汇编语言与逆向技术实验报告

Lab6 Reverse Engineering Challenge

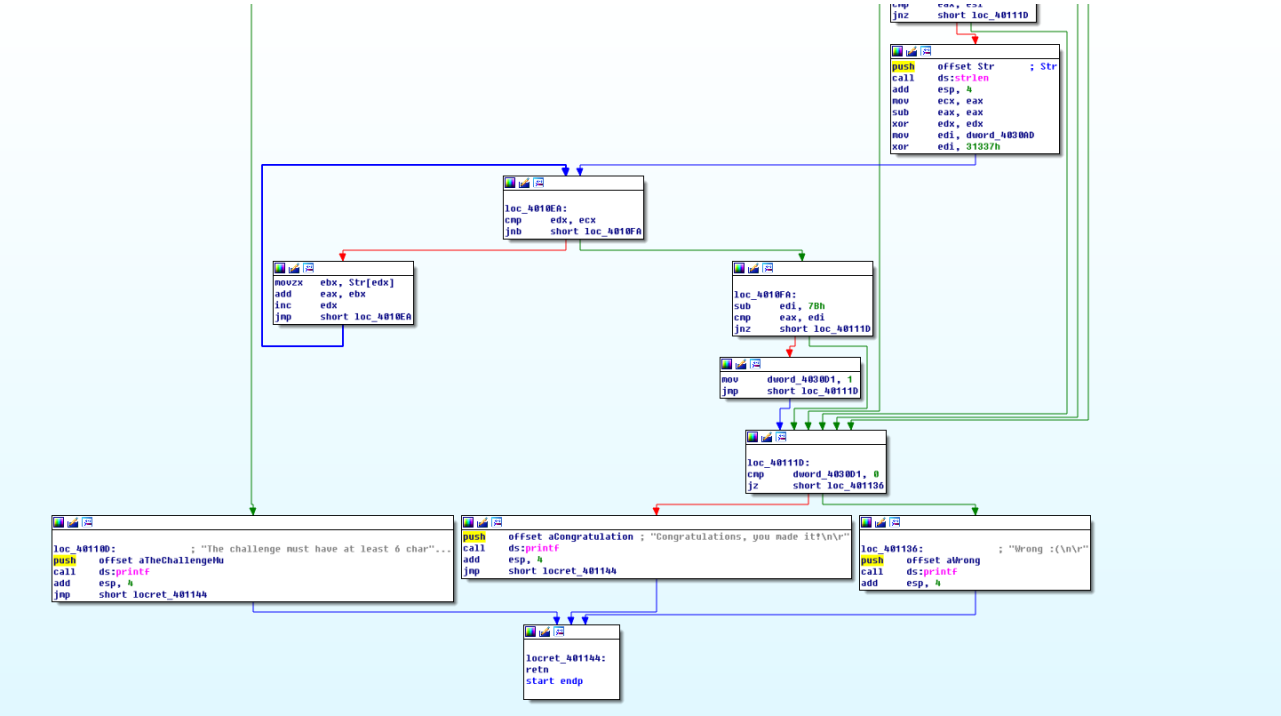
姓名：丁彦添 学号：**1911406**

IDA Pro 的使用，获得二进制代码的反汇编代码

反汇编代码的文本展示

```
.text:00401000 ; =============== S U B R O U T I N E =======================================
.text:00401000
.text:00401000
.text:00401000                 public start
.text:00401000 start           proc near
.text:00401000                 push    offset Format   ; "Please enter a challenge: "
.text:00401005                 call    ds:printf
.text:0040100B                 add     esp, 4
.text:0040100E                 push    offset Str
.text:00401013                 push    offset aS       ; "%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_40110D
.text:00401038                 push    offset aPleaseEnterThe ; "Please enter the solution: "
.text:0040103D                 call    ds:printf
.text:00401043                 add     esp, 4
.text:00401046                 push    offset dword_4030AD
.text:0040104B                 push    offset dword_4030A9
.text:00401050                 push    offset dword_4030A5
.text:00401055                 push    offset dword_4030A1
.text:0040105A                 push    offset aUUUU    ; "%u-%u-%u-%u"
.text:0040105F                 call    ds:scanf
.text:00401065                 add     esp, 14h
.text:00401068                 cmp     eax, 4
.text:0040106B                 jb      loc_40111D
.text:00401071                 movzx   eax, byte_4030B2
.text:00401078                 movzx   ecx, byte_4030B4
.text:0040107F                 add     eax, ecx
.text:00401081                 movzx   ecx, byte_4030B5
.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, 0BADF000Dh
.text:004010A5                 jnz     short loc_40111D
.text:004010A7                 mov     eax, dword_4030A9
.text:004010AC                 mov     ecx, 0C48h
.text:004010B1                 cdq
.text:004010B2                 div     ecx
.text:004010B4                 mov     esi, eax
.text:004010B6                 movzx   eax, Str
.text:004010BD                 movzx   ecx, byte_4030B3
.text:004010C4                 mul     ecx
.text:004010C6                 cmp     eax, esi
00000400 0000000000401000: start
```

反汇编代码的图形化展示



逆向分析二进制代码

- 观察该程序的总体结构图可以发现，这个程序首先判断输入的 challenge 是否长于 6 个字符，如果不长于 6 个字符，则输出“ The challenge must have at least 6 char”。
- 该程序的solution用“-”分隔进行输入，分成四次判断，如果有一组数据错误，则立马输出“Wrong :(”。
- 首先判断的 dword_4030A1 内储存的数据的正确性。通过

  ```
  cmp     eax, dword_4030A1
  ```

  ```
  jnz     loc_40111D
  ```

  这两条语句可以发现如果eax= dword_4030A1，那么就比较下一组数据，所以想知道dword_4030A1 的值，只需要求出eax 里存放的数据的大小。阅读前面的代码可以发现eax 里存放的是challenge 的第2、4、5位数字的ASCII码值之和，所以solution 的第一块就是challenge 的第2、4、5位数字的ASCII码值之和。
- 通过

  ```
  cmp     eax, 0BADF000Dh
  ```

  ```
  jnz     short loc_40111D
  ```

  这两条语句可以发现要想比较下一组数据，则eax 必须等于0BADF000Dh，在这之上还对eax做过 not 操作和加上了18h，因此进行逆运算，对0BADF000Dh先取反，在减去18h，就得到了solution的第二组数据，这组数据与输入无关，是不变的。

- 通过

  ```
  cmp     eax, esi
  jnz     short loc_40111D
  ```

  这两条语句可以发现eax 必须等于esi 才能进行下一组比较。eax=challenge的第一位*challeng的第三位。然后发现esi=eax/C48h，所以eax=esi*C48h。

- 可以发现第四组数据的比较是一个循环，它的循环次数是输入的字符串长度，循环体里进行的是加法操作，即把所有字符对应的ASCII 码求和，放入eax中。当循环结束时，程序对edi减去了7Bh，然后比较eax 和edi 的值，因此减完之后的edi=eax，此时将eax里的值加上7Bh，再与31337h 进行异或运算就可以求得solution 的第四组值。

运行程序，获得成功的结果

```
D:\dyt\Studie\5Junior\D_Assemblersprache\Zuordnung\lab6>challenge.exe
Please enter a challenge: 123456
Please enter the solution: 155-1159790554-7856856-201351
Congratulations, you made it!
```

```
D:\dyt\Studie\5Junior\D_Assemblersprache\Zuordnung\lab6>challenge.exe
Please enter a challenge: 000000
Please enter the solution: 144-1159790554-7243776-201388
Congratulations, you made it!
```