Lab7 Reverse Engineering Exercises – Advanced

一、 实验目的

1、进一步熟悉静态反汇编工具 IDA Freeware；

2、熟悉将反汇编代码进行反编译的过程；

3、掌握对于反编译伪代码的逆向分析；

4、运用熟悉的编程语言，实现简单的脚本编写

二、 实验原理

(一) task3

1. 通过 IDA Freeware 得到 task3.exe 的反汇编代码，如图 1 和图 2 所示。

```
.text:00402A70                  sub     esp, 0A4h
.text:00402A76                  mov     eax, ___security_cookie
.text:00402A7B                  xor     eax, esp
.text:00402A7D                  mov     [esp+0A4h+var_4], eax
.text:00402A84                  mov     [esp+0A4h+var_80], 42h ; 'B'
.text:00402A89                  xor     ecx, ecx
.text:00402A8B                  mov     [esp+0A4h+var_7F], 7Eh ; '~'
.text:00402A90                  mov     [esp+0A4h+var_7E], 77h ; 'w'
.text:00402A95                  mov     [esp+0A4h+var_7D], 73h ; 's'
.text:00402A9A                  mov     [esp+0A4h+var_7C], 61h ; 'a'
.text:00402A9F                  mov     [esp+0A4h+var_7B], 77h ; 'w'
.text:00402AA4                  mov     [esp+0A4h+var_7A], 32h ; '2'
.text:00402AA9                  mov     [esp+0A4h+var_79], 7Bh ; '{'
.text:00402AAE                  mov     [esp+0A4h+var_78], 7Ch ; '|'
.text:00402AB3                  mov     [esp+0A4h+var_77], 62h ; 'b'
.text:00402AB8                  mov     [esp+0A4h+var_76], 67h ; 'g'
.text:00402ABD                  mov     [esp+0A4h+var_75], 66h ; 'f'
.text:00402AC2                  mov     [esp+0A4h+var_74], 32h ; '2'
.text:00402AC7                  mov     [esp+0A4h+var_73], 73h ; 's'
.text:00402ACC                  mov     [esp+0A4h+var_72], 32h ; '2'
.text:00402AD1                  mov     [esp+0A4h+var_71], 61h ; 'a'
.text:00402AD6                  mov     [esp+0A4h+var_70], 66h ; 'f'
.text:00402ADB                  mov     [esp+0A4h+var_6F], 60h ; '`'
.text:00402AE0                  mov     [esp+0A4h+var_6E], 7Bh ; '{'
.text:00402AE5                  mov     [esp+0A4h+var_6D], 7Ch ; '|'
.text:00402AEA                  mov     [esp+0A4h+var_6C], 75h ; 'u'
.text:00402AEF                  mov     [esp+0A4h+var_6B], 28h ; '('
.text:00402AF4                  mov     [esp+0A4h+var_6A], 18h
.text:00402AF9                  mov     [esp+0A4h+var_69], 12h
.text:00402AFE                  xchg    ax, ax
```

图 1　task3.exe 的反汇编代码

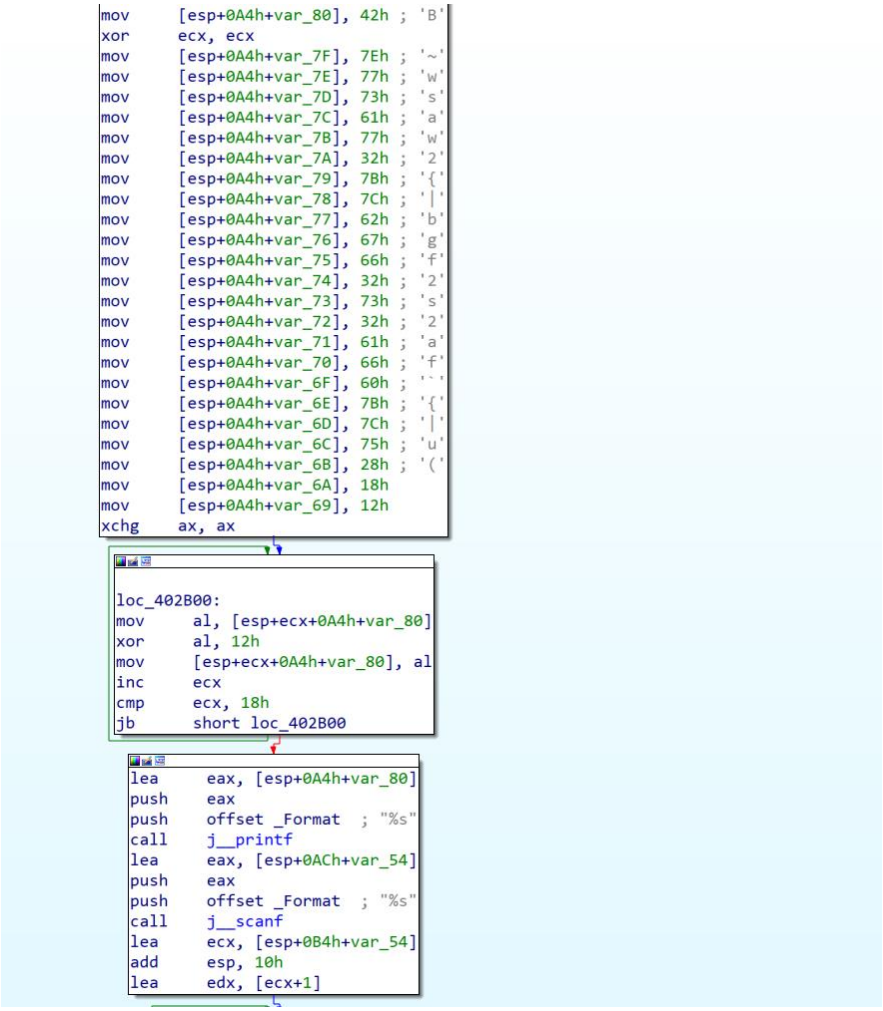

图 2　task3.exe 反汇编代码的图形化显示

2. 使用 IDA 的反编译功能（F5 快捷键）得到伪代码，如图 3 所示。

```
int __cdecl main()
{
  unsigned int v0; // ecx
  unsigned int v1; // ecx
  const char *v2; // eax
  int v3; // edx
  unsigned int v4; // ecx
  unsigned int v6; // ecx
  char v7[8]; // [esp+30Ch] [ebp-A4h] BYREF
  char v8; // [esp+314h] [ebp-9Ch] BYREF
  char v9[8]; // [esp+315h] [ebp-9Bh] BYREF
  char v10; // [esp+320h] [ebp-90h] BYREF
  char v11[12]; // [esp+321h] [ebp-8Fh] BYREF
  char v12[24]; // [esp+330h] [ebp-80h] BYREF
  char v13[20]; // [esp+348h] [ebp-68h]
  char v14[80]; // [esp+35Ch] [ebp-54h] BYREF

  qmemcpy(v12, "B~wsaw2{|bgf2s2af`{|u(", 22);
  v0 = 0;
  v12[22] = 24;
  v12[23] = 18;
  do
    v12[v0++] ^= 0x12u;
  while ( v0 < 0x18 );
  j__printf("%s", v12);
  j__scanf("%s", v14);
  if ( strlen(v14) == 20 )
  {
    v13[0] = -15;
    v3 = 0;
    v13[1] = -55;
    v13[2] = -31;
    v13[3] = -1;
    v13[4] = -25;
```

图 3 task3.exe 的反编译伪代码

3. 通过对反汇编命令及反编译伪代码的分析，逆向推理出待输入字符串的计算公式
4. 使用熟悉的编程语言（C++、Java、Python 等）对待输入字符串进行计算，完成逆向分析挑战。

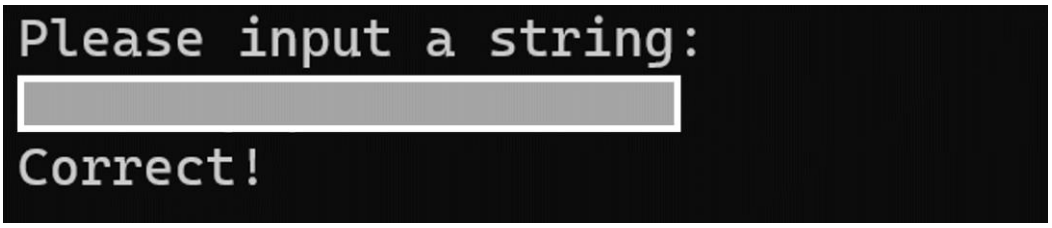

图 4 逆向分析，完成 task3 练习

(二) task4

1. 通过 IDA Freeware 得到 task4.exe 的反汇编代码，如图 5 和图 6 所示。

```
.text:00401470 _main           proc near               ; CODE XREF: _main_0↑j
.text:00401470
.text:00401470 input           = byte ptr -12Ch
.text:00401470 target          = byte ptr -0D8h
.text:00401470 var_6C          = dword ptr -6Ch
.text:00401470 var_68          = dword ptr -68h
.text:00401470 var_64          = dword ptr -64h
.text:00401470 var_60          = dword ptr -60h
.text:00401470 var_5C          = dword ptr -5Ch
.text:00401470 var_58          = word ptr -58h
.text:00401470 var_54          = byte ptr -54h
.text:00401470 var_4           = dword ptr -4
.text:00401470
.text:00401470                 sub     esp, 6Ch
.text:00401473                 mov     eax, ___security_cookie
.text:00401478                 xor     eax, esp
.text:0040147A                 mov     [esp+6Ch+var_4], eax
.text:0040147E                 push    offset _Format  ; "Please input a string:\n"
.text:00401483                 call    j__printf
.text:00401488                 lea     eax, [esp+70h+var_54]
.text:0040148C                 push    eax
.text:0040148D                 push    offset aS       ; "%s"
.text:00401492                 call    j__scanf
.text:00401497                 lea     ecx, [esp+78h+var_54]
.text:0040149B                 add     esp, 0Ch
.text:0040149E                 lea     edx, [ecx+1]
.text:004014A1
.text:004014A1 loc_4014A1:                             ; CODE XREF: _main+36↓j
.text:004014A1                 mov     al, [ecx]
.text:004014A3                 inc     ecx
.text:004014A4                 test    al, al
.text:004014A6                 jnz     short loc_4014A1
```

图 5 task4.exe 的反汇编代码

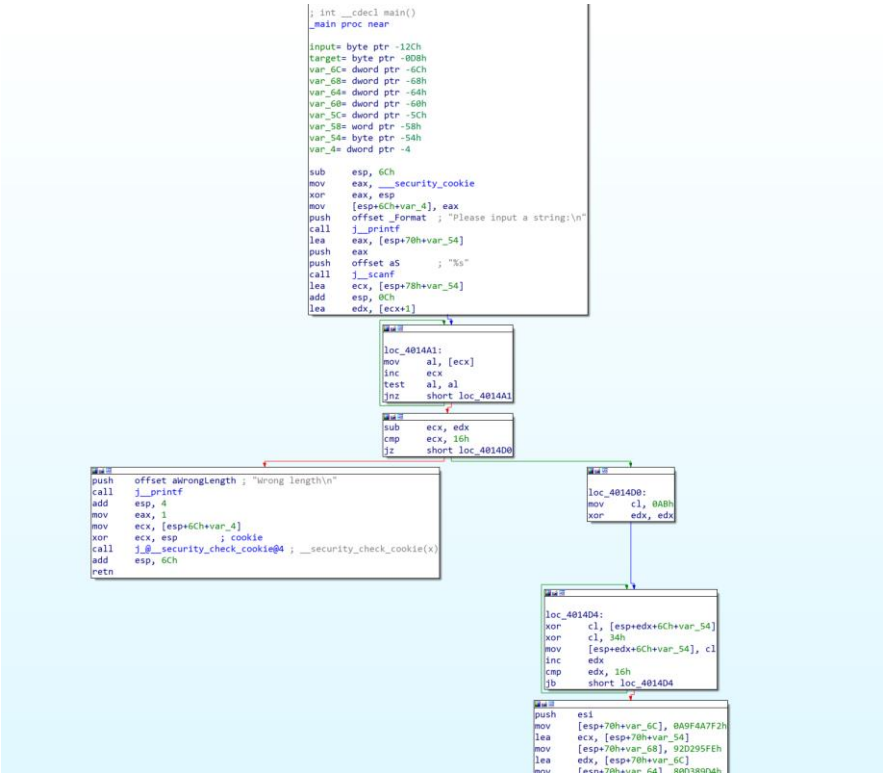

图 6 task4.exe 反汇编代码的图形化显示

2. 使用 IDA 的反编译功能（F5 快捷键）得到伪代码，如图 7 所示。右键点击数字对象可实现数制转换。

```
3  char v1; // cl
4  unsigned int i; // edx
5  char *v3; // ecx
6  int *v4; // edx
7  unsigned int v5; // esi
8  bool v6; // cf
9  int v7[5]; // [esp+C0h] [ebp-6Ch] BYREF
10 __int16 v8; // [esp+D4h] [ebp-58h]
11 char v9[80]; // [esp+D8h] [ebp-54h] BYREF
12
13 j__printf("Please input a string:\n");
14 j__scanf("%s", v9);
15 if ( strlen(v9) == 22 )
16 {
17   v1 = 0xAB;
18   for ( i = 0; i < 0x16; ++i )
19   {
20     v1 ^= v9[i] ^ 0x34;
21     v9[i] = v1;
22   }
23   v7[0] = 0xA9F4A7F2;
24   v3 = v9;
25   v7[1] = 0x92D295FE;
26   v4 = v7;
27   v7[2] = 0x80D389D4;
28   v5 = 0x12;
29   v7[3] = 0xB5E0BCEB;
30   v7[4] = 0xBEE4B5ED;
31   v8 = 0xBCED;
32   while ( *(_DWORD *)v3 == *v4 )
33   {
34     v3 += 4;
35     ++v4;
36     v6 = v5 < 4;
37     v5 -= 4;
38     if ( v6 )
39     {
40       if ( *(_WORD *)v3 == *(_WORD *)v4 )
41       {
42         j__printf("Correct");
43         return 0;
```

图 7 task4.exe 的反编译伪代码

3. 通过对反汇编命令及反编译伪代码的分析，逆向推理出待输入字符串的计算公式
4. 使用熟悉的编程语言（C++、Java、Python 等）对待输入字符串进行计算，完成逆向分析挑战。

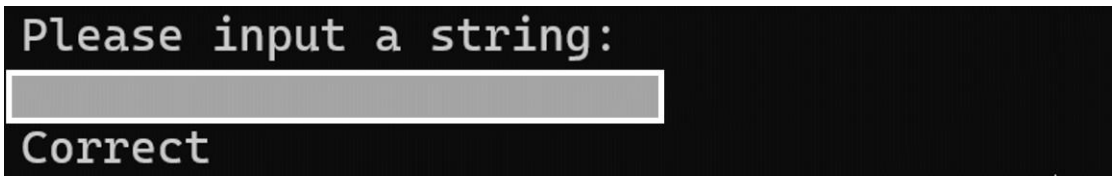

图 8 逆向分析，完成 task4 练习

三、 实验报告

1. 分别针对 task3、task4 使用 IDA Freeware，获得可执行文件的反汇编代码及反编译伪代码，提供截图。
2. 分别针对 task3、task4 反编译伪代码的计算过程、数据结构、条件判断等信息进行逆向分析，列出正确输入字符串的计算公式。
3. 使用熟悉的编程语言，分别针对 task3、task4 编写脚本，计算出正确的字符串，提供脚本输出结果的截图
4. 分别运行程序 task3、task4，输入计算得到的字符串进行验证，获得“Correct”输出，提供截图。