

## 信息安全作业

班级： 物联 1601 姓名： 尹达恒 学号： 1030616134

### 一、 题目要求

8. 请从本书网站上下载文件 `mystery.zip`，并提取出其中的 Windows 可执行程序 `mystery.exe`。正如上面思考题 7 的 e 中提到的，该程序中包括了一部分代码，用于生成与任何有效的用户名相对应的有效序列号。这样的一种算法也被称为密钥发生器，或者可以简单地叫做 `keygen`。如果 Trudy 有 `keygen` 算法的可运行拷贝，那么她就可以生成无数多的有效“用户名/序列号”值对。原则上，对于 Trudy 来说，对 `keygen` 算法进行分析，再完全从零开始编写她自己的(功能上等价的)独立的 `keygen` 程序也是有可能的。但是通常来说，`keygen` 算法都非常复杂，这使得此类攻击在实践中很难操作。不过，也不至于一无所获(至少从 Trudy 的角度来看)，从程序中“剥离出”这种 `keygen` 算法往往是有可能的，而且相对来说还比较简单。这意思是说，攻击者可以提取出代表 `keygen` 算法的汇编代码，再将其直接嵌入到使用 C 语言编写的程序中，这样就创建了独立的 `keygen` 应用程序，同时还不需要去理解 `keygen` 算法的具体细节。

- a. 请从程序 `mystery.exe` 中剥离出 `keygen` 算法，也就是说，请提取出 `keygen` 的汇编代码，再直接将其应用到你自己的独立 `keygen` 程序中。你的程序必须能够接收任意有效的用户名作为输入并输出相应的有效序列号。提示：在 Visual C++ 中，通过使用汇编指令，可以将汇编代码直接嵌入到 C 语言程序中。你可能需要初始化特定寄存器的值，以便剥离出来的代码能够正确运行。
- b. 请使用 a 中编写的程序，为用户名 `markkram` 生成序列号，并通过将之在原始的 `mystery.exe` 程序中进行测试，以验证生成的序列号的正确性。

## 二、 解答过程

- 1、下载并安装反编译工具 IDA;
- 2、使用 IDA 工具反编译所给的 mystery.exe 文件（如下图）;

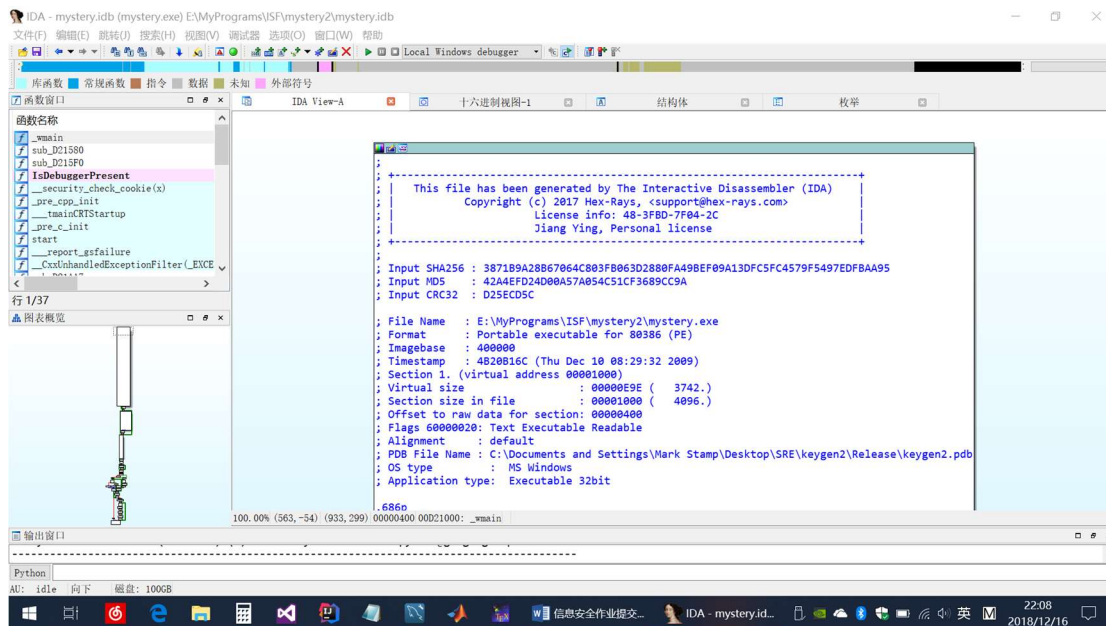


图 1 反编译结果

- 3、按快捷键 F5 生成并查看程序伪代码（如下图）;

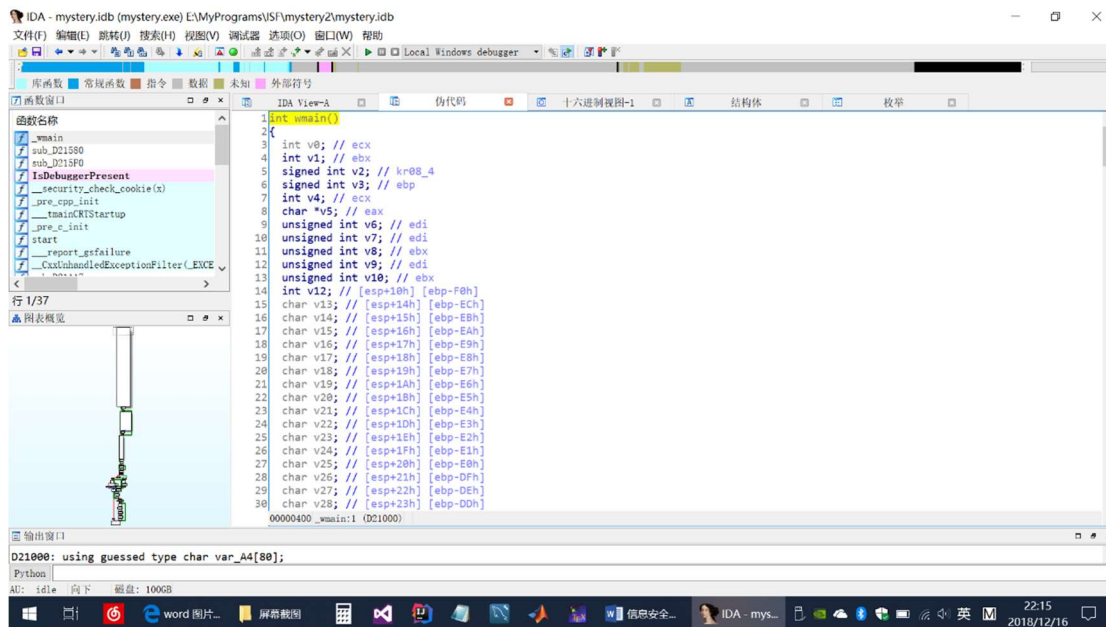


图 2 反编译伪代码

#### 4、启动调试器分析主函数反编译代码运行流程：

(1) 程序开头输出了第一段字符串“Enter username”，然后读入了一个字符串存储在变量 v83 中（如下图）；

```

157 printf("%c", 10);
158 printf("%c", 69);
159 printf("%c", 110);
160 printf("%c", 116);
161 printf("%c", 101);
162 printf("%c", 114);
163 printf("%c", 32);
164 printf("%c", 117);
165 printf("%c", 115);
166 printf("%c", 101);
167 printf("%c", 114);
168 printf("%c", 110);
169 printf("%c", 97);
170 printf("%c", 109);
171 printf("%c", 101);
172 printf("%c", 10);
173 scanf("%s", v83);

```

图 3 输出第一段字符串，读入 v83 的反编译代码

(2) 随后程序判断 v83 中的字符串长度是否小于 5 个字符，如果小于 5 个字符则输出字符串“User name must be at least 5 characters”和“Enter username”并读入一个字符串到 v83，直到 v83 中的字符串长度不小于 5 个字符（如下图）；

```

173 scanf("%s", v83);
174 while ( strlen(v83) < 5 )
175 {
176     printf("\nUser name must be at least 5 characters\n\n");
177     printf("%c", 10);
178     printf("%c", 69);
179     printf("%c", 110);
180     printf("%c", 116);
181     printf("%c", 101);
182     printf("%c", 114);
183     printf("%c", 32);
184     printf("%c", 117);
185     printf("%c", 115);
186     printf("%c", 101);
187     printf("%c", 114);
188     printf("%c", 110);
189     printf("%c", 97);
190     printf("%c", 109);
191     printf("%c", 101);
192     printf("%c", 10);
193     scanf("%s", v83);
194 }

```

图 4 使读入 v83 的字符串长度不小于 5 个字符的反编译代码

(3) 随后程序输出了第一段字符串“Enter serial number”，然后读入一个字符串存储在变量 v84 中（如下图）；

```

194 }
195 printf("%c", 10);
196 printf("%c", 69);
197 printf("%c", 110);
198 printf("%c", 116);
199 printf("%c", 101);
200 printf("%c", 114);
201 printf("%c", 32);
202 printf("%c", 115);
203 printf("%c", 101);
204 printf("%c", 114);
205 printf("%c", 105);
206 printf("%c", 97);
207 printf("%c", 108);
208 printf("%c", 32);
209 printf("%c", 110);
210 printf("%c", 117);
211 printf("%c", 109);
212 printf("%c", 98);
213 printf("%c", 101);
214 printf("%c", 114);
215 printf("%c", 10);
216 scanf("%s", &v84);

```

图 5 使读入 v83 的字符串长度不小于 5 个字符的反编译代码

(4) 随后，程序对输入的两个字符串进行处理并使用一个 if 语句将程序导向两个不同输出（如下图）：

```
216 scanf("%s", &v84);
217 v1 = sub_B61580(v0, v83);
218 v2 = strlen(v83);
219 v3 = v2 >> 1;
220 v4 = 0;
221 if ( v2 >> 1 > 0 )
222 {
223     v5 = &v82 + v2;
224     do
225     {
226         if ( v83[v4] != *v5 )
227             break;
228         ++v4;
229         --v5;
230     }
231     while ( v4 < v3 );
232 }
233 v12 = v4;
234 if ( IsDebuggerPresent() > 0 )
235     v12 = v3 + 1;
236 if ( v12 != v3 || v1 != sub_B615F0(&v84) )
237 {
238     v7 = 0;
```

图 6 处理字符串和判断

(5) 当输入错误序列号时，第 236 行 if 语句判断条件为真，程序输出一串错误信息并结束（如下图）。

```
236 if ( v12 != v3 || v1 != sub_B615F0() )
237 {
238     v7 = 0;
239     do
240     {
241         v8 = v7 % 3;
242         if ( !(v7 % 3) )
243             printf("%c", *(&v13 + v7) + 86);
244         if ( v8 == 1 )
245             printf("%c", *(&v13 + v7) + 69);
246         if ( v8 == 2 )
247             printf("%c", *(&v13 + v7) + 52);
248         ++v7;
249     }
250     while ( (signed int)v7 < 30 );
251     if ( v12 != v3 )
252         printf(" (or username)");
253     v9 = 30;
254     do
255     {
256         v10 = v9 % 3;
257         if ( !(v9 % 3) )
258             printf("%c", *(&v13 + v9) + 86);
259         if ( v10 == 1 )
260             printf("%c", *(&v13 + v9) + 69);
261         if ( v10 == 2 )
262             printf("%c", *(&v13 + v9) + 52);
263         ++v9;
264     }
265     while ( (signed int)v9 < 43 );
266 }
267 else
```

图 7 输入错误序列号时调用的反编译代码

```
E:\MyPrograms\ISF\mystery2\mystery.exe
Enter username
Stamp
Enter serial number
123456
Error! Incorrect serial number (or username). Try again.
```

图 8 输入错误序列号的程序输出



## 5、主函数结构分析

从对程序运行流程的分析可以看出变量 v83 存储了用户输入的用户名，变量 v84 存储了用户输入的序列号，且序列号生成最可能存在的位置是代码运行流程（4）的处理和判断中（见图 6 处理字符串和判断）。在这一部分代码中，可以看到程序进行了以下步骤：

1. 调用 sub\_B16580 对用户输入的用户名（v83）进行了处理（行 217）；
2. 用一段代码对用户输入的用户名（v83）再次进行处理（行 218-233）；
3. 调用 IsDebuggerPresent() 函数进行判断和处理（行 234-235）；
4. 调用 sub\_B165F0 对用户输入的序列号（v84）进行了处理（行 236）；
5. 对步骤 2 的两个处理结果进行比较（行 236）；
6. 对 sub\_B16580 和 sub\_B165F0 的处理结果进行比较（行 236）；
7. 根据比较结果跳转程序（行 236）。

从程序中可以推测，IsDebuggerPresent() 函数调用之后，行 236 中 if 语句的判断条件必为真，程序必然判断输入序列号错误。所以 IsDebuggerPresent() 可能是一个用于反调试的函数。此外，又程序还可以看出，分析密钥生成程序的关键在于以下几点：

1. 函数 sub\_B16580 的作用；
2. 位于 sub\_B16580 调用和 sub\_B165F0 调用代码之间的 v83 处理程序；
3. 函数 sub\_B165F0 的作用。

## 6、程序片段提取测试分析

（1）在函数窗口找到函数 sub\_B16580 的代码并提取进行测试分析，可以看到此函数接收一个字符串，输出一个整数（如下图）。此函数可能是序列号生成程序；

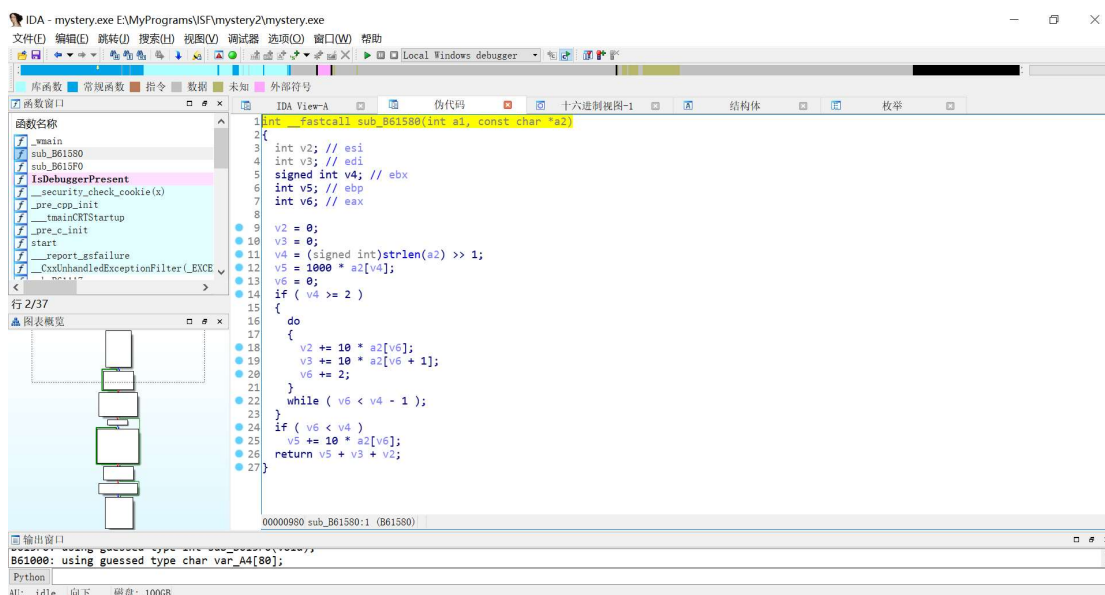


图 9 函数 sub\_B16580 的反编译代码

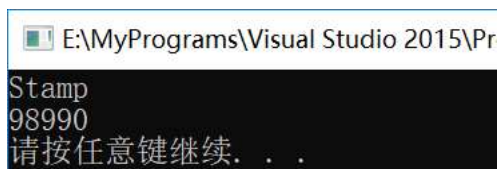


图 10 函数 sub\_B16580 的提取测试结果

(2) 在函数窗口找到函数 sub\_B165F0 的代码并提取进行测试分析，可以看到此函数接收一个数字字符串，输出该字符串对应的整数（如下图）。此函数显然是用于将输入序列号转化为对应整数与 sub\_B16580 进行比较；

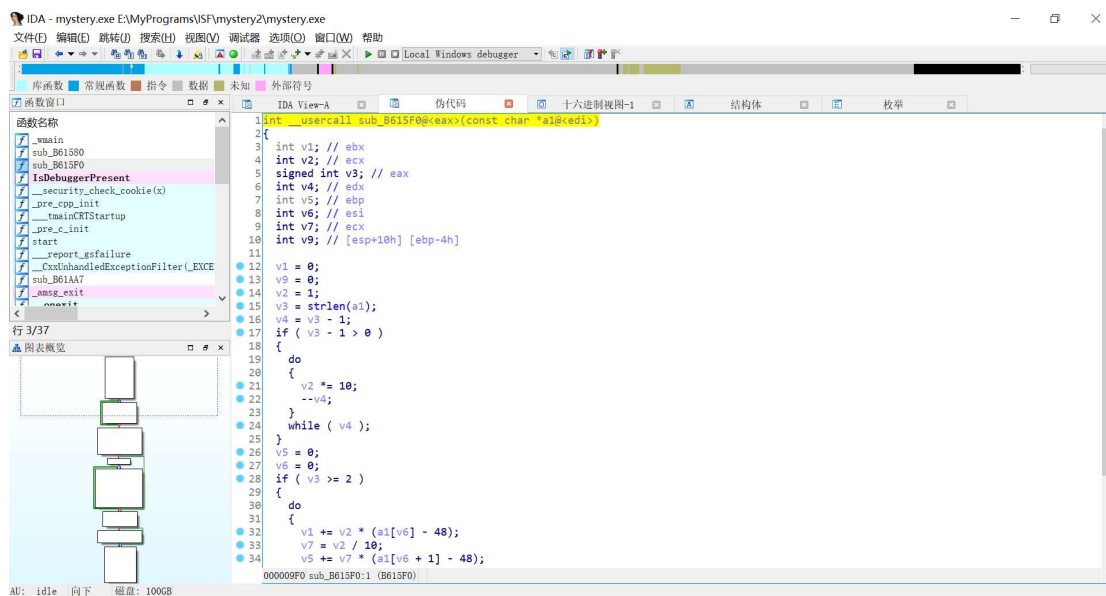


图 11 函数 sub\_B16580 的反编译代码

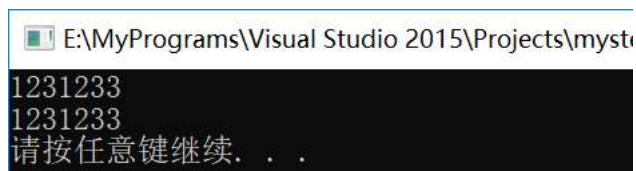
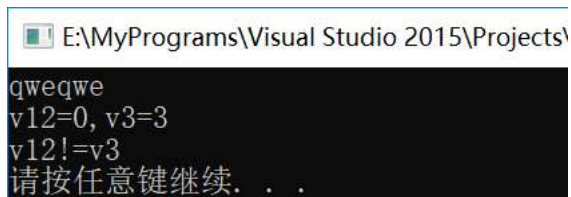


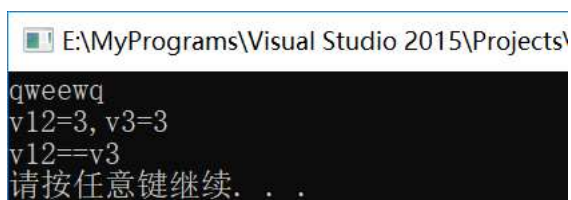
图 12 函数 sub\_B165F0 的提取测试结果

(3) 提取主函数（见图 6 处理字符串和判断）行 218-233 的代码进行测试分析，这一段代码只有在用户输入的用户名（v83）为回文序列时才会使 v12 和 v3 相等。此函数显然是用于判断输入的用户名是否合法；



```
E:\MyPrograms\Visual Studio 2015\Projects\  
qweqwe  
v12=0, v3=3  
v12!=v3  
请按任意键继续. . .
```

图 13 行 218-233 代码的提取测试结果 1（输入非回文）



```
E:\MyPrograms\Visual Studio 2015\Projects\  
qweewq  
v12=3, v3=3  
v12==v3  
请按任意键继续. . .
```

图 14 行 218-233 代码的提取测试结果 2（输入回文）

## 7、构造序列号生成程序

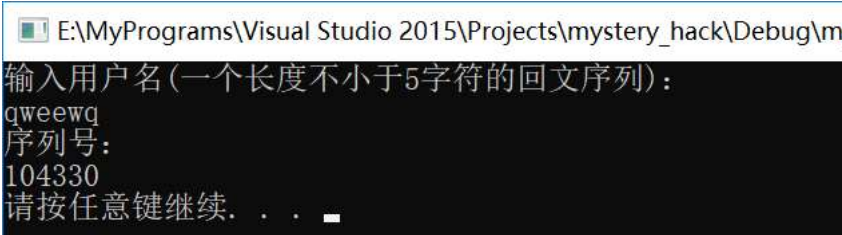
由之前的分析可知，一个合法的用户名-序列号配对应满足如下条件：

1. 用户名长度大于 5 字符；
2. 用户名为回文序列；
3. 序列号为调用函数 sub\_B165F0 对用户名进行处理的结果。

由此构造的序列号生成程序见附件。

### 三、 运行结果

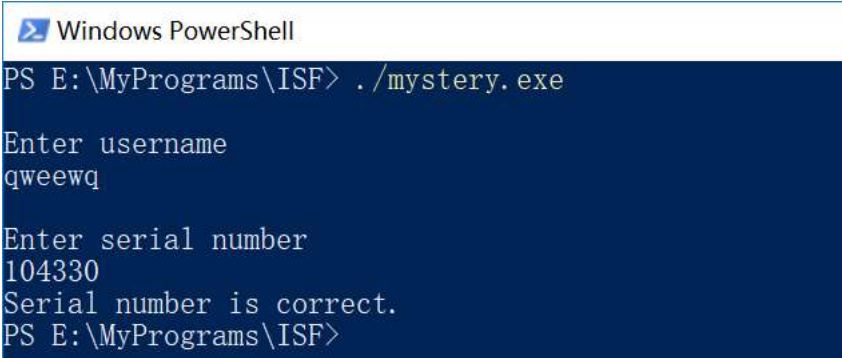
#### 1、生成序列号：



```
E:\MyPrograms\Visual Studio 2015\Projects\mystery_hack\Debug\mystery.exe
输入用户名(一个长度不小于5字符的回文序列):
qweewq
序列号:
104330
请按任意键继续. . .
```

图 15 生成序列号运行结果

#### 2、验证序列号的正确性：



```
Windows PowerShell
PS E:\MyPrograms\ISF> ./mystery.exe

Enter username
qweewq

Enter serial number
104330
Serial number is correct.
PS E:\MyPrograms\ISF>
```

图 16 验证序列号的正确性