逆向---入坑记

原创

IzuruKamuku 于 2018-11-23 17:49:17 发布 385 收藏 1

Hello,RE!

80

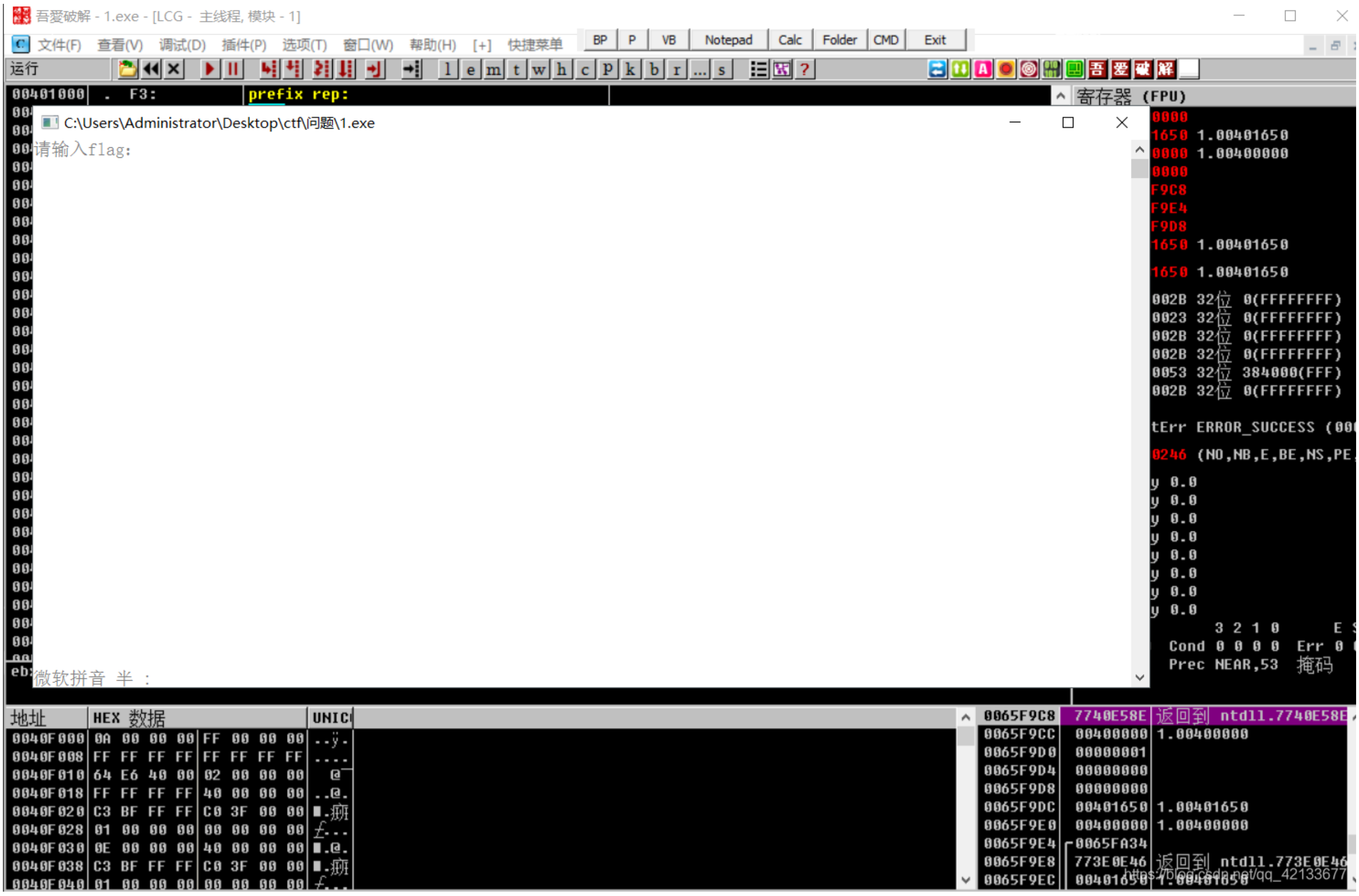
或许你需要去学习下IDA的使用，但是只需要学一点点就能做这题了
PS:IDA里面按R可以把奇怪的数字变成字符串

格式为`flag{*****}`包含flag{}提交

参考资料：
《IDA Pro 权威指南》
各种CTF比赛的逆向部分的writeup

http://ctf.nuptzj.cn/static/uploads/0b562710385edcf655dfa0ae65c69592/1.exe

入门题，直接丢到OD里运行：


吾爱破解 - 1.exe - [LCG - 主线程, 模块 - 1]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+] 快捷菜单
BP P VB Notepad Calc Folder CMD Exit
运行
寄存器 (FPU)
地址 HEX 数据 UNIC

```
00401000  .  F3:        prefix rep:
```

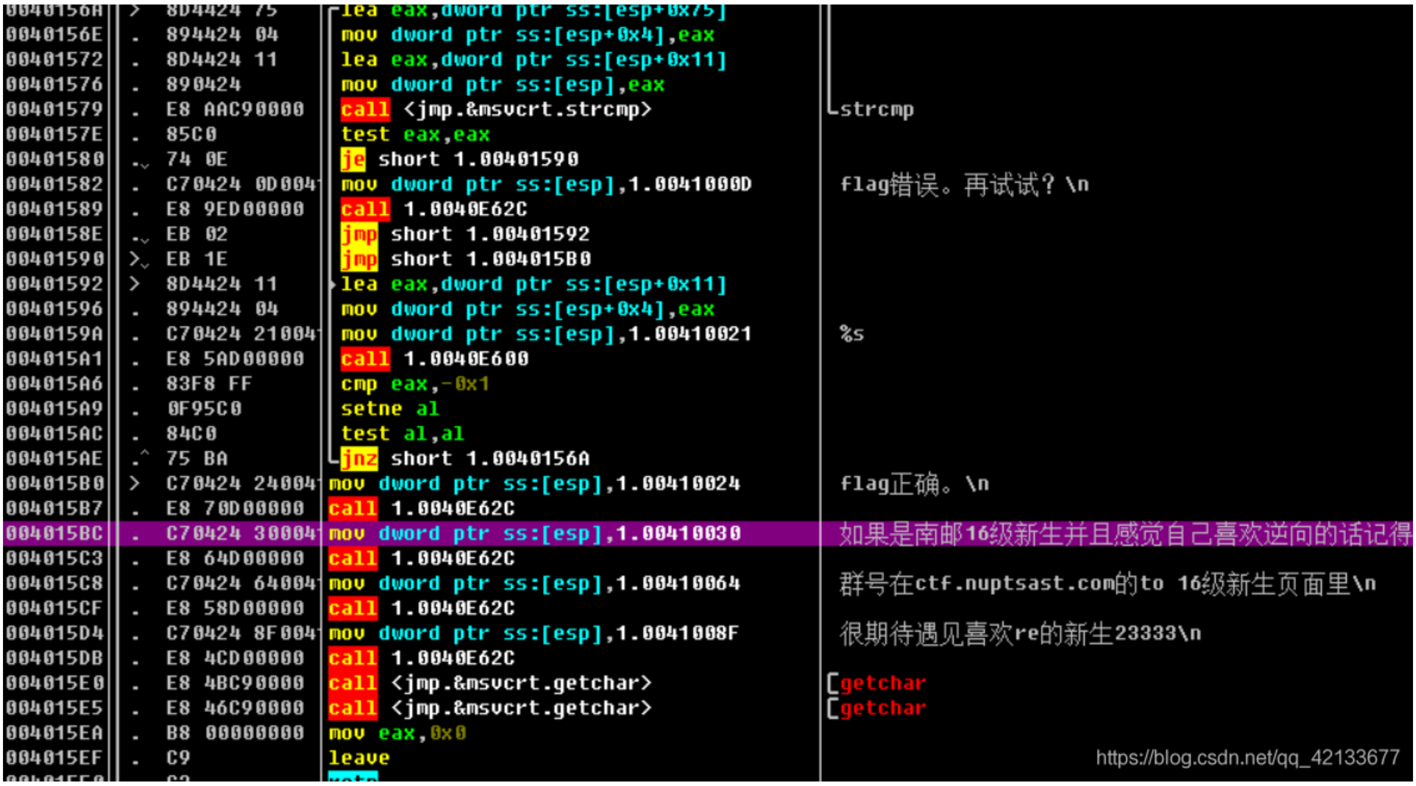
```
C:\Users\Administrator\Desktop\ctf\问题\1.exe
请输入flag:
```

```
地址     HEX 数据                 UNIC
0040F000 0A 00 00 00 FF 00 00 00 ..ÿ.
0040F008 FF FF FF FF FF FF FF FF ....
0040F010 64 E6 40 00 02 00 00 00   @
0040F018 FF FF FF FF 40 00 00 00 ..@.
0040F020 C3 BF FF FF C0 3F 00 00 ■.掀
0040F028 01 00 00 00 00 00 00 00 £...
0040F030 0E 00 00 00 40 00 00 00 ■.@.
0040F038 C3 BF FF FF C0 3F 00 00 ■.掀
0040F040 01 00 00 00 00 00 00 00 £...
```

```
0065F9C8  7740E58E  返回到 ntdll.7740E58E
0065F9CC  00400000  1.00400000
0065F9D0  00000001
0065F9D4  00000000
0065F9D8  00000000
0065F9DC  00401650  1.00401650
0065F9E0  00400000  1.00400000
0065F9E4 ┌0065FA34
0065F9E8  773E0E46  返回到 ntdll.773E0E46
0065F9EC  00401650  1.00401650
```

```
0000
1650 1.00401650
0000 1.00400000
0000
F9C8
F9E4
F9D8
1650 1.00401650

1650 1.00401650

002B 32位 0(FFFFFFFF)
0023 32位 0(FFFFFFFF)
002B 32位 0(FFFFFFFF)
002B 32位 0(FFFFFFFF)
0053 32位 384000(FFF)
002B 32位 0(FFFFFFFF)

tErr ERROR_SUCCESS (00
0246 (NO,NB,E,BE,NS,PE

y 0.0
y 0.0
y 0.0
y 0.0
y 0.0
y 0.0
y 0.0
y 0.0
      3 2 1 0       E
 Cond 0 0 0 0   Err 0
 Prec NEAR,53   掩码
```

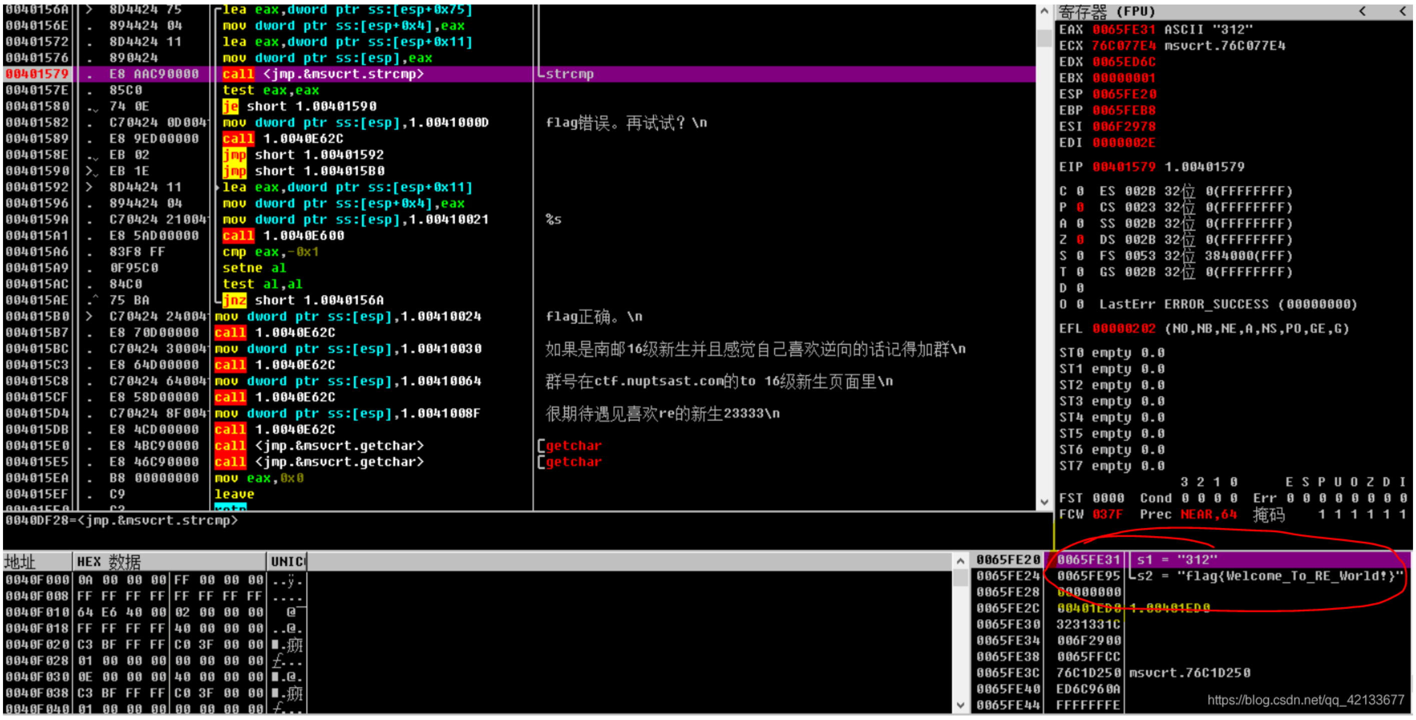
直接搜索中文字符：



```
0040156A  > 8D4424 75      lea eax,dword ptr ss:[esp+0x75]
0040156E  . 894424 04      mov dword ptr ss:[esp+0x4],eax
00401572  . 8D4424 11      lea eax,dword ptr ss:[esp+0x11]
00401576  . 890424         mov dword ptr ss:[esp],eax
00401579  . E8 AAC90000    call <jmp.&msvcrt.strcmp>           └strcmp
0040157E  . 85C0           test eax,eax
00401580  .v 74 0E         je short 1.00401590
00401582  . C70424 0D004   mov dword ptr ss:[esp],1.0041000D   flag错误。再试试？\n
00401589  . E8 9ED00000    call 1.0040E62C
0040158E  .v EB 02         jmp short 1.00401592
00401590  >v EB 1E         jmp short 1.004015B0
00401592  > 8D4424 11      lea eax,dword ptr ss:[esp+0x11]
00401596  . 894424 04      mov dword ptr ss:[esp+0x4],eax
0040159A  . C70424 21004   mov dword ptr ss:[esp],1.00410021   %s
004015A1  . E8 5AD00000    call 1.0040E600
004015A6  . 83F8 FF        cmp eax,-0x1
004015A9  . 0F95C0         setne al
004015AC  . 84C0           test al,al
004015AE  .^ 75 BA         jnz short 1.0040156A
004015B0  > C70424 24004   mov dword ptr ss:[esp],1.00410024   flag正确。\n
004015B7  . E8 70D00000    call 1.0040E62C
004015BC  . C70424 30004   mov dword ptr ss:[esp],1.00410030   如果是南邮16级新生并且感觉自己喜欢逆向的话记得
004015C3  . E8 64D00000    call 1.0040E62C
004015C8  . C70424 64004   mov dword ptr ss:[esp],1.00410064   群号在ctf.nuptsast.com的to 16级新生页面里\n
004015CF  . E8 58D00000    call 1.0040E62C
004015D4  . C70424 8F004   mov dword ptr ss:[esp],1.0041008F   很期待遇见喜欢re的新生23333\n
004015DB  . E8 4CD00000    call 1.0040E62C
004015E0  . E8 4BC90000    call <jmp.&msvcrt.getchar>          [getchar
004015E5  . E8 46C90000    call <jmp.&msvcrt.getchar>          [getchar
004015EA  . B8 00000000    mov eax,0x0
004015EF  . C9             leave
```

https://blog.csdn.net/qq_42133677

可以看到一个字符比较函数，在哪里加一个断点试试（F2）运行：



```
0040156A  > 8D4424 75      lea eax,dword ptr ss:[esp+0x75]
0040156E  . 894424 04      mov dword ptr ss:[esp+0x4],eax
00401572  . 8D4424 11      lea eax,dword ptr ss:[esp+0x11]
00401576  . 890424         mov dword ptr ss:[esp],eax
00401579  . E8 AAC90000    call <jmp.&msvcrt.strcmp>           └strcmp
0040157E  . 85C0           test eax,eax
00401580  .v 74 0E         je short 1.00401590
00401582  . C70424 0D004   mov dword ptr ss:[esp],1.0041000D   flag错误。再试试？\n
00401589  . E8 9ED00000    call 1.0040E62C
0040158E  .v EB 02         jmp short 1.00401592
00401590  >v EB 1E         jmp short 1.004015B0
00401592  > 8D4424 11      lea eax,dword ptr ss:[esp+0x11]
00401596  . 894424 04      mov dword ptr ss:[esp+0x4],eax
0040159A  . C70424 21004   mov dword ptr ss:[esp],1.00410021   %s
004015A1  . E8 5AD00000    call 1.0040E600
004015A6  . 83F8 FF        cmp eax,-0x1
004015A9  . 0F95C0         setne al
004015AC  . 84C0           test al,al
004015AE  .^ 75 BA         jnz short 1.0040156A
004015B0  > C70424 24004   mov dword ptr ss:[esp],1.00410024   flag正确。\n
004015B7  . E8 70D00000    call 1.0040E62C
004015BC  . C70424 30004   mov dword ptr ss:[esp],1.00410030   如果是南邮16级新生并且感觉自己喜欢逆向的话记得加群\n
004015C3  . E8 64D00000    call 1.0040E62C
004015C8  . C70424 64004   mov dword ptr ss:[esp],1.00410064   群号在ctf.nuptsast.com的to 16级新生页面里\n
004015CF  . E8 58D00000    call 1.0040E62C
004015D4  . C70424 8F004   mov dword ptr ss:[esp],1.0041008F   很期待遇见喜欢re的新生23333\n
004015DB  . E8 4CD00000    call 1.0040E62C
004015E0  . E8 4BC90000    call <jmp.&msvcrt.getchar>          [getchar
004015E5  . E8 46C90000    call <jmp.&msvcrt.getchar>          [getchar
004015EA  . B8 00000000    mov eax,0x0
004015EF  . C9             leave
0040DF28=<jmp.&msvcrt.strcmp>
```

```
寄存器 (FPU)
EAX 0065FE31 ASCII "312"
ECX 76C077E4 msvcrt.76C077E4
EDX 0065ED6C
EBX 00000001
ESP 0065FE20
EBP 0065FEB8
ESI 006F2978
EDI 0000002E

EIP 00401579 1.00401579

C 0  ES 002B 32位 0(FFFFFFFF)
P 0  CS 0023 32位 0(FFFFFFFF)
A 0  SS 002B 32位 0(FFFFFFFF)
Z 0  DS 002B 32位 0(FFFFFFFF)
S 0  FS 0053 32位 384000(FFF)
T 0  GS 002B 32位 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
                   3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0
FCW 037F  Prec NEAR,64  掩码      1 1 1 1 1 1
```

```
地址      HEX 数据                  UNIC
0040F000  0A 00 00 00 FF 00 00 00  ..ÿ.
0040F008  FF FF FF FF FF FF FF FF  ....
0040F010  64 E6 40 00 02 00 00 00   @
0040F018  FF FF FF FF 40 00 00 00  ..@.
0040F020  C3 BF FF FF C0 3F 00 00  ■.瘢
0040F028  01 00 00 00 00 00 00 00  £...
0040F030  0E 00 00 00 40 00 00 00  ■.@.
0040F038  C3 BF FF FF C0 3F 00 00  ■.瘢
0040F040  01 00 00 00 00 00 00 00  £...
```

```
0065FE20  0065FE31  ┌s1 = "312"
0065FE24  0065FE95  └s2 = "flag{Welcome_To_RE_World!}"
0065FE28  00000000
0065FE2C  00401ED0  1.00401ED0
0065FE30  3231331C
0065FE34  006F2900
0065FE38  0065FFCC
0065FE3C  76C1D250  msvcrt.76C1D250
0065FE40  ED6C960A
0065FE44  FFFFFFFE
```

https://blog.csdn.net/qq_42133677

可以看到 flag啦！！！

ReadAsm2

150

读汇编是逆向基本功。

给出的文件是func函数的汇编
main函数如下
输出的结果即为flag，格式为`flag{**********}`，请连flag{}一起提交

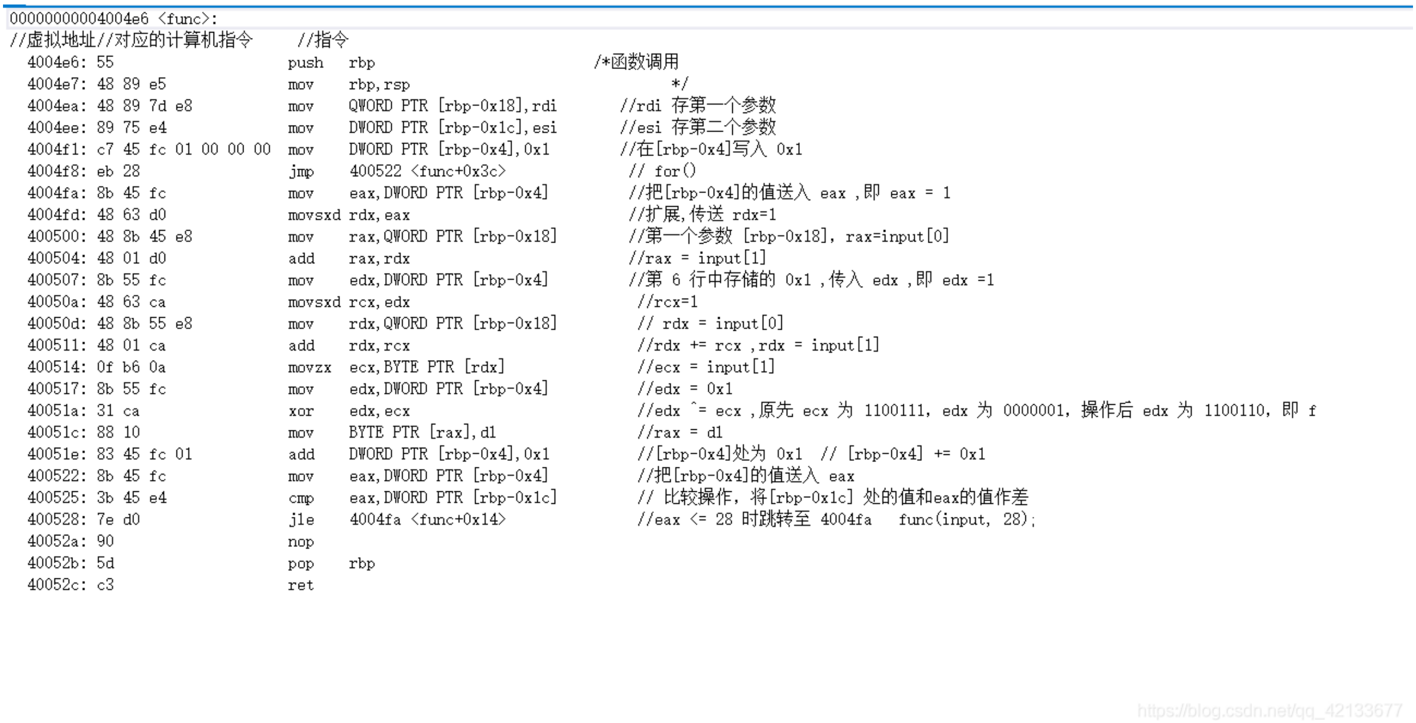
编译环境为linux gcc x86-64
调用约定为System V AMD64 ABI
请不要利用汇编器，IDA等工具。。这里考的就是读汇编与推算汇编结果的能力

```
int main(int argc, char const *argv[])
{
  char input[] = {0x0,  0x67, 0x6e, 0x62, 0x63, 0x7e, 0x74, 0x62, 0x69, 0x6d,
                  0x55, 0x6a, 0x7f, 0x60, 0x51, 0x66, 0x63, 0x4e, 0x66, 0x7b,
                  0x71, 0x4a, 0x74, 0x76, 0x6b, 0x70, 0x79, 0x66 , 0x1c};
  func(input, 28);
  printf("%s\n",input+1);
  return 0;
}
```

参考资料:
https://github.com/veficos/reverse-engineering-for-beginners
《汇编语言》王爽
《C 反汇编与逆向分析技术揭秘》

http://ctf.nuptzj.cn/static/uploads/a480ff52cdbc70bd1443763f27f35279/2.asm

把下载的汇编文件打开后（记事本就行）：



```
00000000004004e6 <func>:
//虚拟地址//对应的计算机指令      //指令
  4004e6: 55                    push   rbp                            /*函数调用
  4004e7: 48 89 e5              mov    rbp,rsp                                  */
  4004ea: 48 89 7d e8           mov    QWORD PTR [rbp-0x18],rdi          //rdi 存第一个参数
  4004ee: 89 75 e4              mov    DWORD PTR [rbp-0x1c],esi          //esi 存第二个参数
  4004f1: c7 45 fc 01 00 00 00  mov    DWORD PTR [rbp-0x4],0x1           //在[rbp-0x4]写入 0x1
  4004f8: eb 28                 jmp    400522 <func+0x3c>                 // for()
  4004fa: 8b 45 fc              mov    eax,DWORD PTR [rbp-0x4]            //把[rbp-0x4]的值送入 eax ,即 eax = 1
  4004fd: 48 63 d0              movsxd rdx,eax                            //扩展,传送 rdx=1
  400500: 48 8b 45 e8           mov    rax,QWORD PTR [rbp-0x18]           //第一个参数 [rbp-0x18], rax=input[0]
  400504: 48 01 d0              add    rax,rdx                            //rax = input[1]
  400507: 8b 55 fc              mov    edx,DWORD PTR [rbp-0x4]            //第 6 行中存储的 0x1 ,传入 edx ,即 edx =1
  40050a: 48 63 ca              movsxd rcx,edx                             //rcx=1
  40050d: 48 8b 55 e8           mov    rdx,QWORD PTR [rbp-0x18]            // rdx = input[0]
  400511: 48 01 ca              add    rdx,rcx                             //rdx += rcx ,rdx = input[1]
  400514: 0f b6 0a              movzx  ecx,BYTE PTR [rdx]                  //ecx = input[1]
  400517: 8b 55 fc              mov    edx,DWORD PTR [rbp-0x4]             //edx = 0x1
  40051a: 31 ca                 xor    edx,ecx                             //edx ^= ecx ,原先 ecx 为 1100111, edx 为 0000001, 操作后 edx 为 1100110, 即 f
  40051c: 88 10                 mov    BYTE PTR [rax],dl                   //rax = dl
  40051e: 83 45 fc 01           add    DWORD PTR [rbp-0x4],0x1             //[rbp-0x4]处为 0x1  // [rbp-0x4] += 0x1
  400522: 8b 45 fc              mov    eax,DWORD PTR [rbp-0x4]             //把[rbp-0x4]的值送入 eax
  400525: 3b 45 e4              cmp    eax,DWORD PTR [rbp-0x1c]            // 比较操作, 将[rbp-0x1c] 处的值和eax的值作差
  400528: 7e d0                 jle    4004fa <func+0x14>                  //eax <= 28 时跳转至 4004fa   func(input, 28);
  40052a: 90                    nop
  40052b: 5d                    pop    rbp
  40052c: c3                    ret
```

然后写一个小程序即可：

```
a = [0x0, 0x67, 0x6e, 0x62, 0x63, 0x7e, 0x74, 0x62, 0x69, 0x6d,

0x55, 0x6a, 0x7f, 0x60, 0x51, 0x66, 0x63, 0x4e, 0x66, 0x7b,

0x71, 0x4a, 0x74, 0x76, 0x6b, 0x70, 0x79, 0x66 , 0x1c]

s = ''

for i in range(1,len(a)):
    print(a[i]^i,end = " ")
    s += chr(a[i]^i)
print()
print (s)
```

运行即可

得到flag;

Py交易

150

Python 2.7

提取密码：kd37

下载后直接丢到py反汇编的在网站：

让后写一个程序即可：

```
import base64

correct ='XlNkVmtUI1MgXWBZXCFeKY+AaXNt'

s = base64.b64decode(correct)

flag = ''

for i in s:

    i = chr((i-16)^32)   // 如果是py2环境i需要改为 ord(i);

    flag += i

print (flag)
```

总结：

入门题较为简单，不需要对汇编有多深的了解，会用工具即可；