xctf攻防世界 REVERSE 高手进阶区 srm-50

原创

l8947943 已于 2022-04-01 09:58:55 修改 250 收藏

分类专栏： 攻防世界reverse之路 文章标签： reverse python字符画 安全小工具

于 2022-04-01 09:44:08 首次发布

本文链接：https://blog.csdn.net/l8947943/article/details/123888895

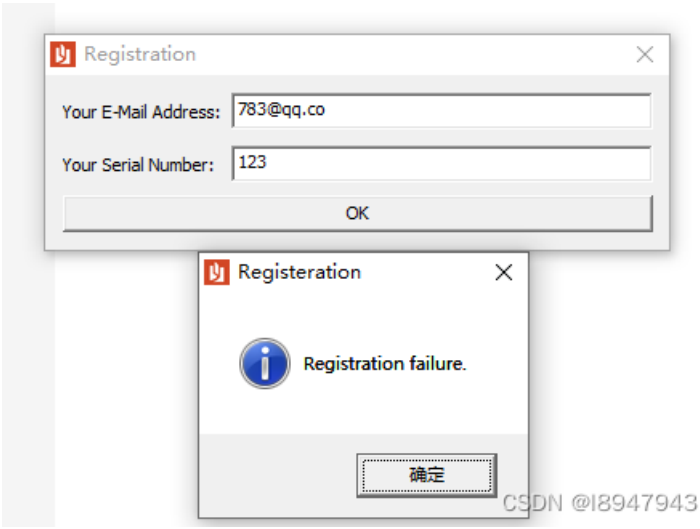
攻防世界reverse之路 专栏收录该内容

14 篇文章 0 订阅

订阅专栏

0x1. 进入环境，下载附件

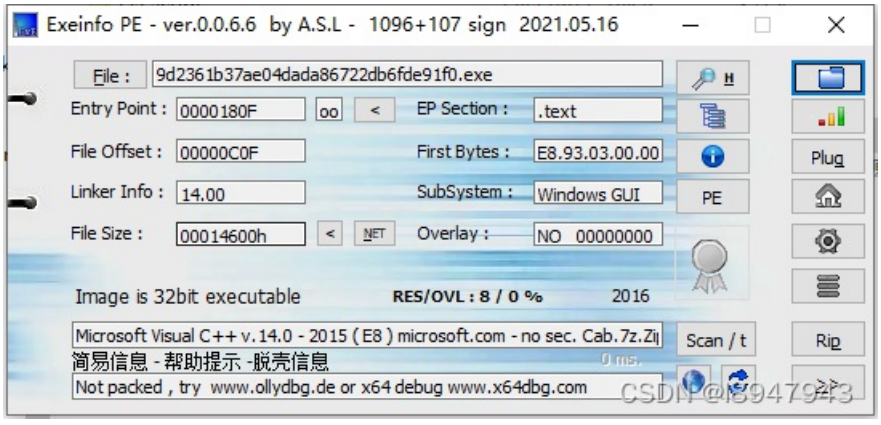
题目给出了一个exe文件，打开运行后，效果如图：



0x2. 问题分析

0x2_1. 检查是否套壳

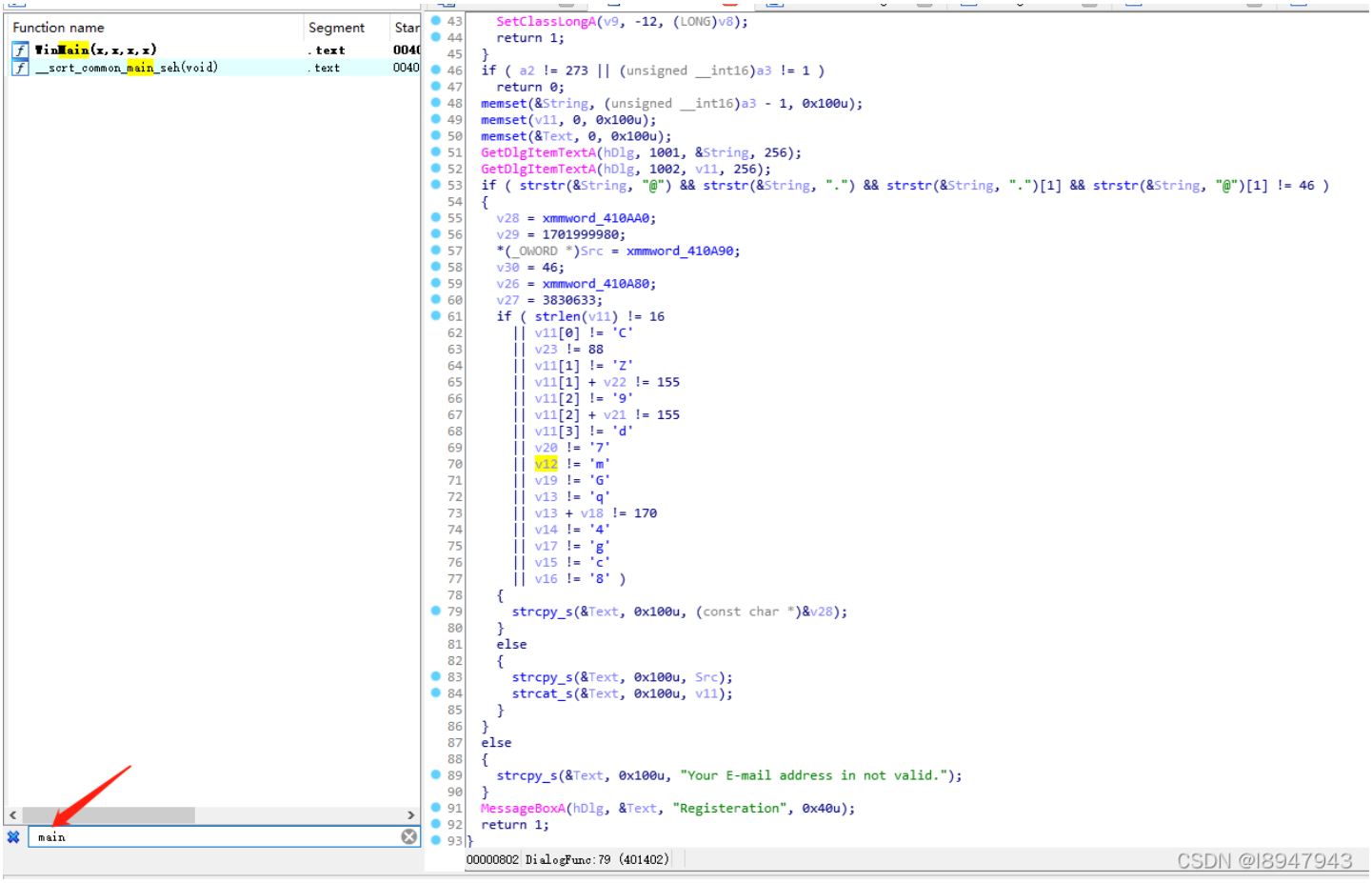
使用exeinfo PE打开文件，如图：



没有任何套壳。

0x2_2. 使用IDA分析

将程序丢入IDA中，搜索main函数，F5反编译得到伪代码，如图：



针对重点代码进行分析：

```
BOOL __stdcall DialogFunc(HWND hDlg, UINT a2, WPARAM a3, LPARAM a4)
{
  if ( strstr(&String, "@") && strstr(&String, ".") && strstr(&String, ".")[1] && strstr(&String, "@")[1] != 46
) // 判断用户输入的邮箱格式是否正确
  {
    v28 = xmmword_410AA0;    // 双击变量，变成字符可以看到 'iaf noitartsigeR'
    v29 = 1701999980;
    *(_OWORD *)Src = xmmword_410A90; // 'cuS noitartsigeR'
    v30 = 46;
    v26 = xmmword_410A80; //  galf ruoY !ssec
    v27 = 3830633;
    if ( strlen(v11) != 16 // 前文定义v11是4位字符，但是此处是需要v11长度为16
      || v11[0] != 'C'
      || v23 != 88
      || v11[1] != 'Z'
      || v11[1] + v22 != 155
      || v11[2] != '9'
      || v11[2] + v21 != 155
      || v11[3] != 'd'
      || v20 != '7'
      || v12 != 'm'
      || v19 != 'G'
      || v13 != 'q'
      || v13 + v18 != 170
      || v14 != '4'
      || v17 != 'g'
      || v15 != 'c'
      || v16 != '8' )
    {
      strcpy_s(&Text, 0x100u, (const char *)&v28); // 如果上述条件满足，则v28变量的内容送到提示框
    }
    else
    {
      strcpy_s(&Text, 0x100u, Src); // 将src提示信息送入提示框
      strcat_s(&Text, 0x100u, v11); // 再将v11的信息粘贴到src后面
    }
  }
  else
  {
    strcpy_s(&Text, 0x100u, "Your E-mail address in not valid.");
  }
  MessageBoxA(hDlg, &Text, "Registeration", 0x40u); // 返回弹框对象
  return 1;
}
```

这里很迷糊的一点就是，明明前面定义v11是char类型的4位长度，结果if语句中判断要求是11位，那么到底是多少？我们双击一下v12变量，看看存放的地址，如图：

```
-00000242                db ? ; undefined
-00000241                db ? ; undefined
-00000240 var_240        db 4 dup(?)
-0000023C var_23C        db ?
-0000023B var_23B        db ?
-0000023A var_23A        db ?
-00000239 var_239        db ?
-00000238 var_238        db ?
-00000237 var_237        db ?
-00000236 var_236        db ?
-00000235 var_235        db ?
-00000234 var_234        db ?
-00000233 var_233        db ?
-00000232 var_232        db ?
-00000231 var_231        db ?
-00000230                db ? ; undefined
-0000022F                db ? ; undefined
```


var240地址是v11，存放4个地址位后，v12放在的var23C位置，那么相当于输入多余的内容会放入后续的，v12，v13，v14…等地址位置，这样看就一目了然了，将if的内容转换成字符：

```
if ( strlen(v11) != 16
  || v11[0] != 'C'
  || v23 != 88
  || v11[1] != 'Z'
  || v11[1] + v22 != 155
  || v11[2] != '9'
  || v11[2] + v21 != 155
  || v11[3] != 'd'
  || v20 != '7'
  || v12 != 'm'
  || v19 != 'G'
  || v13 != 'q'
  || v13 + v18 != 170
  || v14 != '4'
  || v17 != 'g'
  || v15 != 'c'
  || v16 != '8' )
```

v11分别为：CZ9d

v12-v20分别为：mq4c8g9G7bAX

因此最终的答案为：`CZ9dmq4c8g9G7bAX`

效果如图：