

HGAME2023-writeup-Week1 by crumbling

[Crypto](#)

- [兔兔的车票](#)
- [RSA](#)
- [Be Stream](#)
- [神秘的电话](#)

[Reverse](#)

- [test your IDA](#)
- [easyasm](#)
- [easyenc](#)
- [a cup of tea](#)
- [encode](#)

[Misc](#)

- [Sign In](#)
- [神秘的海报](#)
- [e99p1ant wan](#)

Crypto

兔兔的车票

阅读encrypt.py文件，大致明白代码实现了多张原图片与3张函数随机生成图片的异或。

随机生成的图片显然无法获得，又因为pics文件中一共只有16张图片，数量不多，那就遍历每一个图片与图片的异或，查看结果试试。

以下为在原代码基础上修改而来的解密用代码：

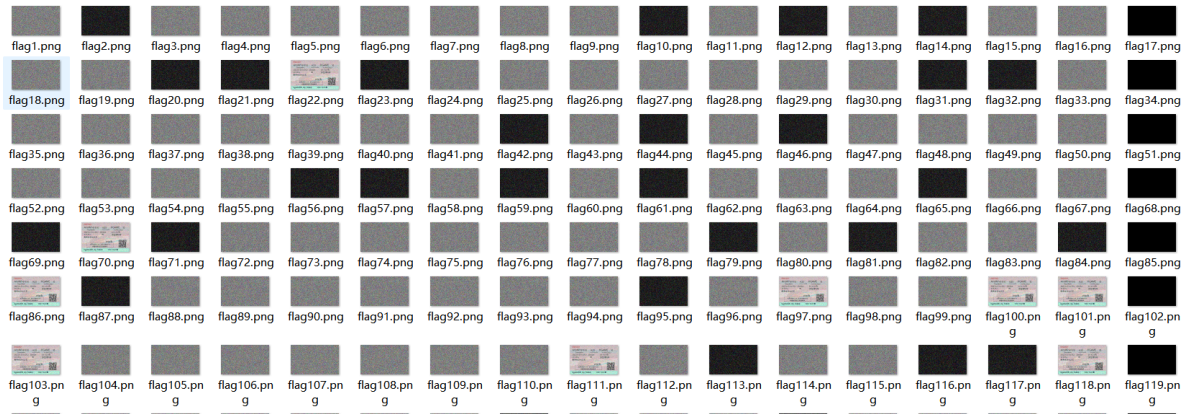
```
from PIL import Image
from Crypto.Util.number import *
from random import shuffle, randint, getrandbits

flagImg = Image.open('pics/enc0.png')
width = flagImg.width
height = flagImg.height

def xorImg(keyImg, sourceImg):
    img = Image.new('RGB', (width, height))
    for i in range(height):
        for j in range(width):
            p1, p2 = keyImg.getpixel((j, i)), sourceImg.getpixel((j, i))
            img.putpixel((j, i), tuple([(p1[k] ^ p2[k]) for k in range(3)]))
    return img
```

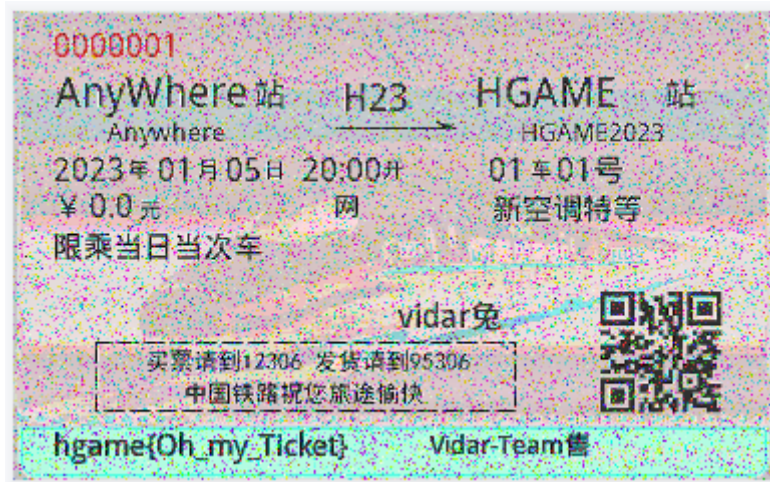
```
e=0
for i in range(16):
    im=Image.open(f"pics/enc{i}.png")
    for j in range(16):
        key = Image.open(f"pics/enc{j}.png")
        flagImg = xorImg(key, im)
        flagImg.save(f'pics/flag{e}.png')
        e += 1
```

在所有结果图片中确实有车票的图片，以下为文件中部分截图：



不太明白为什么两个enc图片的异或能够得到flag图片，也许是有特殊的flag图片吧。

总之点开图片就得到了flag



RSA

很正常的RSA，用factordb.com分解n得到pq后求解即可，以下为解密代码：

```

from Crypto.Util.number import long_to_bytes
from gmpy2 import *
n =
gmpy2.mpz(1351271383482997573741964470626408584169203500983200999931159497190513
54213545596643216739555453946196078110834726375475981791223069451364024181952818
05680208956706492651029412459417447812321651660036833476384920694294282471153133
4239106807454086389211139153023662266125937481669520771879355089997671125020789)
c =
gmpy2.mpz(1106747926740177482432323511858960196604347183420016869065277898762649
76328686134101972125493938434992787002915562500475480693297360867681000092725583
28461635354342238848920811454500713860654367804079865183602743338328217708103415
1589935024292017207209056829250152219183518400364871109559825679273502274955582)
e =65537
p=112391349878049935867635590281872450576525502195152017686447707338690881853207
40938450178816138394844329723311433549899499795775655921261664087997097294813
q=120229126614209415925697517318026393750884274634301622521130826196178370109130
02515450223656942836378041122163833359097910935638423464006252814266959128953
r= (p - 1) * (q - 1)
d = gmpy2.invert(e, r)
flag=long_to_bytes(pow(c,d,n))
print(flag)

```

运行程序得到flag: hgame{factordb.com_is_strong!}

Be Stream

阅读代码，明白代码实现了生成water，后进行异或加密。

根据题目已知，只要运行就能获得water的数据完成解密，然而递归和i增大后的大数都影响了结果的获得，于是对原代码进行加工。

先附上解密代码：

```

enc=b'\x1a\x15\x05\t\x17\t\xf5\xa2-\x06\xec\xed\x01-
\xc7\xcc2\x1exA\x1c\x157[\x06\x13/!-\x0b\xd4\x91-\x06\x8b\xd4-\x1e+*\x15-
pm\x1f\x17\x1bY'
key=[4784265876259235186, 2019423192753765707364]
key=[114,100]#分别为key[0]%256,key[1]%256
def stream(a,i):
    key2=0
    if i==0:
        return key[0]
    elif i==1:
        return key[1]
    else:
        for j in range(a,i,1):
            key2=(7*key[0]+4*key[1])%256
            key[0],key[1]=key[1],key2
        return key2

k=0
flag = b""
for i in range(len(enc)):
    if i%2==0:

```

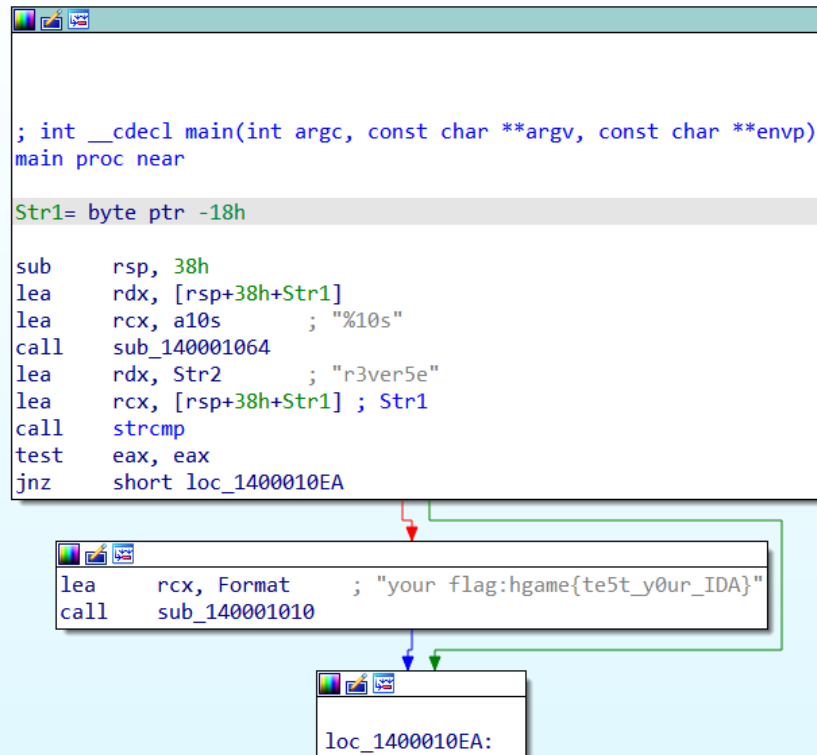

4.因为没有成功将“北欧神话”翻译成密钥vidar，最后通过[Vigenere Solver |_guballa.de](https://guballa.de/Vigenere-Solver/)爆破获得了最终结果

flag: hagme{welcome_to_hgame2023_and_enjoy_hacking}

Reverse

test your IDA

用ida打开，等待分析，可以直接看到flag



easyasm

读懂汇编语言（挠头

主要加密算法为异或。

```
.text:00401190          xor     eax, 33h
```

以下为python解密代码：

```
from Crypto.Util.number import long_to_bytes
a=
[0x5b,0x54,0x52,0x5e,0x56,0x48,0x44,0x56,0x5f,0x50,0x3,0x5e,0x56,0x6c,0x47,0x3,0
x6c,0x41,0x56,0x6c,0x44,0x5c,0x41,0x2,0x57,0x12,0x4e]
b=b''
for i in range(len(a)):
    b+=long_to_bytes(a[i]^0x33)
print(b)
```

运行得到flag: hgame{welc0me_t0_re_wor1d!}

easyenc

放进ida反编译，发现加密算法主要是异或。

```
v7[9] = 101123568;
v8 = -7;
scanf("%50s", v9);
v4 = -1i64;
do
    ++v4;
while ( *(v9 + v4) );
if ( v4 == 41 ) // 输入刚好有41个字符
{
    while ( 1 )
    {
        v5 = *(v9 + v3) ^ 0x32 - 86;
        *(v9 + v3) = v5;
        if ( *(v7 + v3) != v5 )
            break;
        if ( ++v3 >= 41 )
        {
            printf("you are right!");
            return 0;
        }
    }
    printf("wrong!");
}
return 0;
}
```

提取v7的内容后写了个解密代码

```
#include<stdio.h>
unsigned char v7[] =
{
    0xC7, 0x45, 0xE7, 0x04, 0xFF, 0xFD, 0x09, 0x33, 0xC0, 0xC7,
    0x45, 0xEB, 0x01, 0xF3, 0xB0, 0x00, 0xC7, 0x45, 0xEF, 0x00,
    0x05, 0xF0, 0xAD, 0xC7, 0x45, 0xF3, 0x07, 0x06, 0x17, 0x05,
    0x0F, 0x11, 0x45, 0x27, 0xC7, 0x45, 0xF7, 0xEB, 0x17, 0xFD,
    0x17, 0x0F, 0x11, 0x45, 0x37, 0xC7, 0x45, 0xFB, 0xEA, 0x01,
    0xEE, 0x01, 0xC7, 0x45, 0xFF, 0xEA, 0xB1, 0x05, 0xFA, 0xC7,
    0x45, 0x03, 0x08, 0x01, 0x17, 0xAC, 0xC7, 0x45, 0x07, 0xEC,
    0x01, 0xEA, 0xFD, 0xC7, 0x45, 0x0B, 0xF0, 0x05, 0x07, 0x06
};
int main()
{
    char v10[100];
    int v4=-1,i;
    char v5;
    for (i=0;i<=81;i++)
    {
        v10[i]=(v7[i]+0x56)^0x32;
        printf("%c",v10[i]);
    }
    return 0;
}
```

因为数据提取得有些丑陋，所以输出来个怪东西

```
/hgam?/: e{4d/ ) dit1/ ) on_iWUm/?s_a_WU/ \ reve/ > r5ib/""le_0/ . pera/kgtiondd
-----
Process exited after 0.04483 seconds with return value 0
请按任意键继续. . . |
```

总之手动把怪东西去掉之后就得到了flag: hgame{4ddit1on_is_a_rever5ible_0peration}

a_cup_of_tea

没有改动的tea算法。以下为解密代码：

```
#include<stdio.h>
unsigned int Buf2[8] = { 0x2E63829D, 0xC14E400F, 0x9B39BFB9, 0x5A1F8B14, 0x61886DDE, 0x6565C6CF, 0x9F064F64, 0x236A43F6 };
unsigned int key[] = {0x45678901, 0x34567890, 0x23456789, 0x12345678};

int main()
{
    unsigned int a1[8] = { 0 };
    unsigned int v2, v3, v4, v5, v6;
    unsigned int i, j, v7, v9;
    for (j = 0; j < 4; j++)
    {
        v7 = Buf2[2 * j];
        v9 = Buf2[2 * j + 1];
        v3 = -0x54321000 * 32;
        v2 = key[3];
        v4 = key[2];
        v5 = key[1];
        v6 = key[0];
        for (i = 0; i < 32; i++)
        {
            v9 -= (v3 + v7) ^ (v5 + 16 * v7) ^ (v6 + (v7 >> 5));
            v7 -= (v3 + v9) ^ (v2 + 16 * v9) ^ (v4 + (v9 >> 5));
            v3 += 0x54321000;
        }
        a1[2 * j] = v7;
        a1[2 * j + 1] = v9;
    }
    printf("%s\n", a1);
    return 0;
}
```

(1. 变量类型影响结果，均为unsigned int

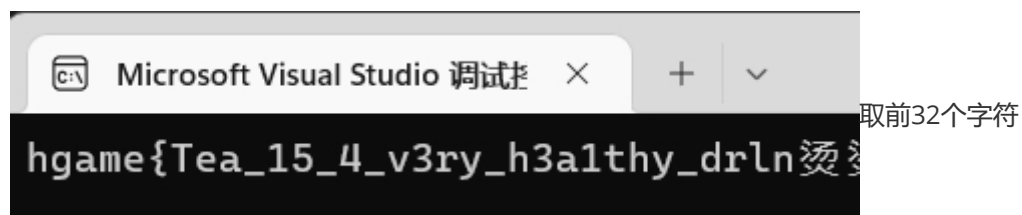
(2. Key的获得：

```
key_ = _mm_load_si128((const __m128i *)&key);
```

(据说_mm_load_si128会以相反的顺序加载数据，于是尝试更改了一下key[]的顺序)

```
.rdata:00007FF7543B22B0 key xmmword 45678901345678902345678912345678h
```

以下为运行结果：



取前32个字符

回看反编译后的memcmp函数可知，共比较0x22也就是34个字节

手动补上k得到flag：hgame{Tea_15_4_v3ry_h3a1thy_drlnk}

encode

用32位ida打开，反编译后简单调整。

程序实现了v4偶数位（从0开始）存放v5某字节低4位二进制对应的10进制数，奇数位对应高4位，后比较v4与dword_403000所存数据是否一致。


```

IDA View-A  Pseudocode-A  Stack of _main  Hex View-1  Struc
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     int v4[100]; // [esp+0h] [ebp-1CCh] BYREF
4     char v5[52]; // [esp+190h] [ebp-3Ch] BYREF
5     int j; // [esp+1C4h] [ebp-8h]
6     int i; // [esp+1C8h] [ebp-4h]
7
8     memset(v5, 0, 0x32u);
9     memset(v4, 0, sizeof(v4));
10    scanf("%50s", v5);
11    for ( i = 0; i < 50; ++i )
12    {
13        v4[2 * i] = v5[i] & 0xF; // 取低4位 按位赋值
14        v4[2 * i + 1] = (v5[i] >> 4) & 0xF; // 取高4位
15    }
16    for ( j = 0; j < 100; ++j )
17    {
18        if ( v4[j] != dword_403000[j] )
19        {
20            printf("Wrong! You are not good at encode");
21            return 0;
22        }
23    }
24    printf("Yes! You are right!");
25    return 0;
26 }

```

解题：获取dword_403000数据，编写解密程序。

```

1 #include<stdio.h>
2 unsigned char ida_chars[] =
3 {
4     0x08, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x07, 0x00,
5     0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00,
6     0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x00,
7     0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00,
8     0x08, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x05, 0x00,
9     0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x00,
10    0x06, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x06, 0x00,
11    0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00,
12    0x04, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x05, 0x00,
13    0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00,
14    0x05, 0x00, 0x00, 0x00, 0x09, 0x00, 0x00, 0x00, 0x06, 0x00,
15    0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00,
16    0x0F, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x05, 0x00,
17    0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00,
18    0x06, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x07, 0x00,
19    0x00, 0x00, 0x09, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00,
20    0x0F, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x06, 0x00,
21    0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00,
22    0x06, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x07, 0x00,
23    0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00,
24    0x01, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x0F, 0x00,
25    0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00,
26    0x07, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x06, 0x00,
27    0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00,
28    0x05, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x02, 0x00,
29    0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00,
30    0x07, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x06, 0x00,
31    0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00,
32    0x05, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x0E, 0x00,
33    0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00,
34    0x06, 0x00, 0x00, 0x00, 0x09, 0x00, 0x00, 0x00, 0x06, 0x00,

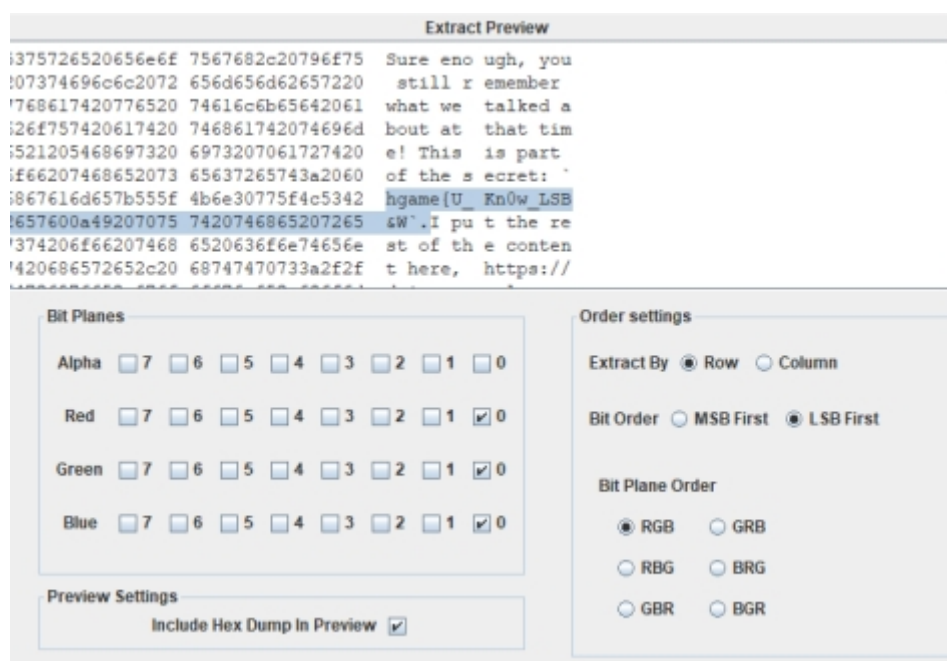
```


Misc

base64在线解密得到flag: hgame{Welcome_To_HGAME2023!}

根据题目表示

尝试了用stegsolve解密LSB 尝试后出现了:



得到一半flag: hgame{U_Kn0w_LSB&W

并根据提示得到通过steghide加密的音频文件，且得知需要6位的密码

根据<https://www.cnblogs.com/pcat/p/5503237.html>提供的脚本爆破得到密码123456

```
from subprocess import *

def foo():
    stegoFile='Bossanova.wav'
    extractFile='passport.txt'
    passFile='english.dic'

    errors=['could not extract','steghide --help','Syntax error']
    cmdFormat='steghide extract -sf "%s" -xf "%s" -p "%s"'
    f=open(passFile,'r')

    for line in f.readlines():
        cmd=cmdFormat %(stegoFile,extractFile,line.strip())
        p=Popen(cmd,shell=True,stdout=PIPE,stderr=STDOUT)
        content=str(p.stdout.read(),'gbk')
        for err in errors:
            if err in content:
                break
        else:
            print(content)
            print('the passphrase is %s' %(line.strip()))
            f.close()
            return

if __name__ == '__main__':
    foo()
    print('ok')
    pass
```

打开cmd 来到steghide.exe所在文件夹，输入steghide extract -sf Bossanova.wav -p 123456

获得flag2.txt文件。

至此获得全部flag: hgame{U_Kn0w_LSB&Wav^Mp3_Stego}

e99p1ant_wan

根据题目表述，图片crc检验错误。

尝试爆破宽高（写训练平台misc的时候用过的代码，出处忘记了。

```
import binascii
import struct

crcbp = open("e99p1ant_want_girlfriend.png", "rb").read()
crc32frombp = int(crcbp[29:33].hex(), 16)
print(crc32frombp)

for i in range(10000):
    for j in range(10000):
        data = crcbp[12:16] + \
            struct.pack('>i', i) + struct.pack('>i', j) + crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        # print(crc32)
        if (crc32 == crc32frombp):
            print(i, j)
            print('hex:', hex(i), hex(j))
            exit(0)
```

运行得到结果，宽高为：0x200 0x2c2。

用010editor更改错误值。

e99p1ant_want_girlfriend.png^x													
	0	1	2	3	4	5	6	7	8	9	A	B	C
h:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49
h:	00	00	02	00	00	00	02	C2	08	06	00	00	00
h:	45	00	00	00	09	70	48	59	73	00	00	0B	13
h:	13	01	00	9A	9C	18	00	00	0A	4D	69	43	43
h:	6F	74	6F	73	68	6F	70	20	49	43	43	20	70
h:	69	6C	65	00	00	78	DA	9D	53	77	58	93	F7

获得图片



提取文字 得到flag