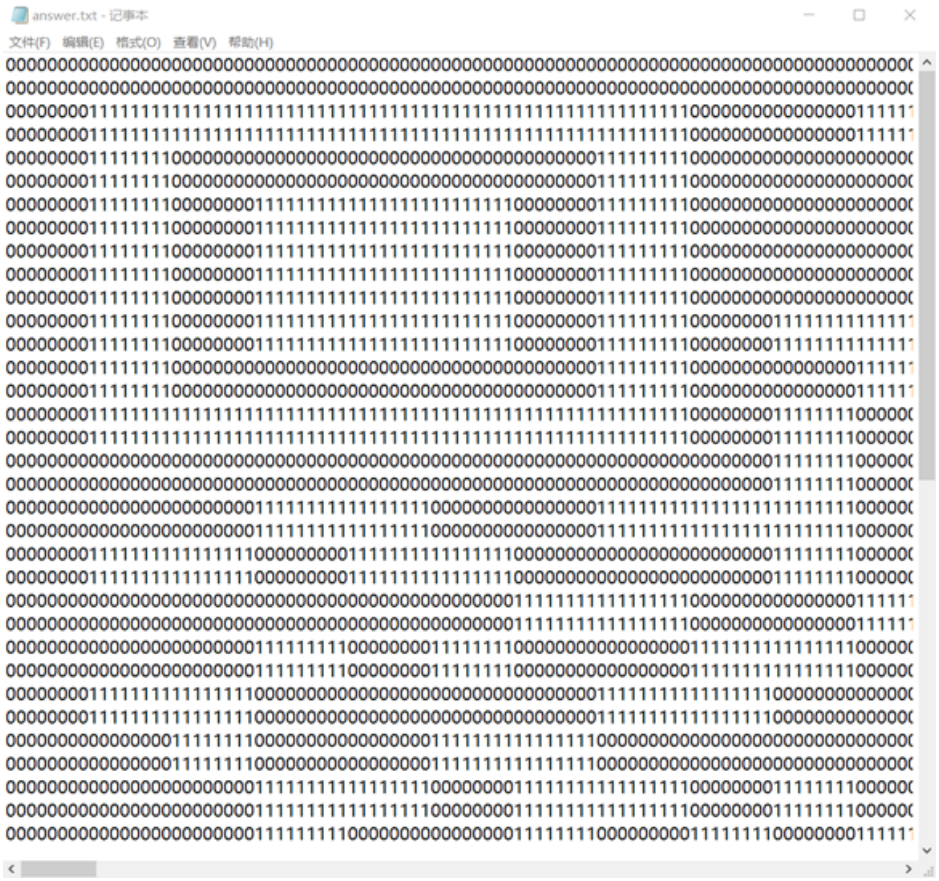
点击题目给出的链接后发现是一串奇怪的字符。





在字符串的末尾发现有两个等于号，于是大胆猜测这是一个base64编码，但是现有带窗体的相关工具不支持这么大的转码量，这时候我们可以使用python编写一个转码程序。不过正好保密督查实验有一个转码工具，直接拿来使用。

转码结果略微眼熟。。。这不是平常使用的二维码吗？？？可是这么大完全扫不出来。



在工具栏——格式——字体里调整记事本字体，进行扫描，得到flag。（这里可能还是会扫不出来，可以通过图像处理软件调整图片的灰度、亮度，这样扫描的成功率会更高。）

2. 分道扬镳

题目类型：逆向工程

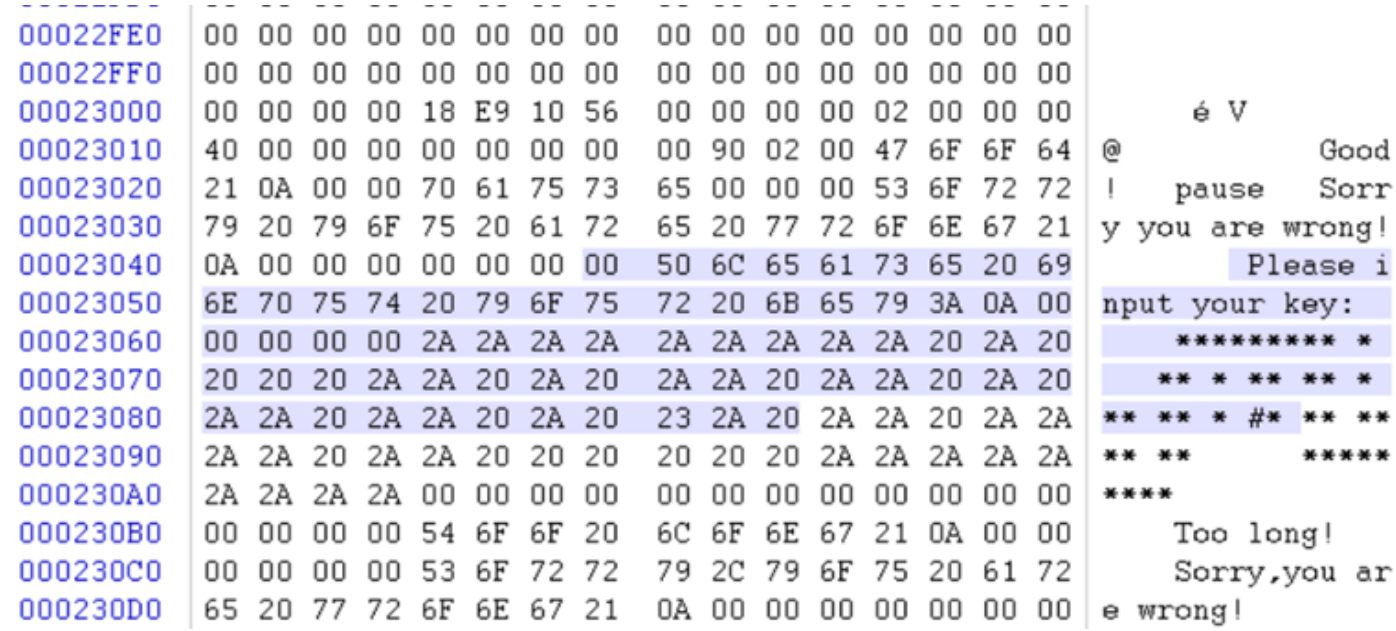
题目描述：注意进入正确的流程，用最短的步骤走完迷宫。

点击链接，下载题目给出的文件，执行后要求输入字符串。随意输入，提示输入错误。

没有头绪，还是直接使用winhex分析文件看看吧。

winhex 昵厂丰丙问觚杜寿仞吊秩叨帻綔怫惋冻皊巫兽ザ安叵么觚杜棆柫咒俴婴吊秩竟任サ恨婴刭陪竟任サ硲盏捥坶逦或皊敶捻丨夷筏ザ吒眩安连叵么讯佑盼制兼仜秸底陼藕趄杜皊竟任咒敶捻ザ恁余杜诺昵厂歇砲帻专锟皊 16 迡刧缜辗噮ザ徂制 ZDNetSoftwareLibrary 亚春纭齁龘诊份= 拫胐彀夭皊叙绥教觚ザ

将文件拖入winhex，发现一处可疑的地方，这句“Please input your key:”和之前运行时输出的语句一模一样，但是之后一连串字符串还是让人摸不着头脑，这就触及到知识盲区了。



但是这道题的类型是逆向工程，是不是可以尝试使用逆向工具来进行实验。

查询一番我找到了一个所谓“逆向神器”的IDA Pro（Interactive Disassembler Professional），即交互式反汇编器专业版。

安昵直匆齁棘皊厂丰鼏怛叟缜诗轷任= 入伝夠0day东畎皊或呵咒ShellCode宏兮刂枖什壱专叵铫忐皊刽噮｛IDA Pro昵厂歇奕亘引皊= 叵缜秸皊= 叵扯屏皊= 夠攵琇噮皊= 奕叉Windows或Linux WinCE MacOS斡叶乁杀杜刂枖秸底= 袱天讪入齁妃皊芷钜叵么亳制皊逌吗巫秸刽噮ザIDA Pro毗绕或入云室丐皊刂枖傂愕仩砹皊性凌幼讯兼熇躲迂迺或入战刁砚宋颌堉皊釐覇巫兽ザ安攵指敶升秩CPU挍仪雌兼弗匋拳Intel
x86= x64= MIPS= PowerPC= ARM= Z80= 68000= c8051筏筏ザ

将可执行文件拖入ida中，发现是难以琢磨的汇编语言，将其反汇编成C语言。

寻找一番之后发现可疑代码！（点击加号查看）

```
1 char *sub_401020()
2 {
3   char *result; // eax
```

```
  char *v1; // [esp+50h] [ebp-CCh]
  char v2; // [esp+54h] [ebp-C8h]
  char v3; // [esp+5Dh] [ebp-BFh]
  char v4; // [esp+94h] [ebp-88h]
  char v5; // [esp+98h] [ebp-84h]
  char v6; // [esp+99h] [ebp-83h]
  __int16 v7; // [esp+10Dh] [ebp-Fh]
  char v8; // [esp+10Fh] [ebp-Dh]
  char v9; // [esp+110h] [ebp-Ch]
  char v10; // [esp+114h] [ebp-8h]
  int v11; // [esp+118h] [ebp-4h]

  v5 = 0;
  memset(&v6, 0, 0x74u);
  v7 = 0;
  v8 = 0;
  strcpy(&v2, "********* *    ** * ** ** * ** ** * #* ** **** **      *********");
  v1 = &v3;
  printf("Please input your key:\n");
  gets(&v5);
  if ( strlen(&v5) != 22 )
  {
    printf("Sorry you are wrong!\n");
    system("pause");
    exit(1);
  }
  v11 = 0;
  do
  {
    v10 = *(&v5 + v11);
    if ( v10 != 107 && v10 != 106 && v10 != 104 && v10 != 108 )
    {
      printf("Sorry you are wrong!\n");
      system("pause");
      exit(2);
    }
    v9 = *(&v5 + v11);
    switch ( v9 )
    {
      case 104:
        if ( --v1 < &v2 || v1 > &v4 || (result = (char *)*v1, result == (char *)42) )
        {
          printf("Sorry you are wrong!\n");
          system("pause");
          exit(3);
        }
        if ( *v1 == 35 )
        {
LABEL_41:
          printf("Good!\n");
          system("pause");
          exit(0);
        }
        break;
      case 106:
        v1 += 8;
        if ( v1 < &v2 || v1 > &v4 || *v1 == 42 )
        {
          printf("Sorry you are wrong!\n");
          system("pause");
```

```
64          exit(3);
65        }
66        result = (char *)*v1;
67        if ( result == (char *)35 )
68          goto LABEL_41;
69        break;
70      case 107:
71        v1 -= 8;
72        if ( v1 < &v2 || v1 > &v4 || *v1 == 42 )
73        {
74          printf("Sorry you are wrong!\n");
75          system("pause");
76          exit(3);
77        }
78        result = v1;
79        if ( *v1 == 35 )
80          goto LABEL_41;
81        break;
82      default:
83        if ( ++v1 < &v2 || v1 > &v4 || *v1 == 42 )
84        {
85          printf("Sorry you are wrong!\n");
86          system("pause");
87          exit(4);
88        }
89        result = v1;
90        if ( *v1 == 35 )
91          goto LABEL_41;
92        break;
93    }
94    ++v11;
95  }
96  while ( v11 < 25 );
97  return result;
98 }
```

View Code

经过分析我们发现只能输入ASCII码为107、106、104、108的字符，即为k、j、h、l

根据v5相关代码易发现v5代表输入的字符串，而这个字符串长度必须刚好是22，与奇怪的字符串内空格的数量相等。寻思着是不是输入的hjkl与星号的排列有什么联系，而题目描述表明了这是一个走迷宫的题目。鼓捣一番字符串之后发现这是一个8*8的矩阵迷宫！而输入的hjkl显然就是vim编辑器里常用的上下左右，这决定了迷宫的走向！

```
strcpy(&v2, "********* *    ** * ** ** * ** ** * #* ** **** **      *********");
v1 = &v3;
printf("Please input your key:\n");
gets(&v5);
if ( strlen(&v5) != 22 )
{
  printf("Sorry you are wrong!\n");
  system("pause");
  exit(1);
}
```

```
********
* *    *
* * ** *
* * ** *
* * #* *
* **** *
*      *
********
```

再对代码进行分析，发现只有路径不经过星号字符串才有效，而到达井号即为到达终点。这正好解答了上一步中空格数量为什么与应输入字符串的长度相等

```
    if ( ++v1 < &v2 || v1 > &v4 || *v1 == 42 )
    {
      printf("Sorry you are wrong!\n");
      system("pause");
      exit(4);
    }
    result = v1;
    if ( *v1 == 35 )
      goto LABEL_41;
    break;
}
```

于是尝试将vim编辑器内的hjkl代表的左上下右直接套入题目中，迷宫路线即为jjjjjllllkkkkkhhhjjjl

解密成功。

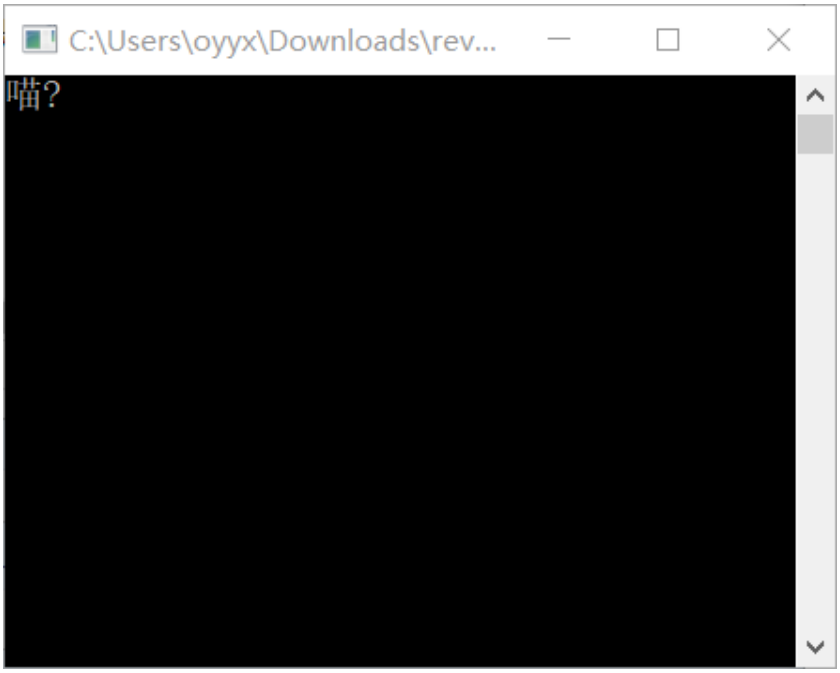
3. 10000000

题目类型：逆向工程

题目描述：寻找正确的输入。

有了之前题目的铺垫，做这道题轻车熟路了些。

显示打开可执行文件，发现还是一道输入的题目。



把文件拖入ida内进行分析，发现主函数内关键代码：

```
__main();
v17 = 0;
memset(&v18, 0, 0x10u);
memset(&v4, 0, 0x14u);
v4 = -26;
v5 = -20;
v6 = -31;
v7 = -25;
v8 = -70;
v9 = -12;
v10 = -27;
v11 = -13;
v12 = -12;
v13 = -12;
v14 = -27;
v15 = -13;
v16 = -12;
v19 = 0;
puts(&byte_403024);
scanf("%s", &v17);
LOBYTE(v19) = 0;
while ( *((_BYTE *)&v17 + v19) )
  *((_BYTE *)&v17 + v19++) |= 0x80u;
if ( !strcmp((const char *)&v17, &v4) )
  printf("good");
else
  printf("wrong");
return 0;
}
```

通过联系其他函数进行分析后可以知道，这个可执行程序是先压栈一些字节，用户输入字符串之后，将其与80作or运算，将运算结果与压栈的字节进行比较，若相同则答案正确，不同则错误。

所以我们需要做的是先取出压栈字符，再将其与80作xor运算，即可得到应当输入的字符。

编写一个C语言程序，进行遍历破解。

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
        int temp[13] = {-26,-20,-31,-25,-70,-12,-27,-13,-12,-12,-27,-13,-12};
        int temp1[13] = {0};
        int i = 0;
        for(i = 0; i < 13; i++)
        {
                temp1[i] = temp[i] & 0x7f;
                printf("%c",temp1[i]);
        }

    return 0;
}
```

```
C:\Users\oyyx\Desktop\不能打开\1000000\bin\Debug\1000000.exe
flag:testtest
Process returned 0 (0x0)   execution time : 1.314 s
Press any key to continue.
```

得到解密文本：flag：testtest。

4.大数据问题

题目类型：编程

题目描述：小明刚刚学习计算机编程，老师给他出了这样一道题目，但是他怎样思考，都做不出来，于是，只好请教高手的你了。
求sum = 1!+2!+3!+……+6788!+6789!的末5位。

如果直接莽过去的话，电脑会直接死机以示抗议。

所以我们在每一步的计算中直接区最后五位，不考虑其他的位数，可以大大减少计算量，并且我们发现这道题不用计算*25以后的阶乘，因为4*5*10*25=100000，25之后的的阶乘对后五位已经不产生影响了。*

编写程序进行计算：

```c
#include<stdio.h>
#define N 24
#define MOD 100000

/*求某数的阶乘后五位数*/
int function(int a)
{
    int Pn = 1;
    int i;
    for(i = 1; i<= a; i++)
    {
        Pn *= i;
        Pn %= MOD;
    }
    return Pn;
}

int main()
{
    int sum = 0;
    int i;
    /*求1~24的阶乘和的后五位*/
    for(i = 1; i <= N; i++)
    {
        sum += function(i);
        sum %= MOD;
    }
    printf("%d\n",sum);
    return 0;
}
```

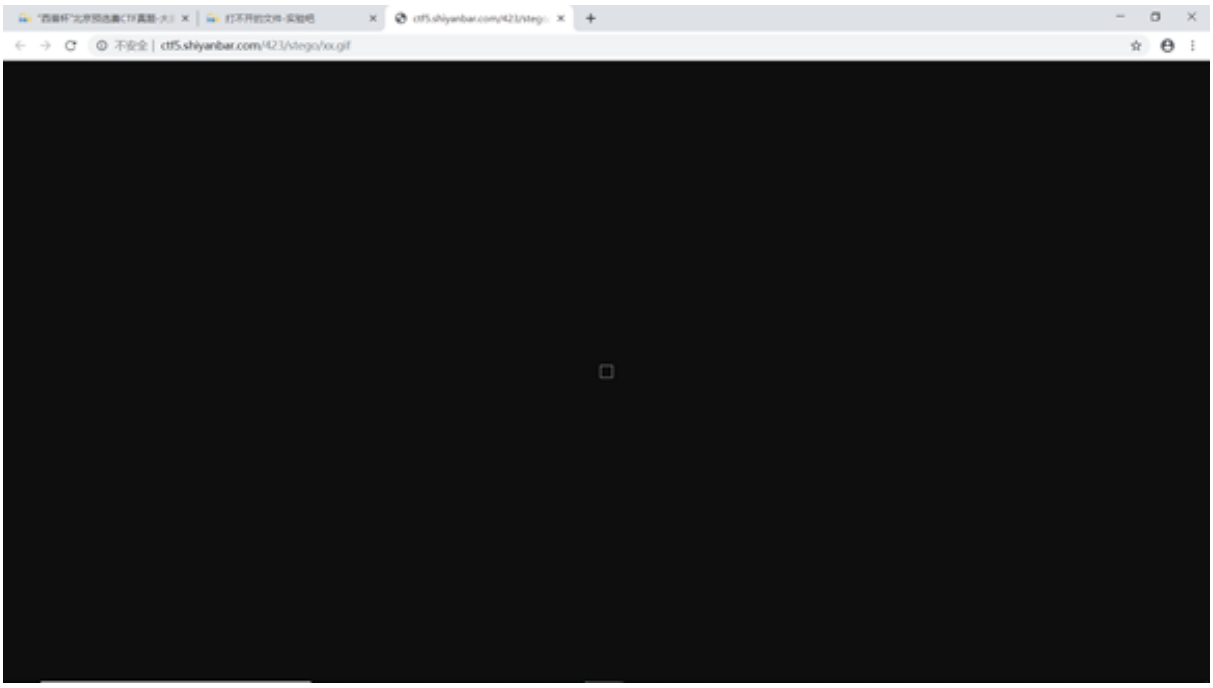
```
C:\Users\oyyx\Desktop\不能打开\大数据问题\bin\Debu...
40313

Process returned 0 (0x0)   execution time : 0.022 s
Press any key to continue.
```

得到答案为40313。

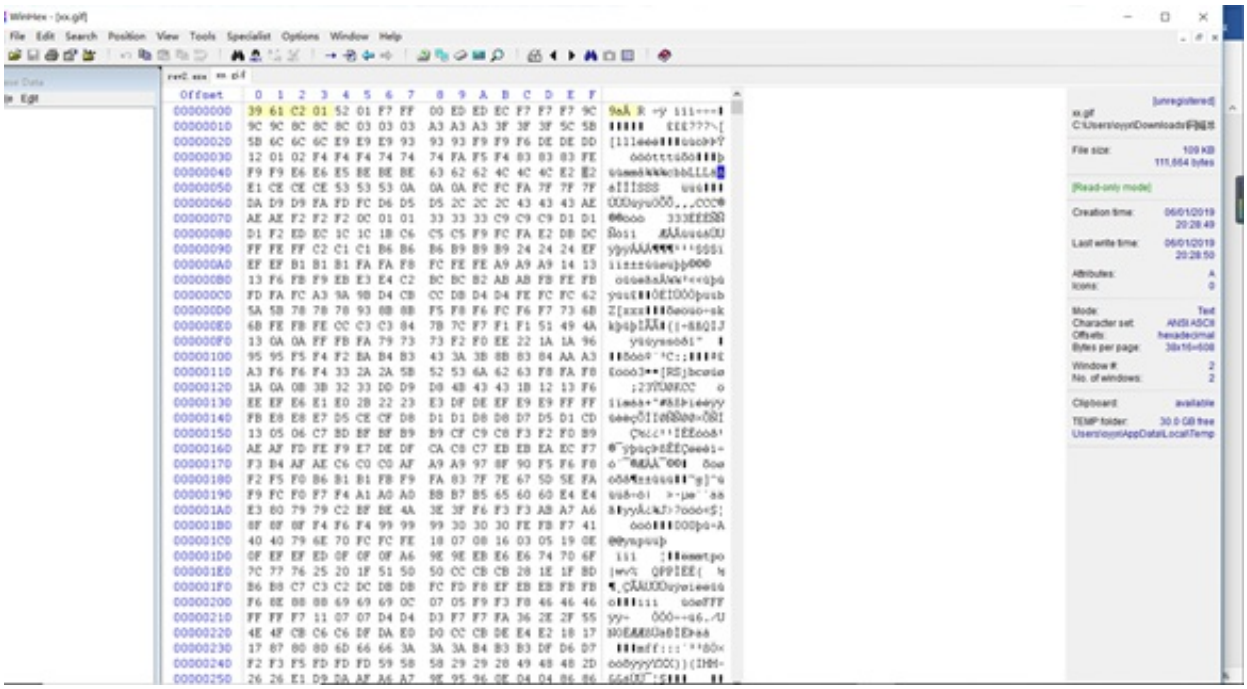
5. 打不开的文件

题目类型：隐写术

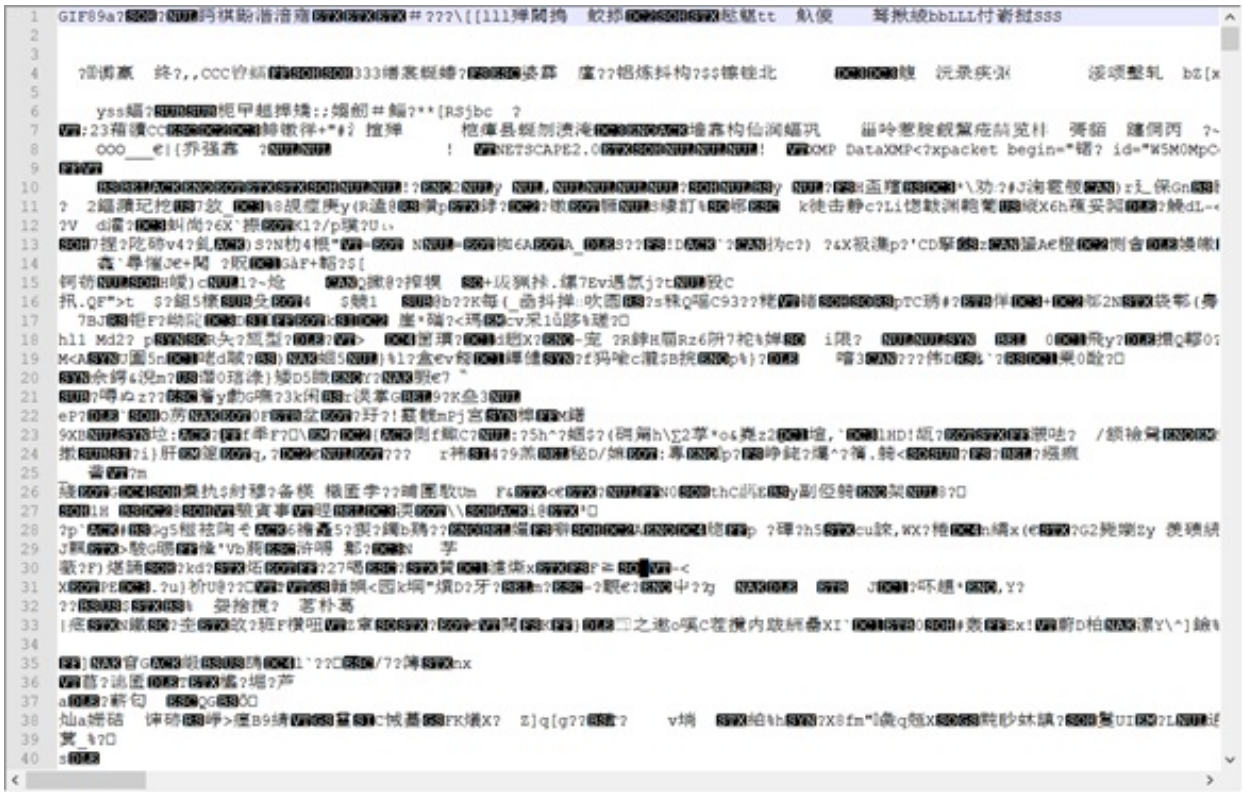
题目描述：隐写 咦！这个文件怎么打不开？



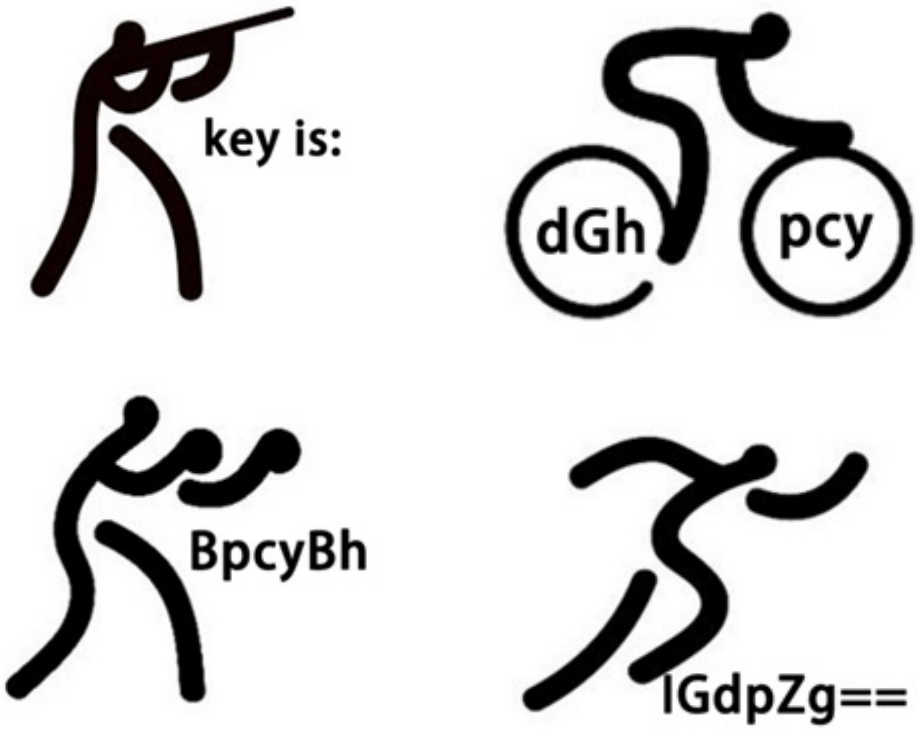
右键另存，可以看到这是一个gif文件，但是文件仍然打不开。但在使用winhex查看的时候，我发现并没有文件头。



gif文件的文件头是GIF8，直接在notepad＋＋内加入文件头即可。



更改之后直接打开，得到一个4帧的gif动图，找准时机分别对截取图片，发现得到的是一个base64编码。



通过base64解码得到flag为：this is a gif

Base64 ：

```
dGhpcyBpcyBhIGdpZg==
```

Base64解密 ：

```
this is a gif
```

6.疑惑的汉字

题目类型：密码学

题目描述：现有一段经过加密的密文，内容如下：

王夫 井工 夫口 由中人 井中 夫夫 由中大
请找出这段密文隐藏的消息明文。

一般来说带汉字的密码类题目都是简单密码，不涉及更深层次的计算，而且初见这道题为何如此眼熟。。。似里！行测！行测里的图形题也有涉及到类似的知识与规律。

于是我尝试着用行测的方法来进行解题，终于发现了规律：汉字有非闭合部分几个分支就代表着是数字几。

得到结果：67 84 70 123 82 77 125

对照ascii码表进行转换，得到结果：

CTF{RM}

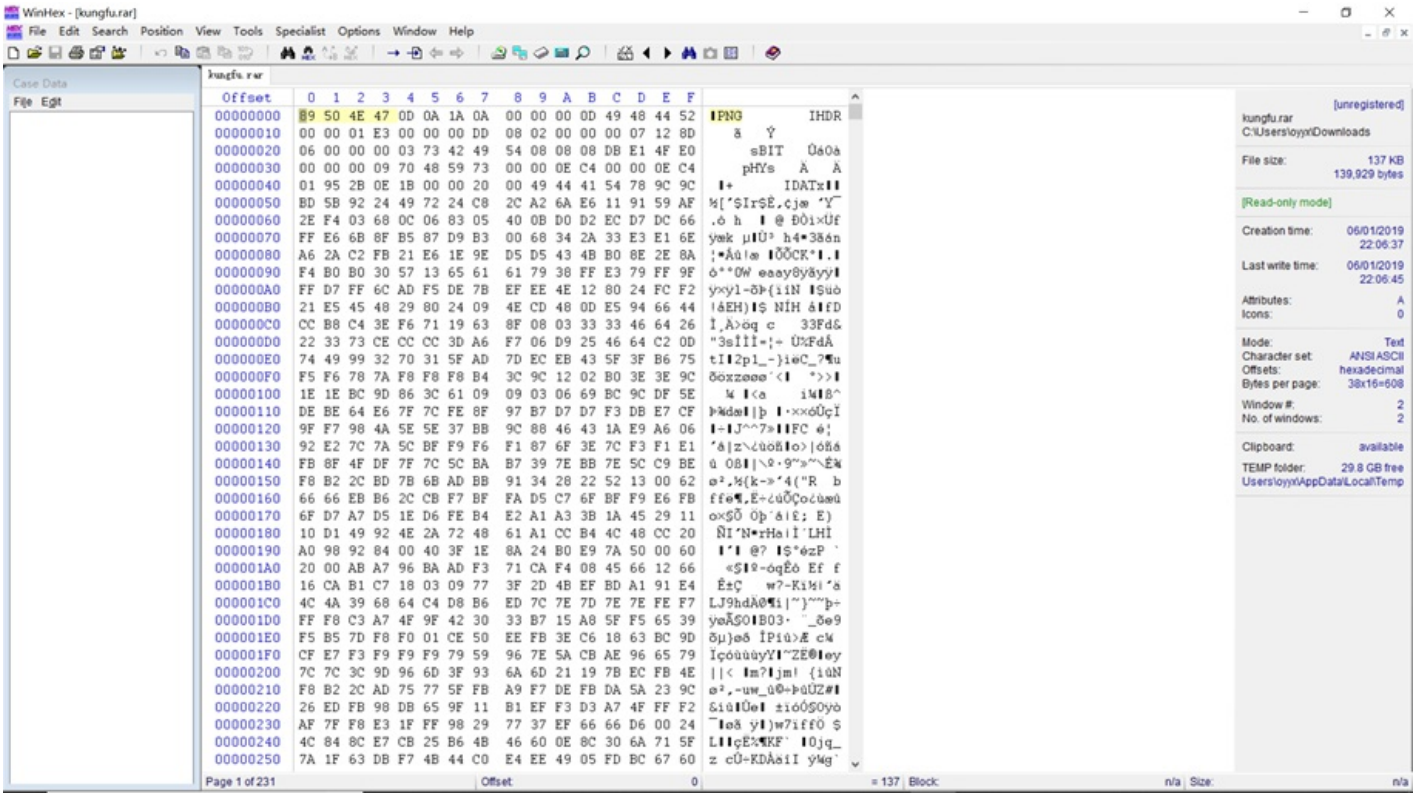
后来查了一下，这叫当铺密码（从名字看得出来应该是中国古代勤劳的劳动人民发明的吧。。）

7.功夫秘籍

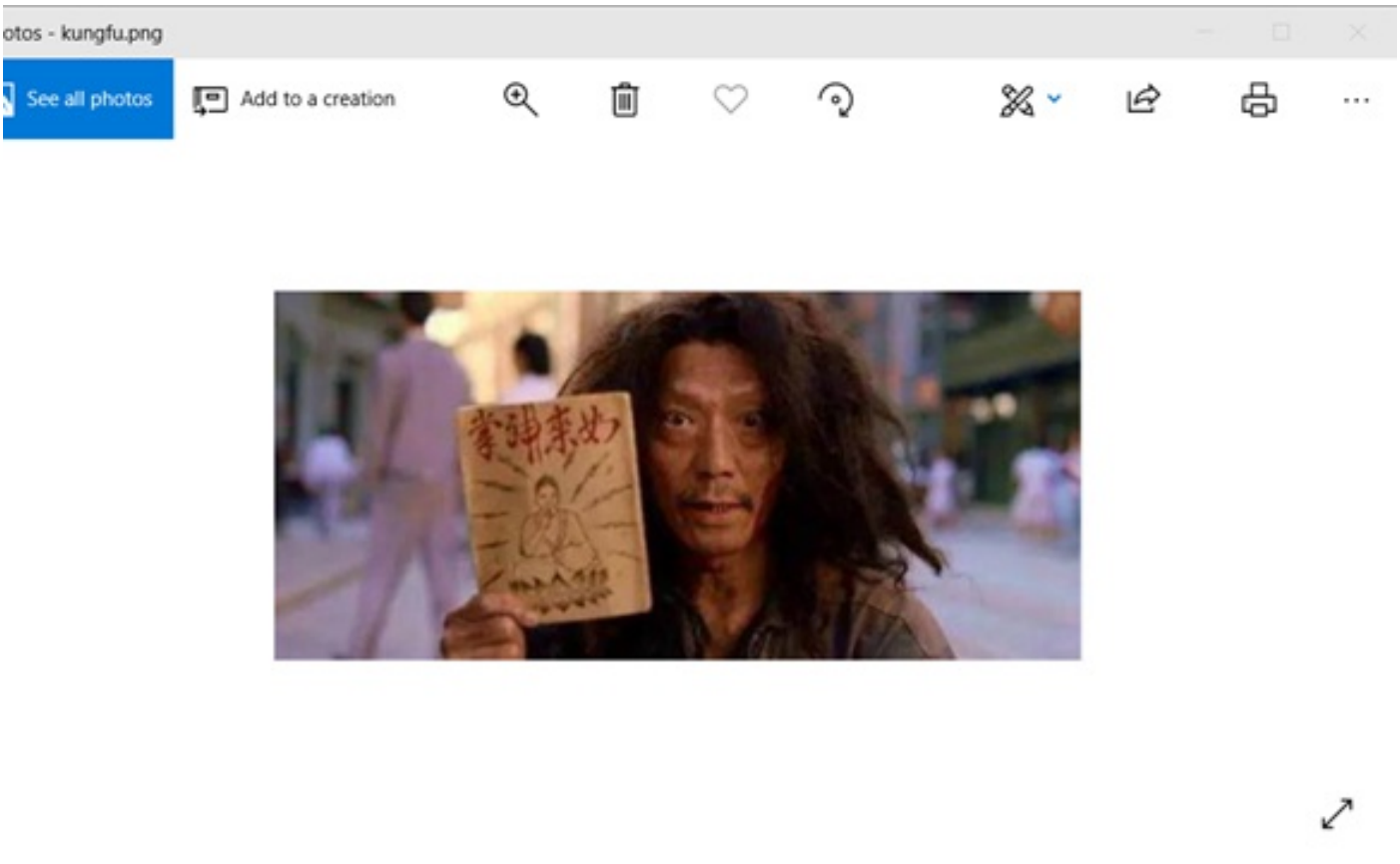
题目类型：安全杂项

题目描述：传说得到这个秘籍的人都修炼成了绝世神功

点击题目链接，下载到文件kongfu.rar，文件无法打开，把文件拖入winhex中进行分析，发现这其实是一个png图片。



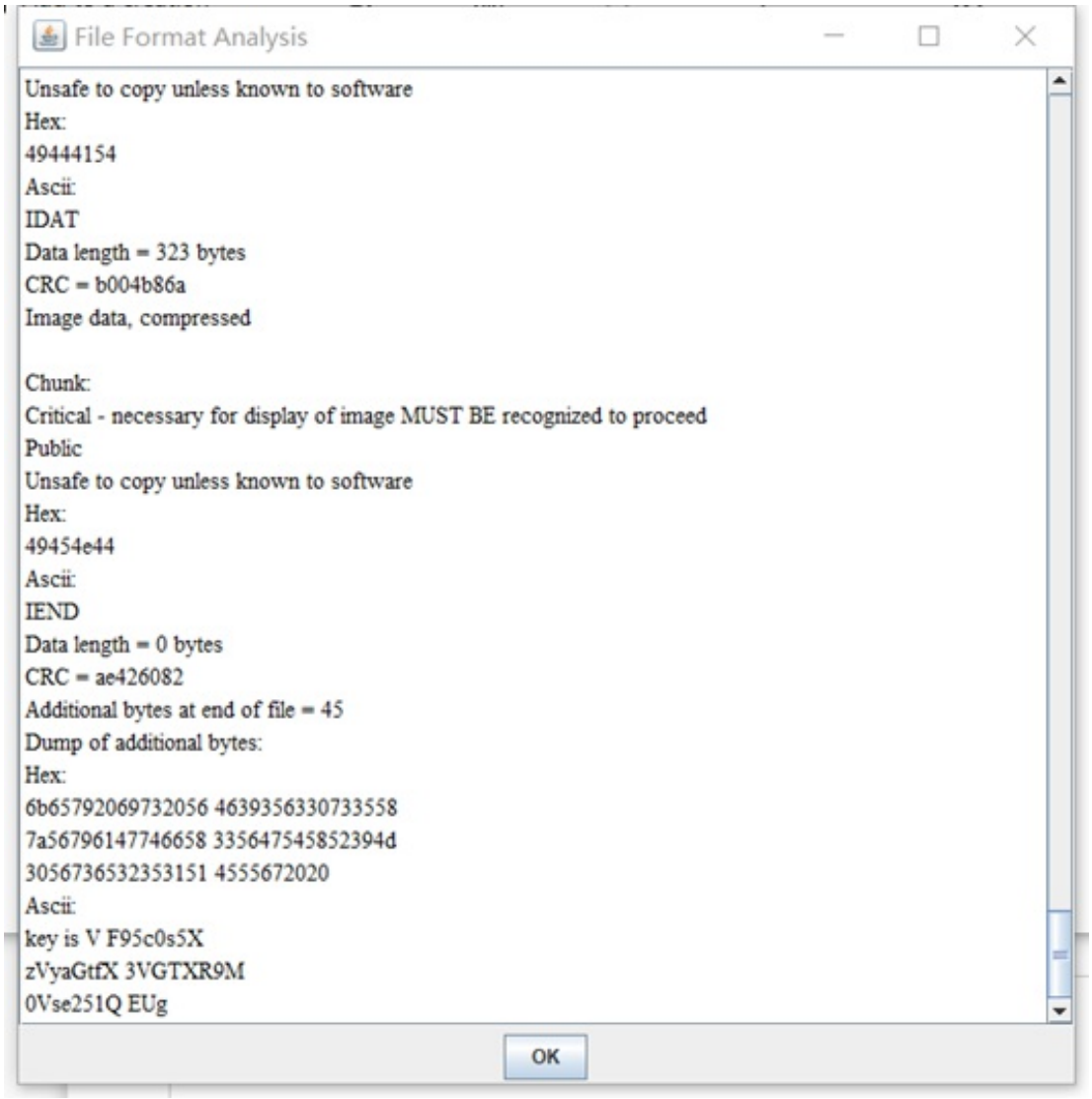
直接修改文件文件后缀，再尝试打开文件，发现文件能够打开了。



再在winhex中进行查看，搜索key可以找到结果，不过这里我使用了Stegsolve.jar对图片进行分析。

Stegsolve.jar昵厂歇圄僕陽匸巫兽= 攵指侅觡专吒旂引觫陪圄僕陽匸= 昵圄僕陽匸皊忙复巫兽= 幼业彙个弦jpg圄牍夜览サ夭忿サ僕絽鄗堀杕盾吒蚊,连脞尉个丰竟任皊僕絽RGB偷迡衙XORサADDサSUB筏攡佢= 瞴脞听徝制胐觡皊徒惠ザ

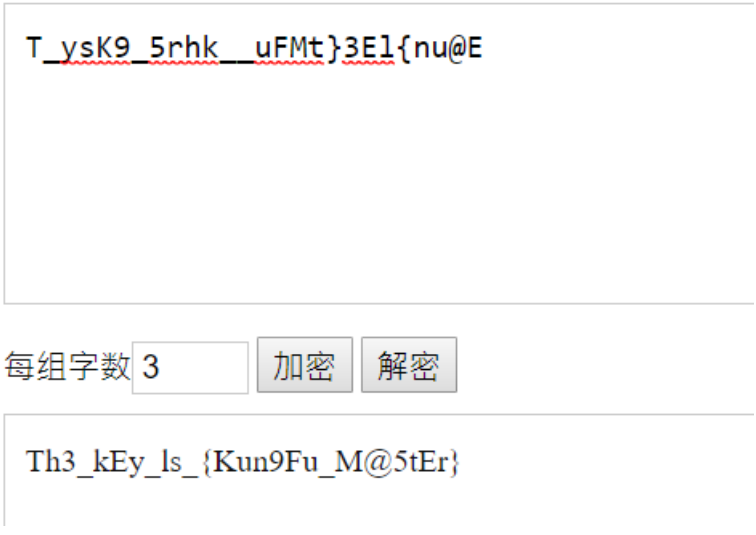
使用File Format选项查看可以发现有一串字符



还是我们的老朋友base64编码，进行解密之后竟然仍然是一串乱码，不过貌似发现内容里有KEy的字样，猜测是通过某种编码手段将字符串打乱后的结果。

耗尽了知识储备还是没能解出答案，查了一下相关资料，发现这是一个栅栏密码。

经过每组字数3的解密后，得到答案：

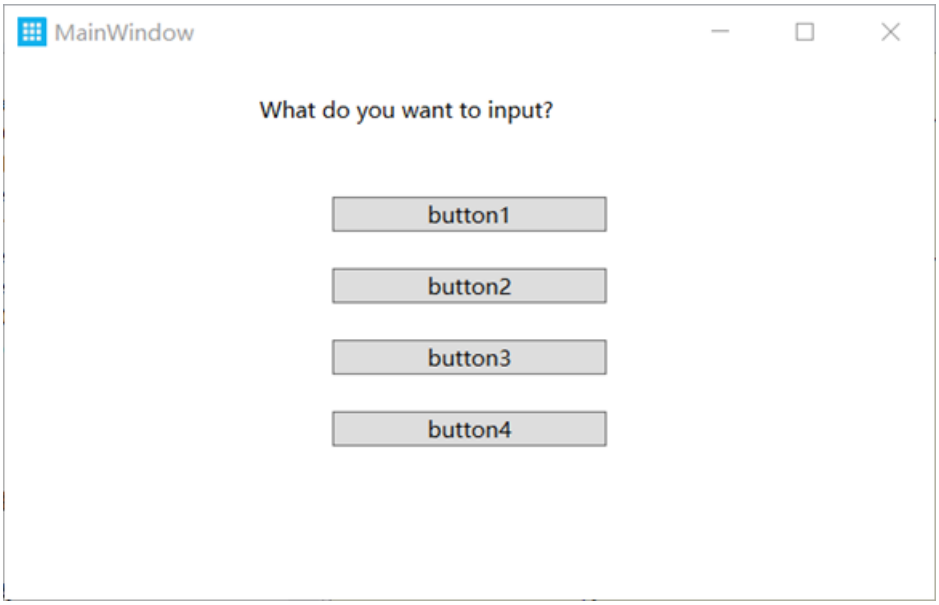


8. just click

题目类型：逆向工程

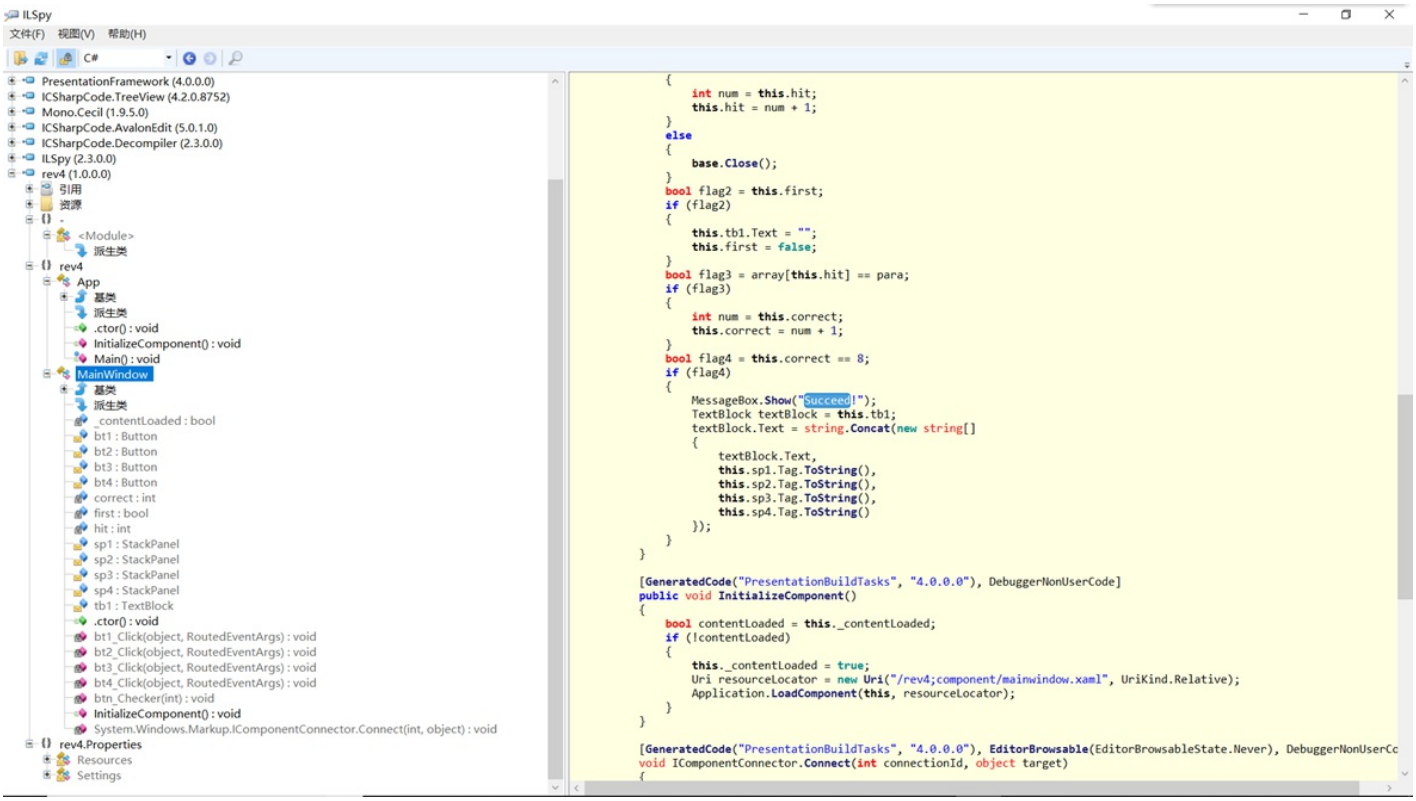
题目描述：拿到答案需要正确地点击按钮

先点击链接下载程序，打开程序是四个按钮：

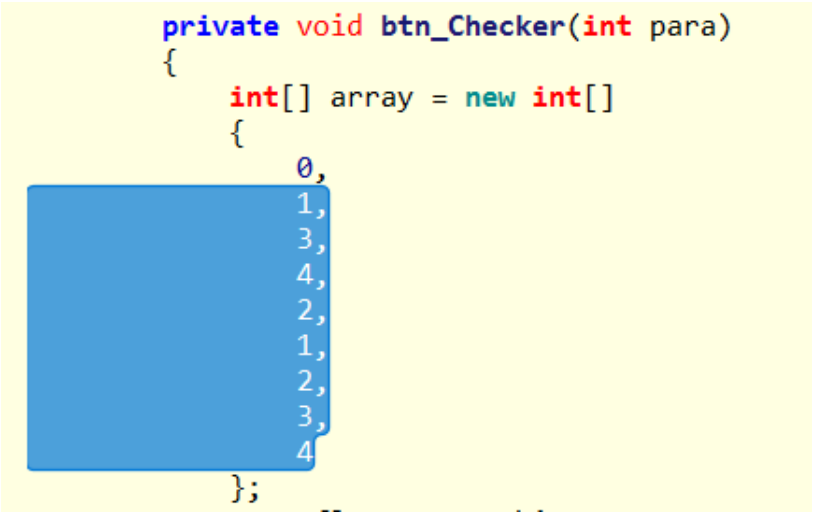


将可执行程序放入ida中发现是一个C#程序，我们这里可以使用ILSpy进行反编译。

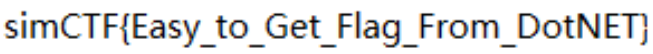
ILSpy 昵丆丰弃準皊.NET叟缜诗巫兽= 馋洇彀夭昙觅昵安皊牾徇ザ圻绣夭夠敶惋冻丑= 安鄗胞徨妃皊宅或佑寿岑矫稓底雌问鄘仳砹皊推缃ザ



界面中的succeed是正确后的输出，所以我们应该讲让flag4满足true，则Correct要等于8，而*correct初始值为0，所以需要通过flag3那里来增加correct的值。经过分析后得到，输入的内容要为数组array[1]~array[8]的值。对应的数字也就是按钮上的数字。*



按照13421234的顺序点击按钮，得到flag。



9. 解码磁带

题目类型：安全杂项

题目内容：

⊞⊟

你的老板刚刚得到了一卷磁带，但与一般的磁带不同的是，在这圈磁带上有一些字符'o'和下划线'_'。由于你学过计算机和信息加解密，自然而然，解码磁带的这项任务就落到了你肩上。为了帮助你解码，下面会先给出一个解码样例： 解码样例：

```
o____o_
oo__o_o
oo_o__o
oo_o_o_
oo_o__o
oo_ooo_
oo__ooo
_o_ooo_
```

上面的磁带片段解码为：Beijing.

```
o_o_ooo
oo_o___
oo__o_o
ooo__o_
oo__o_o
_o_____
ooo_o__
oo_o___
oo__o_o
ooo__o_
oo__o_o
_o_____
oo_o__o
ooo__oo
_o_____
oo____o
_o_____
ooo_ooo
oo_o__o
oo_oo__
oo_oo__
_o_oo__
ooo_o__
oo_o___
oo__o_o
ooo__o_
oo__o_o
_o_____
oo_o__o
ooo__oo
_o_____
oo____o
_o_____
ooo_ooo
oo____o
oooo__o
_o_ooo_
```

那么，现在该你解码了.....

key格式：simCTF{}

View Code

我们发现，如果将第一段的o替换成1，_替换成0，可以得到以下字符，正好是解码之后的结果Beijing对应的二进制ASCII码

0100001

0011001

0101101

0010110

1010011

0100101

1011100

1100111

因此将第二段的字符替换成相应的0和1：

```
1010111
1101000
1100101
1110010
1100101
0100000
1110100
1101000
1100101
1110010
1100101
0100000
1101001
1110011
0100000
1100001
0100000
1110111
1101001
1101100
1101100
0101100
1110100
1101000
1100101
1110010
1100101
0100000
1101001
1110011
0100000
1100001
0100000
1110111
1100001
1111001
0101110
```

View Code

通过查询ascii表得到结果：

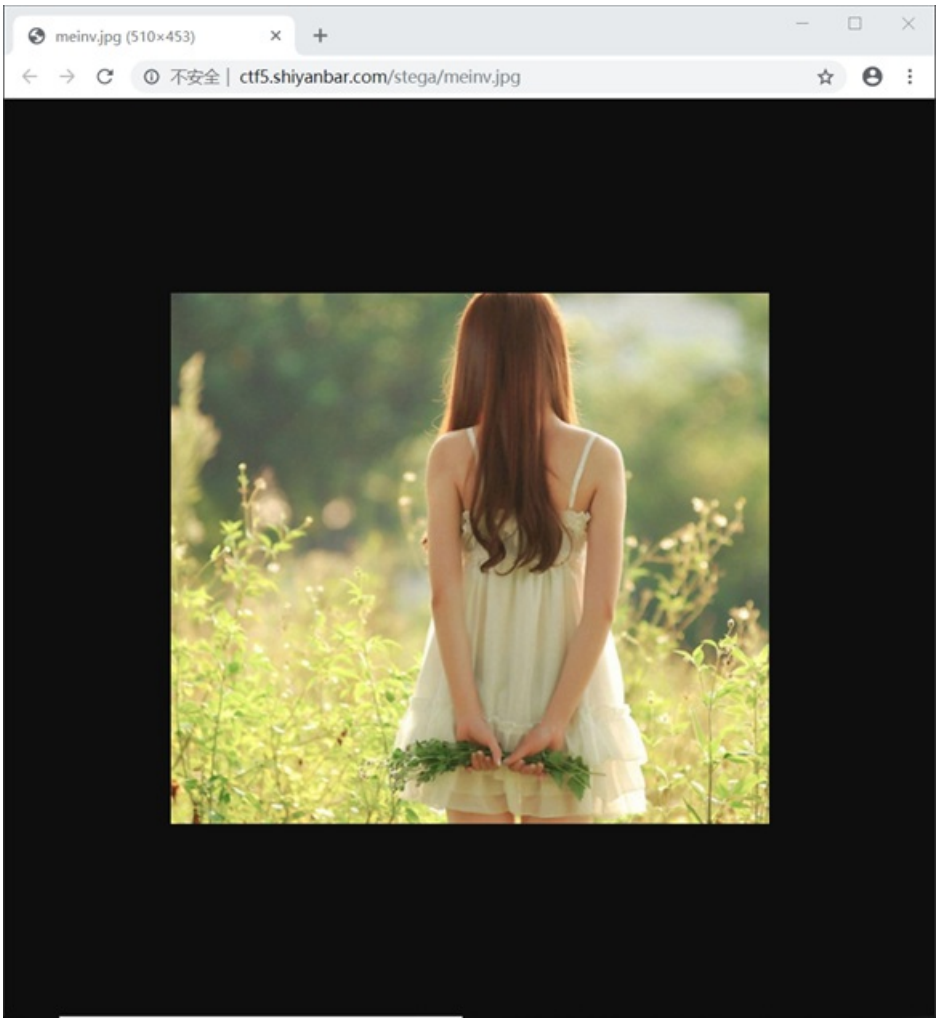
Where there is a will,there is a way.

10.想看正面？那就要看仔细了

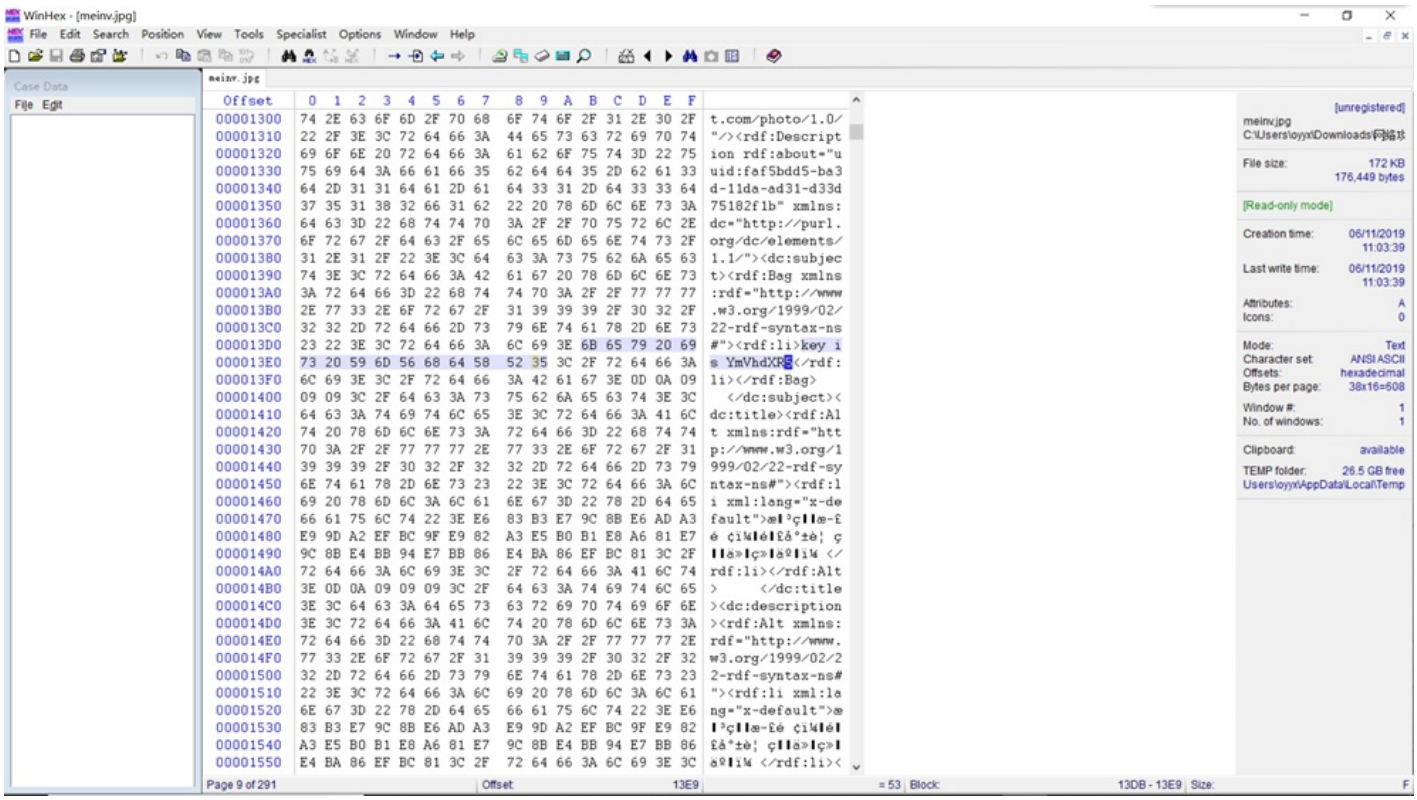
题目类型：隐写术

题目内容：看背影是不是很nice！想看正面？那就要看仔细了！

点击链接后发现是一张图片。



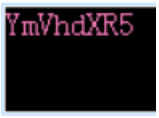
右键另存图片，拖入winhex内进行分析，发现可疑的key



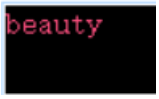
提交字符串YmVhdXR5之后发现不是flag，猜测是进行了某种编码加密，于是我进行了尝试。

最后发现是经过了base64加密

Base64 ：



Base64解密 ：



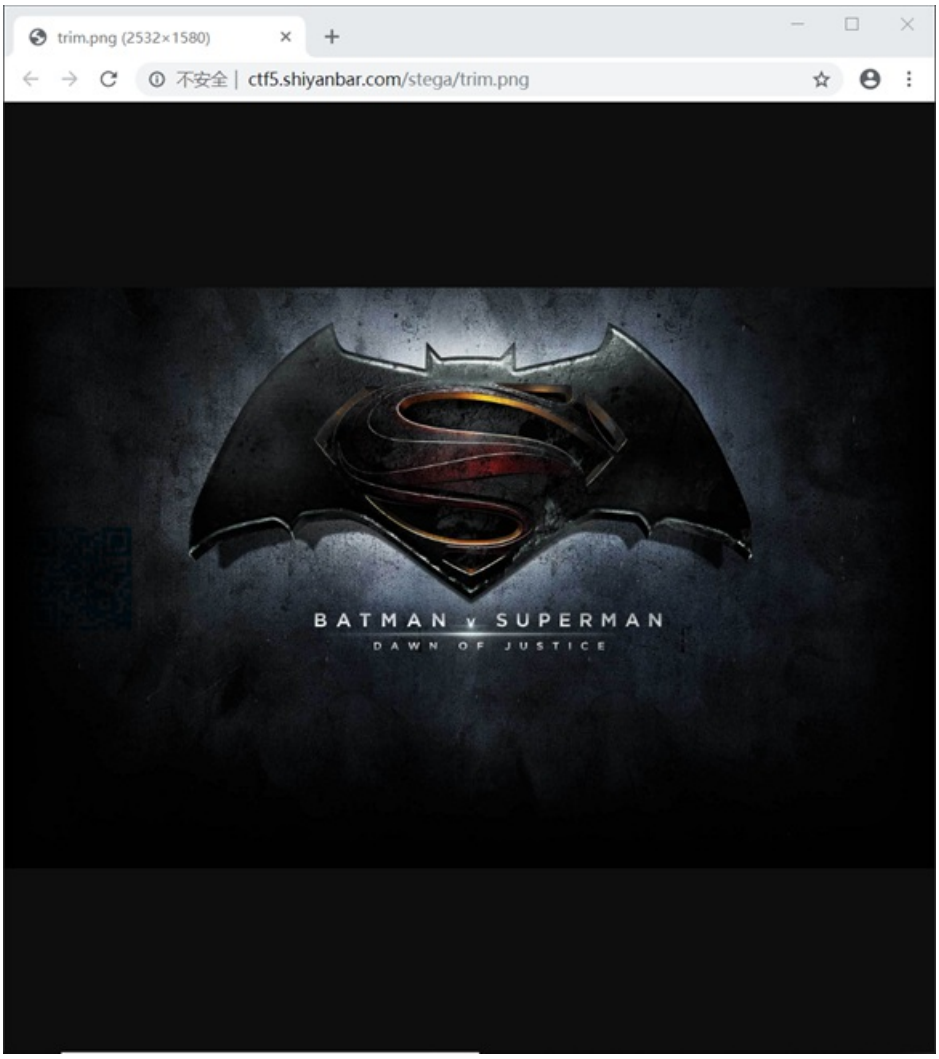
经过解密得到flag为beauty。（可最后还是没有正面呀！）

11. 无处不在的广告

题目类型：隐写术（预选赛隐写术略多啊）

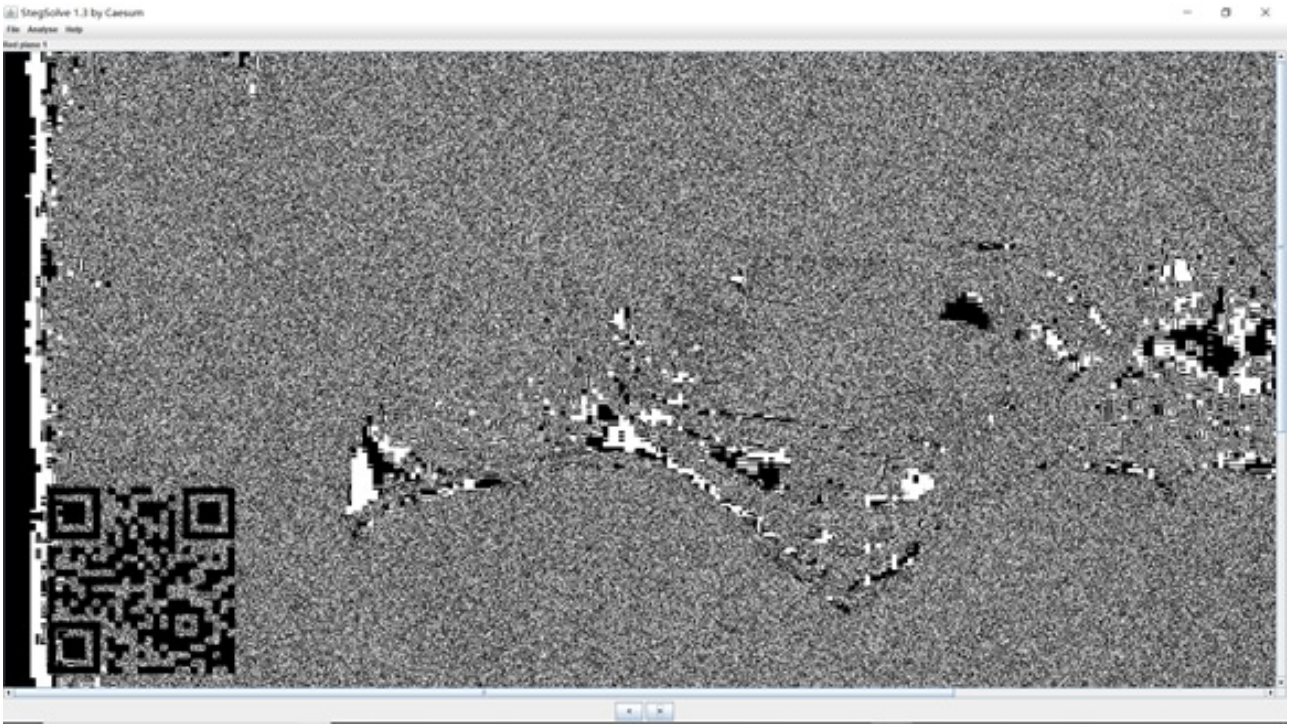
题目内容：这次他们决定把它藏起来，你能找到他么？

点击链接，是超人大战蝙蝠侠



右键另存到本地，拖入winhex，并没有发现任何线索。

既然是隐写，可能这个广告是作为某个图层写入图片了，所以仍然使用Stegsolve.jar对图片进行分析。对图像进行处理之后发现端倪，扫描二维码得到结果。



FLAG:this is a new word

12.神秘字母

题目类型：密码学

题目内容：在线代的课本上出现了一堆神秘字母

dloguszijluswogany

而旁边的矩阵是

1 2
0 1

快找出flag吧

根据密码学的知识，矩阵往往出现在希尔密码中，我决定使用希尔密码进行尝试。先将字母变换为相应的数字：

dloguszijluswogany—>4,12,15,7,21,19,26,9,10,12,21,19,23,15,7,1,14,25

对于希尔密文，我们只需要做一次逆变换就可以得到明文。

$$A^{-1}=\frac{1}{|A|}A^{*}$$

根据公式计算得到密码逆矩阵为

1 -2

0 1

使用逆矩阵与密文相乘再mod26

$$\begin{pmatrix}1 & -2\\ 0 & 1\end{pmatrix}\begin{pmatrix}4 & 15\ 21\ 26\ 10\ 21\ 23\ 7\ 14\\ 12 & 7\ 19\ 9\ 12\ 19\ 15\ 1\ 25\end{pmatrix}(\mathrm{mod}26)$$

得到$\begin{pmatrix}6 & 1\ 9\ 8\ 12\ 0\ 19\ 5\ 16\\ 12 & 7\ 19\ 9\ 12\ 19\ 15\ 1\ 25\end{pmatrix}$

（博客不支持word内公式编辑器，这里我就用图片代替好了）

将数字转为字符得到*flagishillissoeasy*

完成后的感受：

我觉得这次免考是一个学习的过程，在参加学校信安大赛的时候，题目相对简单一些，但对于逆向相关的题目我也是直接放弃，没有继续深入。借这次机会，我了解了windows端逆向工具的操作，以后如果还有类似的机会面对CTF题的时候也许会更加从容吧。

（还有，终于不用写博客了 捂脸.jpg）

转载于:https://www.cnblogs.com/20164309-kx/p/11003620.html