

南開大學

汇编语言与逆向技术课程实验报告

实验七： Reverse Engineering Challenge



学 院____网络空间安全学院____
专 业____信息安全、法学双学位____
学 号____2212000____
姓 名____宋奕纬____
班 级____1061____

一、实验目的

- 1、熟悉静态反汇编工具 IDA Freeware；
- 2、熟悉反汇编代码的逆向分析过程；
- 3、掌握反汇编语言中的数学计算、数据结构、条件判断、分支结构的识别和逆向分析

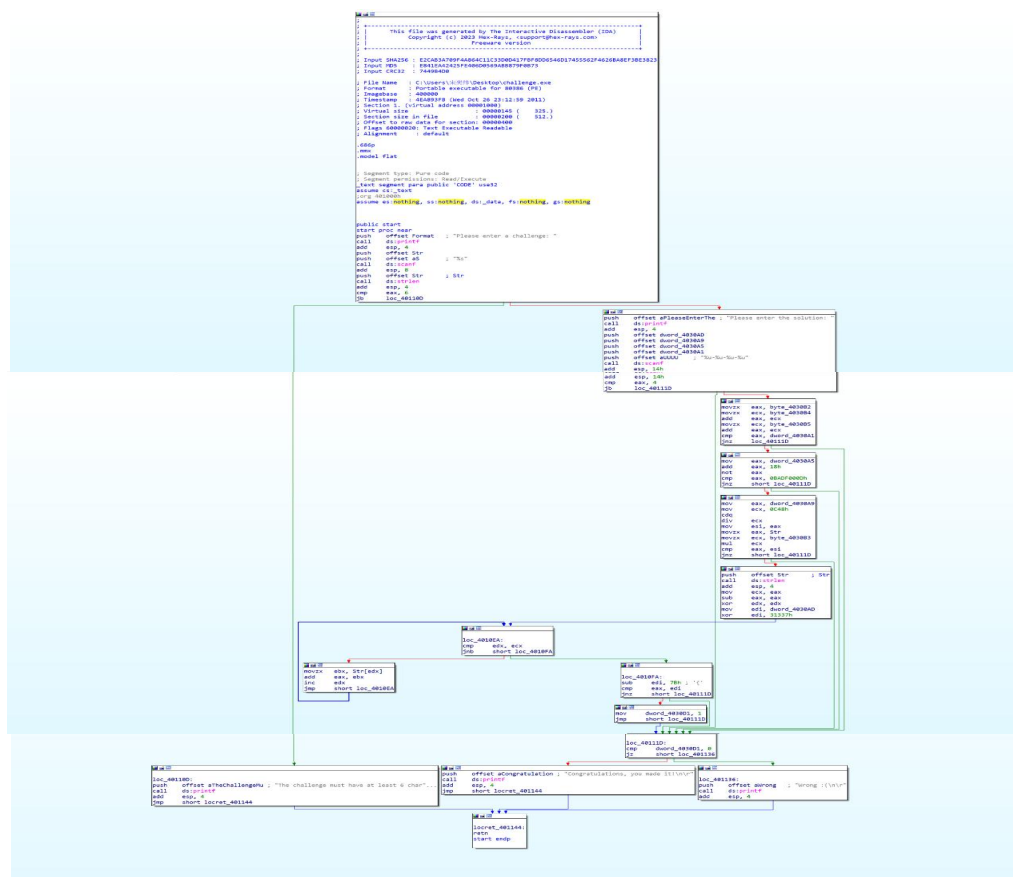
二、实验原理

- 1、实验环境：反汇编工具 IDA Freeware
- 2、实验原理：

通过 IDA 可以得到二进制代码的反汇编代码，利用汇编所学知识对反汇编代码的数学计算、数据结构、条件判断、分支结构进行识别与分析。

三、实验过程

- 1、利用 IDA Freeware 查看反汇编代码（图形界面+文本界面）



```

.text:00401000 ;
.text:00401000 ;
.text:00401000 ; This file was generated by The Interactive Disassembler (IDA)
.text:00401000 ; Copyright (c) 2023 Hex-Rays, <support@hex-rays.com>
.text:00401000 ; Freeware version
.text:00401000 ;
.text:00401000 ;
.text:00401000 ; Input SHA256 : E2CAB3A709F4A864C1C3D0D417F8F8D06546D17455562F4626B8EF3BE3823
.text:00401000 ; Input MD5 : E841EA4252FE406D0569A88879F0873
.text:00401000 ; Input CRC32 : 74689409
.text:00401000 ;
.text:00401000 ; File Name : C:\Users\冰茶\Desktop\challenge.exe
.text:00401000 ; Format : Portable executable for 86386 (PE)
.text:00401000 ; Imagebase : 400000
.text:00401000 ; Timestamp : 4EAB93FB (Wed Oct 26 23:12:59 2011)
.text:00401000 ; Section 1 (virtual address 00001000)
.text:00401000 ; Virtual size : 00000145 ( 325.)
.text:00401000 ; Section size in file : 00000200 ( 512.)
.text:00401000 ; Offset to raw data for section: 00000400
.text:00401000 ; Flags 60000020: Text Executable Readable
.text:00401000 ; Alignment : default
.text:00401000 ;
.text:00401000 ;.686p
.text:00401000 ;.mmx
.text:00401000 ;.model flat
.text:00401000 ;
.text:00401000 ; =====
.text:00401000 ;
.text:00401000 ; Segment type: Pure code
.text:00401000 ; Segment permissions: Read/Execute
.text:00401000 _text segment para public 'CODE' use32
.text:00401000 assume cs:_text
.text:00401000 ;org 401000h
.text:00401000 assume es:nothing, ss:nothing, ds:_data, fs:nothing, gs:nothing

```

```

.text:00401000 public start
.text:00401000 proc near
.text:00401000 push offset Format ; "Please enter a challenge: "
.text:00401005 call ds:printf
.text:00401008 add esp, 4
.text:0040100E push offset Str
.text:00401013 push offset a5 ; "%s"
.text:00401018 call ds:scanf
.text:0040101E add esp, 8
.text:00401021 push offset Str ; Str
.text:00401026 call ds:strlen
.text:0040102C add esp, 4
.text:0040102F cmp eax, 6
.text:00401032 jb loc_401100
.text:00401038 push offset aPleaseEnterThe ; "Please enter the solution: "
.text:0040103D call ds:printf
.text:00401043 add esp, 4
.text:00401046 push offset dword_403040
.text:0040104B push offset dword_403040
.text:00401050 push offset dword_403045
.text:00401055 push offset dword_403041
.text:0040105A push offset aUUU ; "U-U-U-U-U"
.text:0040105F call ds:scanf
.text:00401065 add esp, 14h
.text:00401068 cmp eax, 4
.text:0040106B jb loc_401110
.text:00401071 movzx eax, byte_403082
.text:00401078 movzx ecx, byte_403084
.text:0040107F add eax, ecx
.text:00401081 movzx ecx, byte_403085
.text:00401088 add eax, ecx
.text:0040108A cmp eax, dword_4030A1
.text:00401090 jnz loc_40111D
.text:00401096 mov eax, dword_4030A5
.text:0040109B add eax, 18h
.text:0040109E not eax
.text:004010A0 cmp eax, 0BADF000h
.text:004010A5 jnz short loc_40111D
.text:004010A7 mov eax, dword_4030A9
.text:004010AC mov ecx, 8C4h
.text:004010B1 cdq
.text:004010B2 div ecx
.text:004010B4 mov esi, eax
.text:004010B6 movzx eax, Str
.text:004010BD movzx ecx, byte_403083
.text:004010C4 mul ecx
.text:004010C6 cmp eax, esi
.text:004010CB jnz short loc_40111D
.text:004010CA push offset Str ; Str
.text:004010CF call ds:strlen
.text:004010D5 add esp, 4
.text:004010D8 mov ecx, eax

```

```

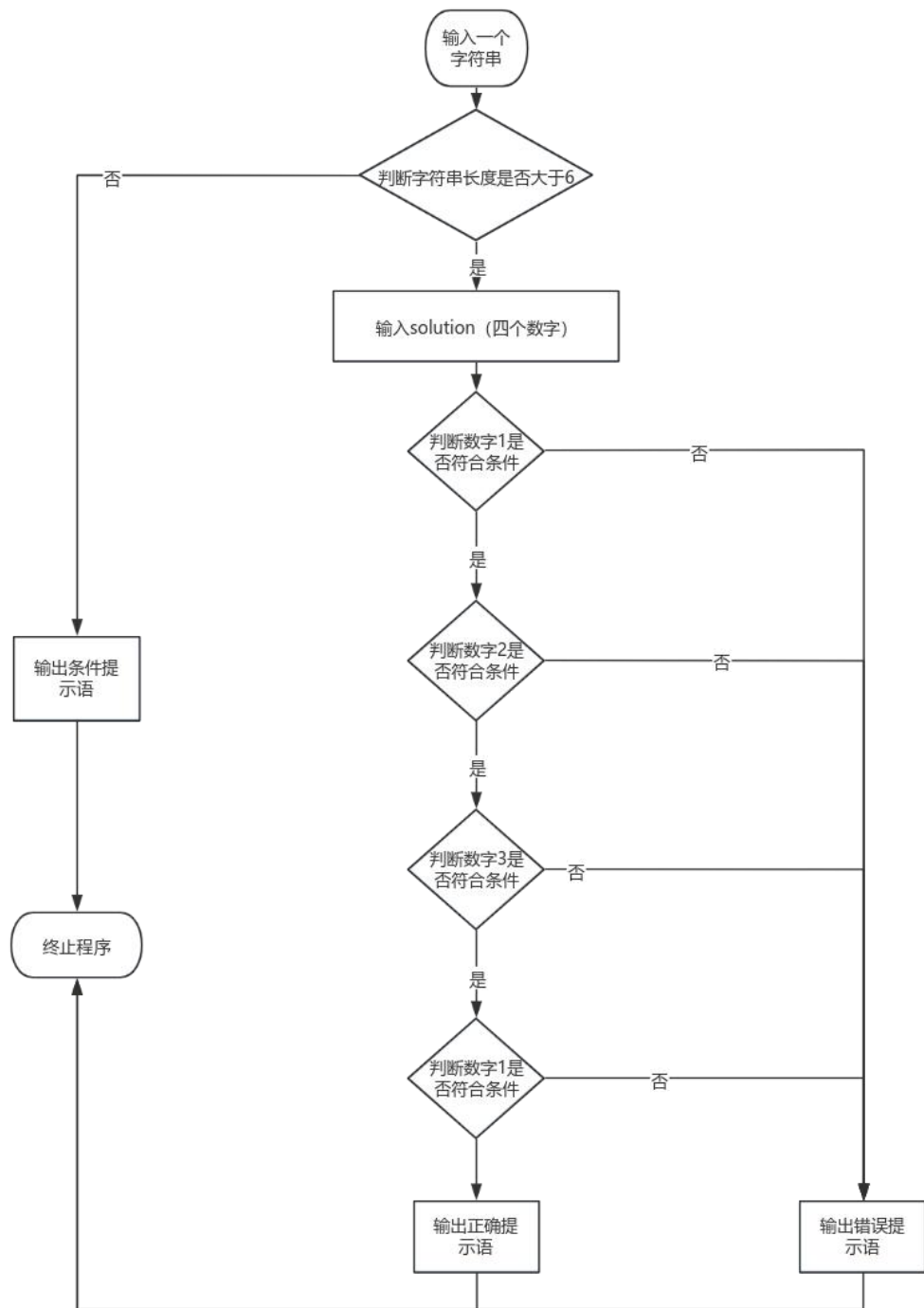
.text:004010D5 add esp, 4
.text:004010D8 mov ecx, eax
.text:004010DA sub eax, eax
.text:004010DC xor edx, edx
.text:004010DE mov edi, dword_4030AD
.text:004010E4 xor edi, 31337h
.text:004010EA
.text:004010EA loc_4010EA: ; CODE XREF: start+FB1j
.text:004010EA cmp edx, ecx
.text:004010EC jnb short loc_4010FA
.text:004010EE movzx ebx, Str[edx]
.text:004010F5 add eax, ebx
.text:004010F7 inc edx
.text:004010F9 jmp short loc_4010EA
.text:004010FA
.text:004010FA loc_4010FA: ; CODE XREF: start+ECTj
.text:004010FA sub edi, 79h ; '['
.text:004010FD cmp eax, edi
.text:004010FF jnz short loc_40111D
.text:00401101 mov dword_403001, 1
.text:00401108 jmp short loc_40111D
.text:0040110D
.text:0040110D loc_40110D: ; CODE XREF: start+321j
.text:0040110D push offset aTheChallengeMu ; "The challenge must have at least 6 char..."
.text:00401112 call ds:printf
.text:00401118 add esp, 4
.text:0040111B jmp short locret_401144
.text:0040111D
.text:0040111D loc_40111D: ; CODE XREF: start+6B1j ; start+901j ...
.text:0040111D cmp dword_403001, 0
.text:00401124 jz short loc_401136
.text:00401126 push offset aCongratulation ; "Congratulations, you made it!\n"
.text:0040112B call ds:printf
.text:00401131 add esp, 4
.text:00401136 jmp short locret_401144
.text:00401136
.text:00401136 loc_401136: ; CODE XREF: start+1241j
.text:00401136 push offset aWrong ; "Wrong :{\n"
.text:0040113B call ds:printf
.text:00401141 add esp, 4
.text:00401144
.text:00401144 locret_401144: ; CODE XREF: start+11B1j ; start+1341j
.text:00401144 retn
.text:00401144 start endp

```

00000546 00401144: start:locret_401144 (Synchronized with Hex View-1)

1

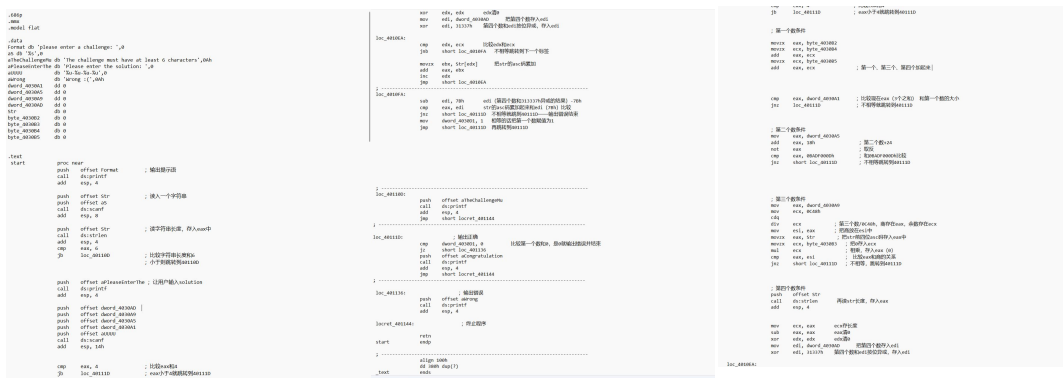
2、分析总体结构，做出流程图



3、细节分析

(1) 将代码改写成熟悉的汇编代码形式

(此处仅放总览图，接下来为详细分析)



(略去库的调用等部分)

(2) 数据段

.data

;提示语定义

Format db 'please enter a challenge: ',0

aTheChallengeMu db 'The challenge must have at least 6 characters',0Ah

aPleaseEnterThe db 'Please enter the solution: ',0

aWrong db 'Wrong :(',0Ah

;solution 的相关定义

aUUUUU db '%u-%u-%u-%u',0

dword_4030A1 dd 0

dword_4030A5 dd 0

dword_4030A9 dd 0

dword_4030AD dd 0

;字符串定义

Str db 0

byte_4030B2 db 0

byte_4030B3 db 0

byte_4030B4 db 0

byte_4030B5 db 0

(3) 代码段

①字符串、解决方案的读取

```
push    offset Format          ; 输出提示语
call    ds:printf
add     esp, 4

push    offset Str             ; 读入一个字符串
push    offset aS
call    ds:scanf
add     esp, 8

push    offset Str             ; 读字符串长度，存入 eax 中
call    ds:strlen
add     esp, 4
cmp     eax, 6
jb      loc_40110D              ; 比较字符串长度和 6
                                   ; 小于则跳转到 40110D，输出信息，终止程序

push    offset aPleaseEnterThe ; 让用户输入 solution
call    ds:printf
add     esp, 4
push    offset dword_4030AD
push    offset dword_4030A9
push    offset dword_4030A5
push    offset dword_4030A1
push    offset aUUUU
call    ds:scanf
add     esp, 14h
```

②对 solution 的判断

A. 对第一个数的判断

```
movzx    eax, byte_4030B2
movzx    ecx, byte_4030B4
add      eax, ecx
movzx    ecx, byte_4030B5
add      eax, ecx      ; 第一个、第三个、第四个加起来
cmp      eax, dword_4030A1      ; 比较现在 eax 和第一个数
jnz      loc_40111D      ; 不相等就跳转到 40111D, 输出
                        错误并终止程序
```

故知道第一个数等于字符串第 1、3、4 位 ascII 码之和

B. 对第二个数的判断

```
mov      eax, dword_4030A5
add      eax, 18h      ; 第二个数+24
not      eax      ; 取反
cmp      eax, 0BADF000Dh      ; 和 0BADF000Dh 比较
jnz      short loc_40111D      ; 不相等就跳转到 40111D, 输出
                        错误并终止程序
```

故第二个数为 0BADF000Dh 取反减去 18h

C. 对第三个数的判断

```
mov      eax, dword_4030A9
mov      ecx, 0C48h
cdq
div      ecx      ; 第三个数/0C48h, 商存在 eax, 余数存在 ecx
mov      esi, eax      ; 把商放在 esi 中
movzx    eax, Str      ; 把 str 第 0 位 ascII 码存入 eax 中
movzx    ecx, byte_4030B3      ; 把 str 第 2 位 ascII 存入 ecx 中
```



```

mul     ecx                ; 相乘，存入 eax
cmp     eax, esi           ; 比较 eax 和商的关系
jnz     short loc_40111D   ; 不相等，跳转到 40111D

```

故第三个数为字符串第 0、2 位的 ascII 码之乘积乘 0C48h

D. 对第四个数的判断

```

push    offset Str
                call    ds:strlen      再读 str 长度，存入 eax
                add     esp, 4

                mov     ecx, eax        ecx 存长度
                sub     eax, eax        eax 清 0
                xor     edx, edx        edx 清 0
                mov     edi, dword_4030AD ; 把第四个数存入
                                           edi
                xor     edi, 31337h

loc_4010EA:    ; 通过循环把 str 的 asc 码累加
                cmp     edx, ecx        ; 比较 edx 和 ecx
                jnb     short loc_4010FA ; 不相等跳转到下一个标签
                movzx   ebx, Str[edx]
                add     eax, ebx
                inc     edx
                jmp     short loc_4010EA

loc_4010FA:
                sub     edi, 7Bh
                cmp     eax, edi
                jnz     short loc_40111D
                mov     dword_4030D1, 1 ; 相等的话把第一个数赋值为 1
                jmp     short loc_40111D ; 再跳转到 40111D

```

故第四个数为（字符串的 ascII 码之和+7Bh）和 31337h 按位异或

E. 别的标签

```
loc_40110D:                                ; 输出提示信息（至少六位）
    push    offset aTheChallengeMu
    call    ds:printf
    add     esp, 4
    jmp     short locret_401144

loc_40111D:                                ; 输出正确
    cmp     dword_4030D1, 0                ; 比较第一个数和 0，
                                           ; 是 0 就输出错误并结束，
                                           ; 故在第四个数判断的末
                                           ; 尾把第一个数再赋值为 1
    jz      short loc_401136
    push    offset aCongratulation
    call    ds:printf
    add     esp, 4
    jmp     short locret_401144

loc_401136:                                ; 输出错误
    push    offset aWrong
    call    ds:printf
    add     esp, 4

locret_401144:                             ; 终止程序
    retn
```

4、计算（用 C++ 计算四个数的值）

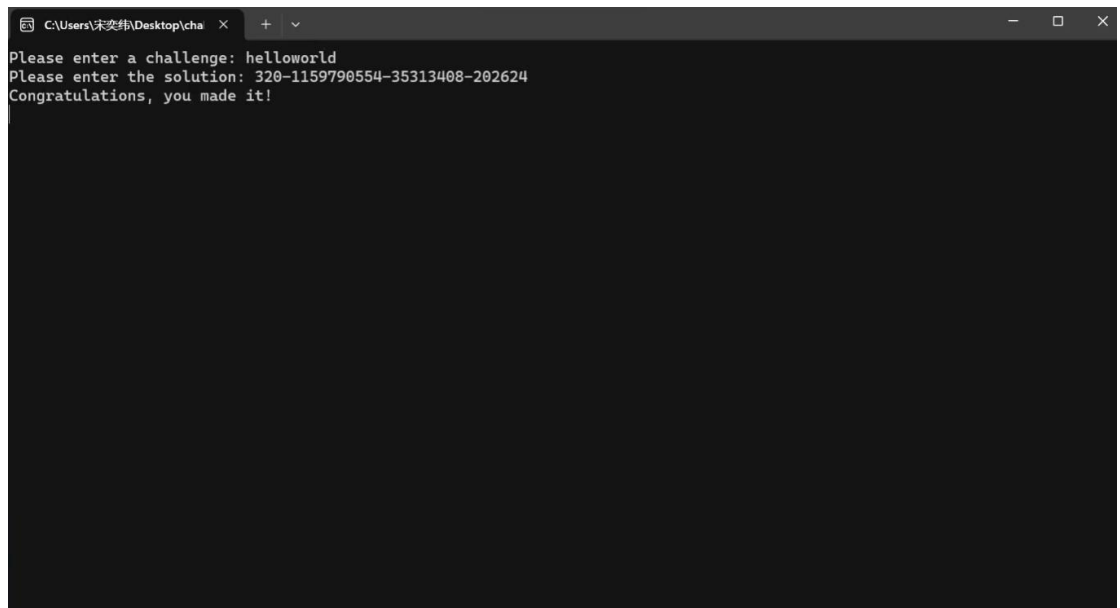
```
1
2 #include <iostream>
3 #include <string>
4
5 using namespace std;
6
7 int main() {
8     // 读入字符串
9     string str;
10    cout << "请输入至少6个字符的字符串: ";
11    cin >> str;
12
13    // 计算第一个数
14    int sum1 = str[0] + str[2] + str[3];
15
16    // 计算第二个数
17    int num2 = 0xABADF00D;
18    int result2 = ~(num2) ^ 0x18;
19
20    // 计算第三个数
21    int product3 = (str[0] * str[2]) * 0xC48;
22
23    // 计算第四个数
24    int sum4 = 0;
25    for (char c : str) {
26        sum4 += c;
27    }
28    int result4 = (sum4 + 0x7B) ^ 0x31337;
29
30    // 输出结果
31    cout << sum1 << "-" << result2 << "-" << product3 << "-" << result4 << endl;
32
33    return 0;
34 }
```

以字符串 helloworld 为例

```
Microsoft Visual Studio 调试
请输入至少6个字符的字符串: helloworld
320-1159790554-35313408-202624

D:\cpp oj\ConsoleApplication15\x64\Debug\ConsoleApplication15.exe (进程 17392)已退出, 代码为 0。
按任意键关闭此窗口。 . . |
```

5、测试结果

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Users\宋奕华\Desktop\cha'. The window contains the following text: 'Please enter a challenge: helloworld', 'Please enter the solution: 320-1159790554-35313408-202624', and 'Congratulations, you made it!'. The background is black, and the text is white.

```
C:\Users\宋奕华\Desktop\cha>
Please enter a challenge: helloworld
Please enter the solution: 320-1159790554-35313408-202624
Congratulations, you made it!
```

得到正确结果（但是一闪而过）

四、实验结论及心得体会

- 1、熟悉静态反汇编工具 IDA Freeware。
- 2、熟悉反汇编代码的逆向分析过程。
- 3、掌握反汇编语言中的数学计算、数据结构、条件判断、分支结构的识别和逆向分析。
- 4、对汇编语言有了更深入的了解。
- 5、提升了解决问题的能力，学会多种编程语言综合使用。