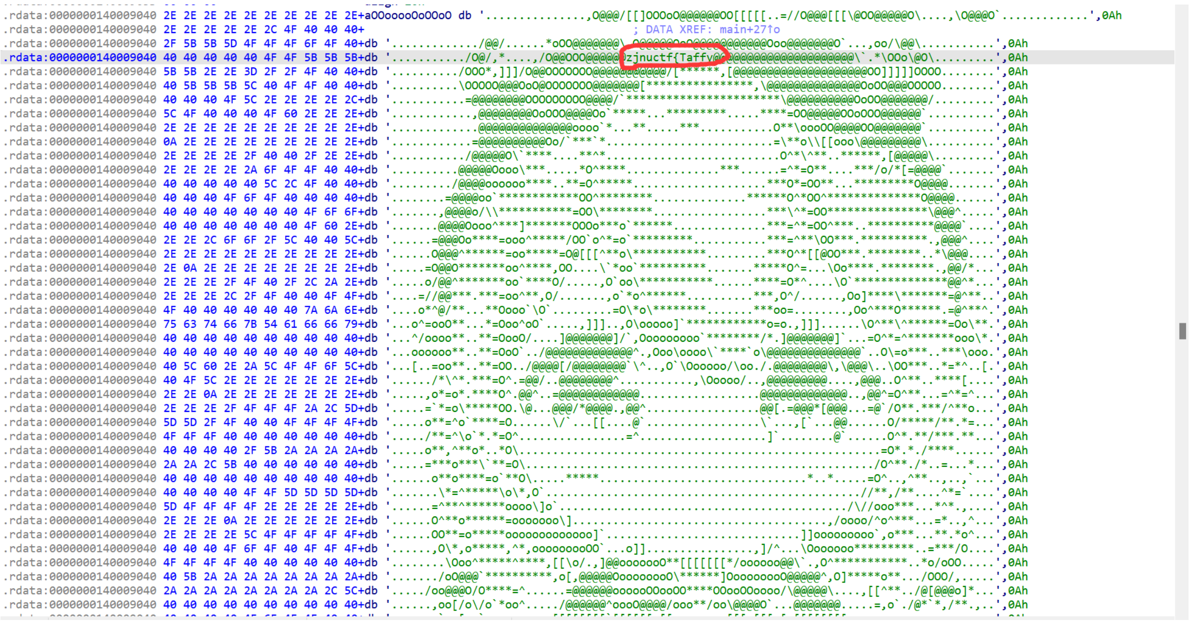
re

题目名称	题目难度	题目分值	知识点	出题人
Taffy_QAQ	简单	200	shift+F12	straw
mikumiku我们稀饭你	中等	300	base换表+异或	straw
派森茶	中等	300	python+标准tea	straw
老爹的武器	困难	400	blowfish加密	straw
Aboutbear	悬赏	500	安卓+frida脱壳+多层加密	ning

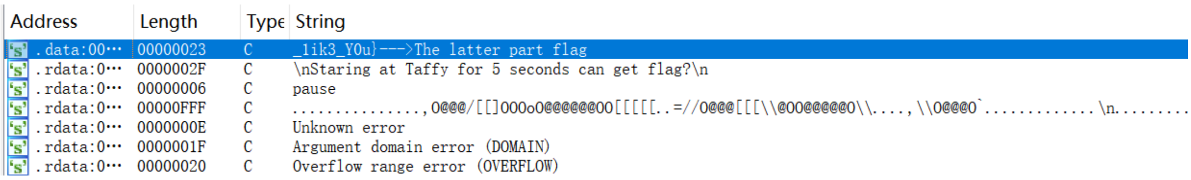
Taffy_QAQ

签到题，一段flag在这 zjnuctf{Taffy



```
.rdata:0000000140009040 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E+aOOooooOoOOoO db '...............,O@@@/[[]OOOoO@@@@@@@OO[[[[[..=//O@@@[[[\@OO@@@@@@O\....,\O@@@O`.............',0Ah
.rdata:0000000140009040 2E 2E 2E 2E 2E 2C 4F 40 40 40+                                     ; DATA XREF: main+27↑o
.rdata:0000000140009040 2F 5B 5B 5D 4F 4F 4F 6F 4F 40+db '............./@@/......*oOO@@@@@@@\.O@@@@@@OoO@@@@@@@@@@@@Ooo@@@@@@@O`...,oo/\@@\...........',0Ah
.rdata:0000000140009040 40 40 40 40 40 4F 4F 5B 5B 5B+db '.........../O@/,*....,/O@@OOO@@@@@zjnuctf{Taffy@@@@@@@@@@@@@@@@@@@@@@@@\`.*\OOo\@O\.........',0Ah
.rdata:0000000140009040 5B 5B 2E 2E 3D 2F 2F 4F 40 40+db '........../OOO*,]]]/O@@OOOOOOO@@@@@@@@@@@/[******,[@@@@@@@@@@@@@@@@@@@@@OO]]]]]OOOO........',0Ah
.rdata:0000000140009040 40 5B 5B 5B 5C 40 4F 4F 40 40+db '..........\OOOOO@@@OoO@OOOOOOO@@@@@@@[***************,\@@@@@@@@@@@@@@OoOO@@@OOOOO........',0Ah
.rdata:0000000140009040 40 40 40 4F 5C 2E 2E 2E 2E 2C+db '...........=@@@@@@@OOOOOOOOO@@@@/`**********************\@@@@@@@@@@OoOO@@@@@@@/.........',0Ah
.rdata:0000000140009040 5C 4F 40 40 40 4F 60 2E 2E 2E+db '............,@@@@@@@@OoOOO@@@@Oo`*****...*********.....****=OO@@@@@OOoOOO@@@@@@`..........',0Ah
.rdata:0000000140009040 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E+db '.............@@@@@@@@@@@@@@oooo`*...**.....***...........O**\oooOO@@@@OO@@@@@@@`..........',0Ah
.rdata:0000000140009040 0A 2E 2E 2E 2E 2E 2E 2E 2E 2E+db '............=@@@@@@@@@@Oo/`***`*...........................=\**o\\[[ooo\@@@@@@@@@\..........',0Ah
.rdata:0000000140009040 2E 2E 2E 2E 2F 40 40 2F 2E 2E+db '............/@@@@@O\`****....**^*..........................O^*\^**..******,[@@@@@\..........',0Ah
.rdata:0000000140009040 2E 2E 2E 2E 2A 6F 4F 4F 40 40+db '..........@@@@@Oooo\***.....*O^****..............***......=^*=O**....***/o/*[=@@@@`........',0Ah
.rdata:0000000140009040 40 40 40 40 40 5C 2C 4F 40 40+db '........./@@@@oooooo****..**=O^*****.......................***O*=OO**...*********O@@@@.......',0Ah
.rdata:0000000140009040 40 40 40 4F 6F 4F 40 40 40 40+db '........=@@@@oo`***********OO^*********...............******O^*OO^**************O@@@@......',0Ah
.rdata:0000000140009040 40 40 40 40 40 40 40 4F 6F 6F+db '.......,@@@@o/\\***********=OO\********.................***\^*=OO**************\@@@^.....',0Ah
.rdata:0000000140009040 40 40 40 40 40 40 40 4F 60 2E+db '.......@@@@Oooo^***]*******OOOo***o`******...............***=^*=OO^***..*********@@@@`....',0Ah
.rdata:0000000140009040 2E 2E 2C 6F 6F 2F 5C 40 40 5C+db '......=@@@Oo****=ooo^*****/OO`o^*=o`*********............***=^**\OO***.**********.,@@@^....',0Ah
.rdata:0000000140009040 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E+db '......O@@@^******=oo*****=O@[[[^**o\***********.........***O^*[[@OO***.********..*\@@@....',0Ah
.rdata:0000000140009040 2E 0A 2E 2E 2E 2E 2E 2E 2E 2E+db '.....=O@@O*******oo^****,OO....\`*oo`**********.......*****O^=...\Oo****.********.,@@/*...',0Ah
.rdata:0000000140009040 2E 2E 2E 2F 4F 40 2F 2C 2A 2E+db '.....o/@@^*******oo`****O/.....,O`oo\***********.....****=O*^....\O`*************@@^*...',0Ah
.rdata:0000000140009040 2E 2E 2E 2C 2F 4F 40 40 4F 4F+db '....=//@@***.***=oo^**,O/.......,o`*o^******..........***,O^/......,Oo]****\*******=@^**..',0Ah
.rdata:0000000140009040 4F 40 40 40 40 40 40 7A 6A 6E+db '....o*^@/**...**Oooo`\O`.........=O\*o\********......***oo=........,Oo^***O******.=@^**^.',0Ah
.rdata:0000000140009040 75 63 74 66 7B 54 61 66 66 79+db '...o^=ooO**...*=Ooo^oO`.....,]]]..,O\ooooo]`***********o=o.,]]].....\O^**\^******=Oo\**.',0Ah
.rdata:0000000140009040 40 40 40 40 40 40 40 40 40 40+db '...^/oooo**..**=OooO/....]@@@@@@@]/`,Ooooooooo`********/*.]@@@@@@@]`...=O^*=^*******ooo\*.',0Ah
.rdata:0000000140009040 40 40 40 40 40 40 40 40 40 40+db '...oooooo**..**=OoO`../@@@@@@@@@@@@@^.,Ooo\oooo\`****`o\@@@@@@@@@@@@@`..O\=o***..***\ooo.',0Ah
.rdata:0000000140009040 40 5C 60 2E 2A 5C 4F 4F 6F 5C+db '...[..=oo**..**=OO../@@@@[/@@@@@@@@`\^..,O`\Oooooo/\oo./.@@@@@@@@\,\@@@\..\OO***..*=*^..[.',0Ah
.rdata:0000000140009040 40 4F 5C 2E 2E 2E 2E 2E 2E 2E+db '....../*\^*.***=O^.=@@/..@@@@@@@@^...........,\Ooooo/..,@@@@@@@@@....,@@@..O^**..****[....',0Ah
.rdata:0000000140009040 2E 2E 0A 2E 2E 2E 2E 2E 2E 2E+db '.....,o*=o*.****O^.@@^..=@@@@@@@@@@@@...............@@@@@@@@@@@@..,@@^=O^**...=^*=^...',0Ah
.rdata:0000000140009040 2E 2E 2E 2F 4F 4F 4F 2A 2C 5D+db '.....=`*=o\*****OO.\@...@@@/*@@@@.,@@^................@@[.=@@@*[@@@...=@`/O**.***/^**o...',0Ah
.rdata:0000000140009040 5D 5D 2F 4F 40 40 4F 4F 4F 4F+db '.....o**=^o`****=O......\/`...[[....@`...............\`...,[`...@@......O/*****/**.*=...',0Ah
.rdata:0000000140009040 4F 4F 4F 40 40 40 40 40 40 40+db '...../**=^\o`*.*=O^.................=^.................]`........@`......O^*.**/***.**...',0Ah
.rdata:0000000140009040 40 40 40 40 2F 5B 2A 2A 2A 2A+db '.....o**,^**o*..*O\.................................................=O*.*./****......',0Ah
.rdata:0000000140009040 2A 2A 2C 5B 40 40 40 40 40 40+db '.....=***o***\`**=O\...............................................*/O^**./*..=...*...',0Ah
.rdata:0000000140009040 40 40 40 40 40 40 40 40 40 40+db '......o**o****=o`**O\.....*****.............................*..*.....=O^..,^**..,..,`...',0Ah
.rdata:0000000140009040 40 40 40 40 4F 4F 5D 5D 5D 5D+db '.......\*=^*****\o\*,O`...................................................//**,/**....^*=`....',0Ah
.rdata:0000000140009040 5D 4F 4F 4F 4F 2E 2E 2E 2E 2E+db '......=^**^*******oooo\]o`................................................/\//ooo***...*^*,.....',0Ah
.rdata:0000000140009040 2E 2E 2E 0A 2E 2E 2E 2E 2E 2E+db '......O^**o******=ooooooo\]...............................,/oooo/^o^***...=*..,^...',0Ah
.rdata:0000000140009040 2E 2E 2E 2E 5C 4F 4F 4F 4F 4F+db '......OO**=o*****oooooooooooo]`.............................]]ooooooooo`,o***...**.*o^...',0Ah
.rdata:0000000140009040 40 40 40 4F 6F 4F 40 4F 4F 4F+db '......,O\*,o*****,^*,ooooooooOO`...o]]................,]/^...\Oooooo*********..=***/O....',0Ah
.rdata:0000000140009040 4F 4F 4F 4F 40 40 40 40 40 40+db '........\Ooo^*****^****,[[\o/.,]@@ooooooO**[[[[[[[*/oooooo@@\`.,O^***********..*o/oOO.....',0Ah
.rdata:0000000140009040 40 5B 2A 2A 2A 2A 2A 2A 2A 2A+db '......./oO@@@`**********,o[,@@@@@OooooooooO\******]OooooooO@@@@@^,O]*****o**.../OOO/,.....',0Ah
.rdata:0000000140009040 2A 2A 2A 2A 2A 2A 2A 2A 2C 5C+db '...../oo@@@O/O****=^......=@@@@@@ooooOOooOO****OOooOOoooo/\@@@@@\....,[[^**../@[@@@o]*...',0Ah
.rdata:0000000140009040 40 40 40 40 40 40 40 40 40 40+db '......,oo[/o\/o`*oo^...../@@@@@@^oooO@@@@/ooo**/oo\@@@@O`...@@@@@@@.....=,o`./@*`*,/**.,..',0Ah
```

另一段在shift+F12里 _1ik3_Y0u}



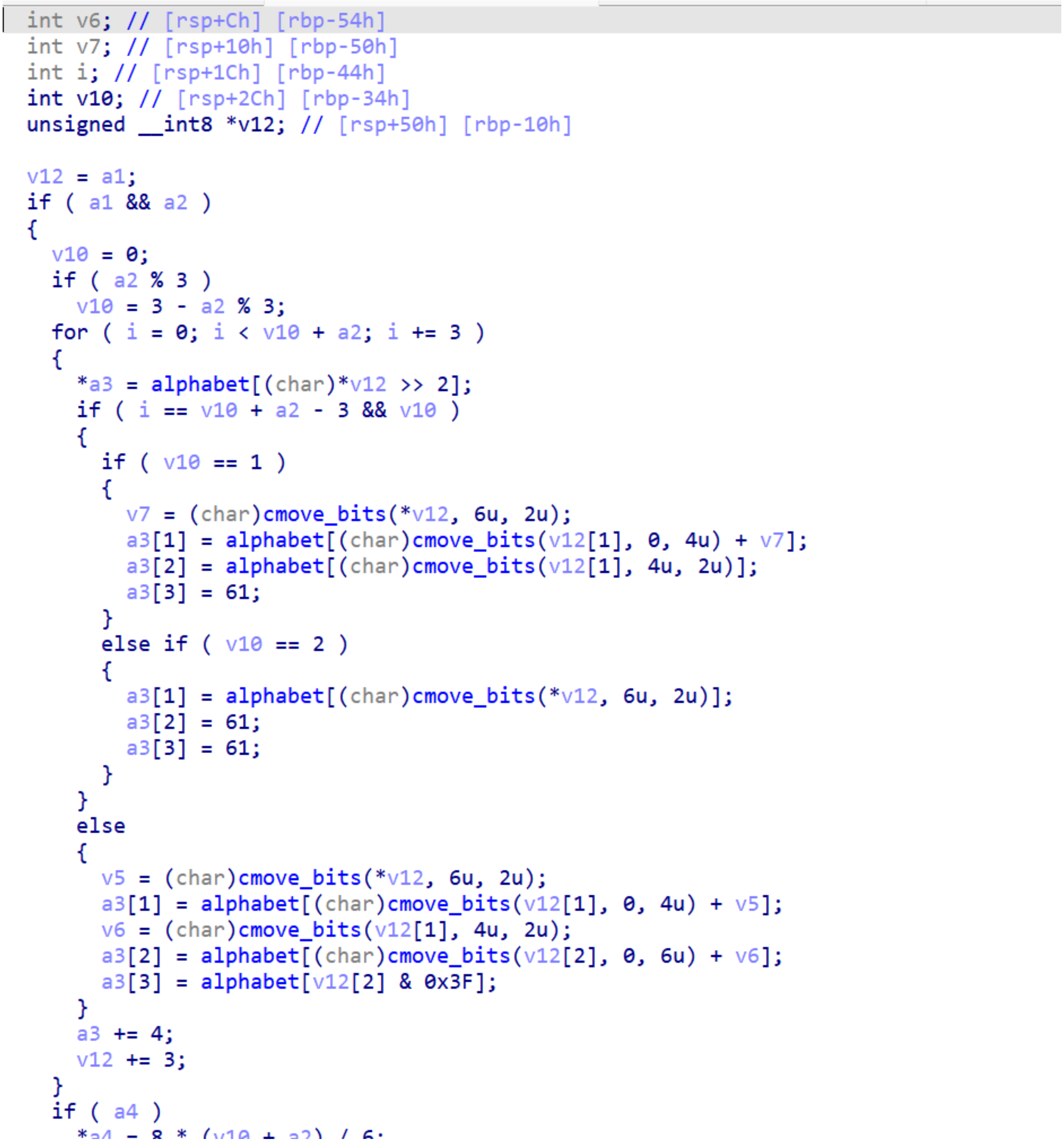
Address	Length	Type	String
.data:00···	00000023	C	_1ik3_Y0u}--->The latter part flag
.rdata:0···	0000002F	C	\nStaring at Taffy for 5 seconds can get flag?\n
.rdata:0···	00000006	C	pause
.rdata:0···	00000FFF	C	...............,O@@@/[[]OOOoO@@@@@@@OO[[[[[..=//O@@@[[[\\@OO@@@@@@O\\....,\\O@@@O`.............\n........
.rdata:0···	0000000E	C	Unknown error
.rdata:0···	0000001F	C	Argument domain error (DOMAIN)
.rdata:0···	00000020	C	Overflow range error (OVERFLOW)

合起来zjnuctf{Taffy_1ik3_Y0u}

mikumiku我们稀饭你

一个linux端下的elf文件

主程序打开能直接看到个encode

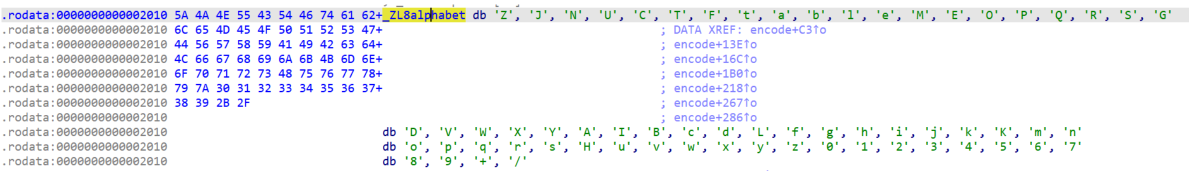


```
  int v6; // [rsp+Ch] [rbp-54h]
  int v7; // [rsp+10h] [rbp-50h]
  int i; // [rsp+1Ch] [rbp-44h]
  int v10; // [rsp+2Ch] [rbp-34h]
  unsigned __int8 *v12; // [rsp+50h] [rbp-10h]

  v12 = a1;
  if ( a1 && a2 )
  {
    v10 = 0;
    if ( a2 % 3 )
      v10 = 3 - a2 % 3;
    for ( i = 0; i < v10 + a2; i += 3 )
    {
      *a3 = alphabet[(char)*v12 >> 2];
      if ( i == v10 + a2 - 3 && v10 )
      {
        if ( v10 == 1 )
        {
          v7 = (char)cmove_bits(*v12, 6u, 2u);
          a3[1] = alphabet[(char)cmove_bits(v12[1], 0, 4u) + v7];
          a3[2] = alphabet[(char)cmove_bits(v12[1], 4u, 2u)];
          a3[3] = 61;
        }
        else if ( v10 == 2 )
        {
          a3[1] = alphabet[(char)cmove_bits(*v12, 6u, 2u)];
          a3[2] = 61;
          a3[3] = 61;
        }
      }
      else
      {
        v5 = (char)cmove_bits(*v12, 6u, 2u);
        a3[1] = alphabet[(char)cmove_bits(v12[1], 0, 4u) + v5];
        v6 = (char)cmove_bits(v12[1], 4u, 2u);
        a3[2] = alphabet[(char)cmove_bits(v12[2], 0, 6u) + v6];
        a3[3] = alphabet[v12[2] & 0x3F];
      }
      a3 += 4;
      v12 += 3;
    }
    if ( a4 )
      *a4 = 8 * (v10 + a2) / 6;
```

根据特征一看就是base64加密

再看看表，发现换表了



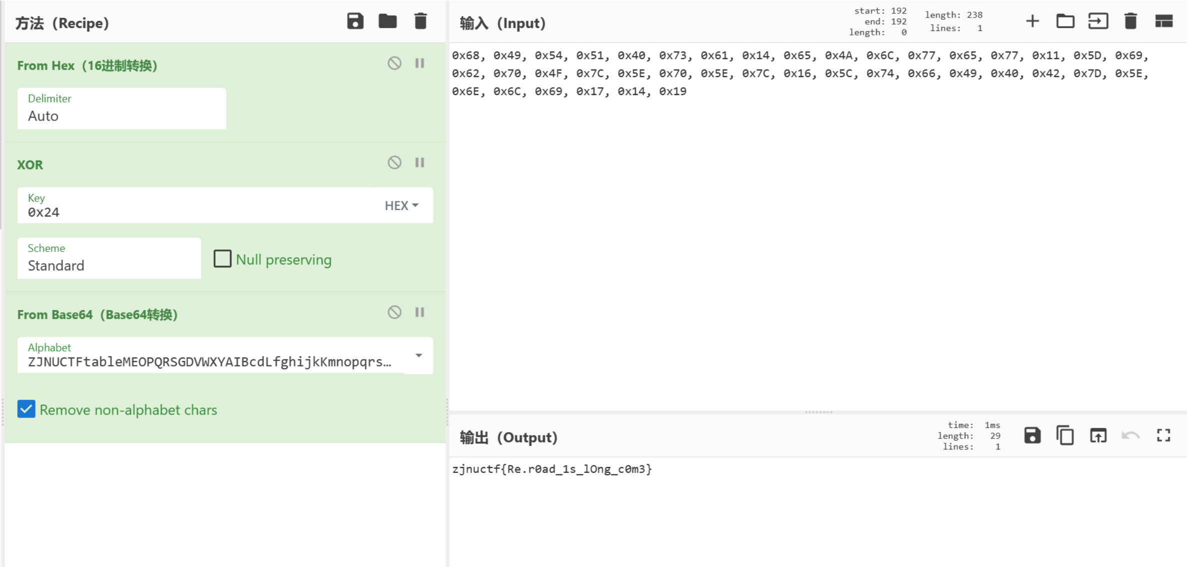
```
.rodata:0000000000002010 5A 4A 4E 55 43 54 46 74 61 62+_ZL8alphabet db 'Z', 'J', 'N', 'U', 'C', 'T', 'F', 't', 'a', 'b', 'l', 'e', 'M', 'E', 'O', 'P', 'Q', 'R', 'S', 'G'
.rodata:0000000000002010 6C 65 4D 45 4F 50 51 52 53 47+                                     ; DATA XREF: encode+C3↑o
.rodata:0000000000002010 44 56 57 58 59 41 49 42 63 64+                                     ; encode+13E↑o
.rodata:0000000000002010 4C 66 67 68 69 6A 6B 4B 6D 6E+                                     ; encode+16C↑o
.rodata:0000000000002010 6F 70 71 72 73 48 75 76 77 78+                                     ; encode+1B0↑o
.rodata:0000000000002010 79 7A 30 31 32 33 34 35 36 37+                                     ; encode+218↑o
.rodata:0000000000002010 38 39 2B 2F                                                        ; encode+267↑o
.rodata:0000000000002010                                                                    ; encode+286↑o
.rodata:0000000000002010                               db 'D', 'V', 'W', 'X', 'Y', 'A', 'I', 'B', 'c', 'd', 'L', 'f', 'g', 'h', 'i', 'j', 'k', 'K', 'm', 'n'
.rodata:0000000000002010                               db 'o', 'p', 'q', 'r', 's', 'H', 'u', 'v', 'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7'
.rodata:0000000000002010                               db '8', '9', '+', '/'
```

把表提出来ZJNUCTFtableMEOPQRSGDVWXYAIBcdLfghijkKmnopqrsHuvwxyz0123456789+/

再看main函数还有一个异或0x24的操作

再找到密文dest

理清思路可以用赛博厨子直接出了



方法（Recipe）

From Hex（16进制转换）
Delimiter: Auto

XOR
Key: 0x24 HEX
Scheme: Standard
□ Null preserving

From Base64（Base64转换）
Alphabet: ZJNUCTFtableMEOPQRSGDVWXYAIBcdLfghijkKmnopqrs…
☑ Remove non-alphabet chars

输入（Input）

```
start: 192   length: 238
  end: 192    lines:   1
length:   0
```

```
0x68, 0x49, 0x54, 0x51, 0x40, 0x73, 0x61, 0x14, 0x65, 0x4A, 0x6C, 0x77, 0x65, 0x77, 0x11, 0x5D, 0x69,
0x62, 0x70, 0x4F, 0x7C, 0x5E, 0x70, 0x5E, 0x7C, 0x16, 0x5C, 0x74, 0x66, 0x49, 0x40, 0x42, 0x7D, 0x5E,
0x6E, 0x6C, 0x69, 0x17, 0x14, 0x19
```

输出（Output）

```
  time:  1ms
length:   29
 lines:    1
```

```
zjnuctf{Re.r0ad_1s_lOng_c0m3}
```

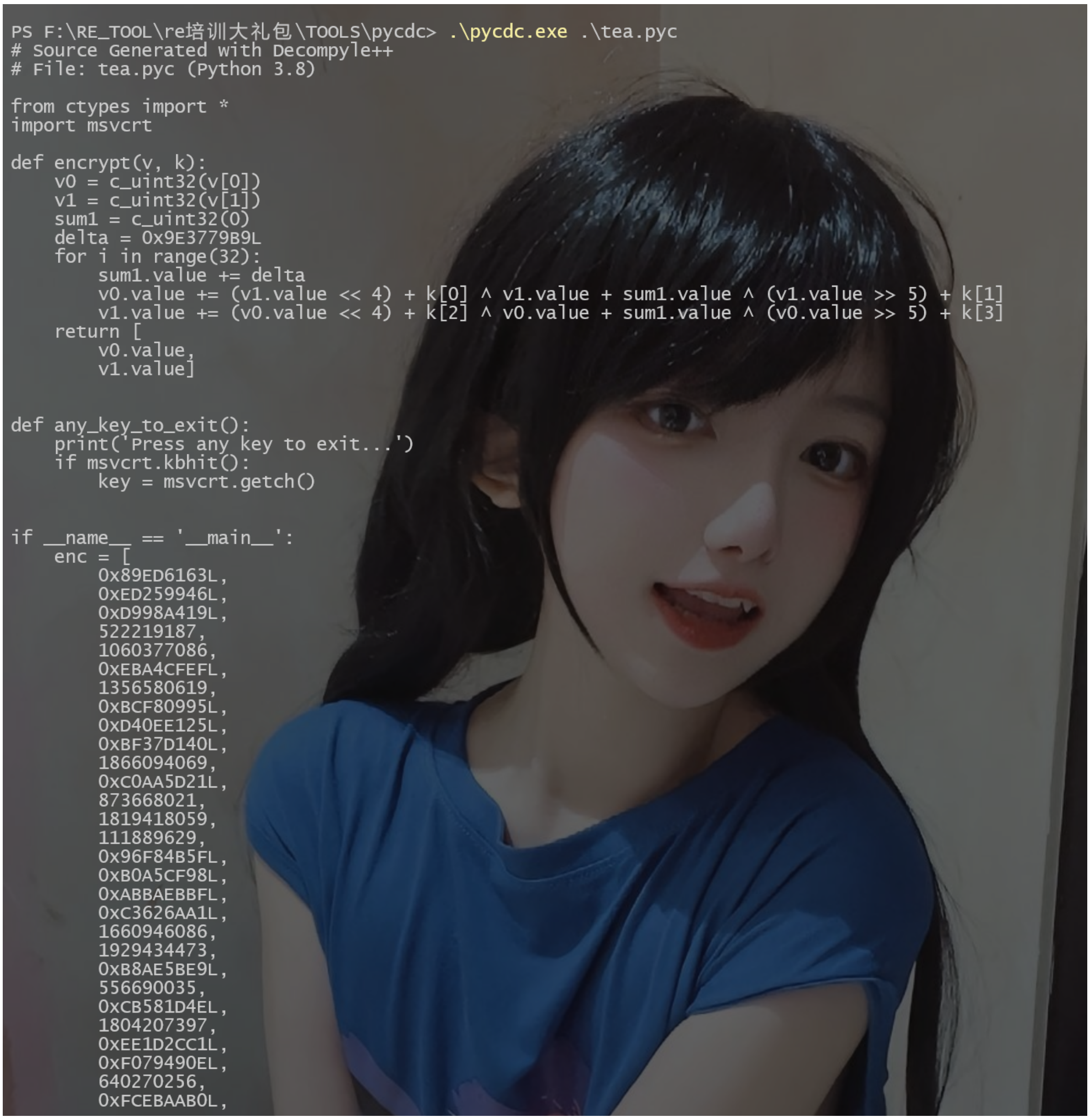
zjnuctf{Re.r0ad_1s_lOng_c0m3}

小彩蛋：在linux端下运行，把这个输入，就可以解密出what_is_me，得到miku的照片一张

派森茶

就是python tea的意思，知道这题是一道python打包的题，最好解包的python版本在3.8——3.10，不然会有问题。

拉到pyinstxtractor中执行python pyinstxtractor.py 派森茶.exe进行反编译，用pycdc.exe看文件中的tea.pyc文件



```
PS F:\RE_TOOL\re培训大礼包\TOOLS\pycdc> .\pycdc.exe .\tea.pyc
# Source Generated with Decompyle++
# File: tea.pyc (Python 3.8)

from ctypes import *
import msvcrt

def encrypt(v, k):
    v0 = c_uint32(v[0])
    v1 = c_uint32(v[1])
    sum1 = c_uint32(0)
    delta = 0x9E3779B9L
    for i in range(32):
        sum1.value += delta
        v0.value += (v1.value << 4) + k[0] ^ v1.value + sum1.value ^ (v1.value >> 5) + k[1]
        v1.value += (v0.value << 4) + k[2] ^ v0.value + sum1.value ^ (v0.value >> 5) + k[3]
    return [
        v0.value,
        v1.value]


def any_key_to_exit():
    print('Press any key to exit...')
    if msvcrt.kbhit():
        key = msvcrt.getch()


if __name__ == '__main__':
    enc = [
        0x89ED6163L,
        0xED259946L,
        0xD998A419L,
        522219187,
        1060377086,
        0xEBA4CFEFL,
        1356580619,
        0xBCF80995L,
        0xD40EE125L,
        0xBF37D140L,
        1866094069,
        0xC0AA5D21L,
        873668021,
        1819418059,
        111889629,
        0x96F84B5FL,
        0xB0A5CF98L,
        0xABBAEBBFL,
        0xC3626AA1L,
        1660946086,
        1929434473,
        0xB8AE5BE9L,
        556690035,
        0xCB581D4EL,
        1804207397,
        0xEE1D2CC1L,
        0xF079490EL,
        640270256,
        0xFCEBAAB0L,
```

可以得到反编译后的，发现加密是一个很标准的tea加密，并且给出了密文，和密钥1145

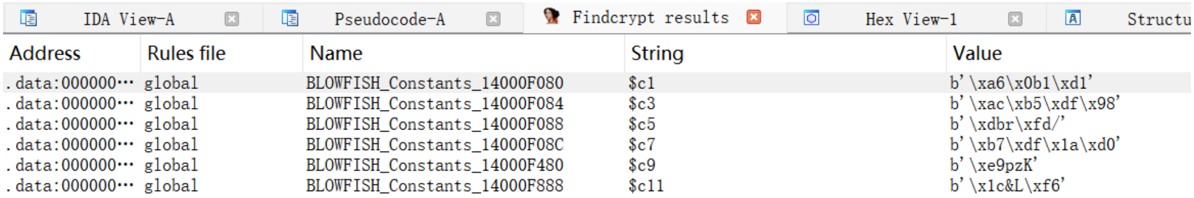
直接copy一个脚本改一下就出

```
#include<stdio.h>
int main(){
    int n=32;//pw的个数
    unsigned int pw[32]={
        0x89ed6163, 0xed259946,0xd998a419,0x1f206eb3,0x3f3411fe, 0xeba4cfef,
        0x50dbc70b, 0xbcf80995,0xd40ee125,0xbf37d140,0x6f3a55f5, 0xc0aa5d21,
        0x34131db5, 0x6c721dcb,0x06ab4cdd,0x96f84b5f,0xb0a5cf98, 0xabbaebbf,
        0xc3626aa1, 0x630006a6,0x7300d569,0xb8ae5be9,0x212e6a73, 0xcb581d4e,
        0x6b8a0525, 0xee1d2cc1,0xf079490e,0x2629bfb0,0xfcebaab0, 0x7202e516,
        0xa37a21f1, 0x05289fd8};//可改
    unsigned int v0;
    unsigned int v1;
    unsigned int sum;
    unsigned int key[4]={1,1,4,5};//可改
    for(int i=0;i<n/2;i++)
    {
        v0=pw[2*i];
        v1=pw[2*i+1];
        sum=-32*0x61C88647;
        for(int i=0;i<32;i++)
        {
            v1 -= ((v0 >> 5) + key[3] )^ (16 * v0 + key[2]) ^ (sum + v0);//容易魔改
            v0 -= ((v1 >> 5) + key[1]) ^ (16 * v1 + key[0]) ^ (sum + v1);
            sum += 0x61C88647;//容易魔改
        }
        for (int j = 0; j<=3; j++)
        {
            printf("%c", (v0 >> (j * 8)) & 0xFF);
        }
        for (int j = 0; j<=3; j++)
        {
            printf("%c", (v1 >> (j * 8)) & 0xFF);
        }
    }
}
```

zjnuctf{1et_us_dr1nk_pyth0n_t3a}

老爹的武器

无时无刻都在提醒你这是一个河豚加密(blowfish)，用findcrypto也能看出来



Address	Rules file	Name	String	Value
.data:000000···	global	BLOWFISH_Constants_14000F080	$c1	b'\xa6\x0b1\xd1'
.data:000000···	global	BLOWFISH_Constants_14000F084	$c3	b'\xac\xb5\xdf\x98'
.data:000000···	global	BLOWFISH_Constants_14000F088	$c5	b'\xdbr\xfd/'
.data:000000···	global	BLOWFISH_Constants_14000F08C	$c7	b'\xb7\xdf\x1a\xd0'
.data:000000···	global	BLOWFISH_Constants_14000F480	$c9	b'\xe9pzK'
.data:000000···	global	BLOWFISH_Constants_14000F888	$c11	b'\x1c&L\xf6'

找到密文和密钥就直接能出了，因为这里用的是一个blowfish ecb模式的加密，还涉及到填充，想用cyberchef一把嗦行不通，最好自己找个对应的python脚本。

```
from Crypto.Cipher import Blowfish
import codecs

class BlowfishCipher:
    def __init__(self):
        pass

    def encrypt(self, plaintext, key):
        key = key.encode("utf-8")
        cipher = Blowfish.new(key, Blowfish.MODE_ECB)

        # 将明文填充到8字节的倍数
        plaintext = plaintext.ljust((len(plaintext) + 7) // 8 * 8)

        ciphertext = cipher.encrypt(plaintext.encode('utf-8'))
        hex_encode = codecs.encode(ciphertext, 'hex_codec').decode('utf-8')
        return hex_encode

    def decrypt(self, ciphertext, key):
        key = key.encode("utf-8")
        cipher = Blowfish.new(key, Blowfish.MODE_ECB)
        ciphertext = codecs.decode(ciphertext, 'hex_codec')
        decrypted_text = cipher.decrypt(ciphertext).decode('utf-8').rstrip()
        return decrypted_text

if __name__ == '__main__':
    key = 'hetuno.O'
    blowfish_cipher = BlowfishCipher()
    encrypted_text='9d0ec04ba44e01aa19eaa0302a66a90f'
    decrypted_text = blowfish_cipher.decrypt(encrypted_text, key)
    print(f"加密: {encrypted_text}, 解密: {decrypted_text}")
```

zjnuctf{pAn9ba1}

AboutBear

flag由账号加密码两部分组成

账号就是一个java层的blowfish加密，数据直接放到赛博厨子上就能解出来

密码的加密部分就一个标准rc4，数据放在java层，加密放在so层

最后就是加了一个梆梆加固的壳，用frida-dexdump脱壳即可

```
账号：A_Green_b3ar_
密码：Fr0m_soft_L1pa
flag:flag{A_Green_b3ar_Fr0m_soft_L1pa}
```