xctf逆向

原创

v晴耕雨读V 于 2020-10-11 23:00:32 发布 120 收藏

分类专栏： 逆向 文章标签： 安全

本文链接：https://blog.csdn.net/weixin_44342879/article/details/109018888

xctf逆向

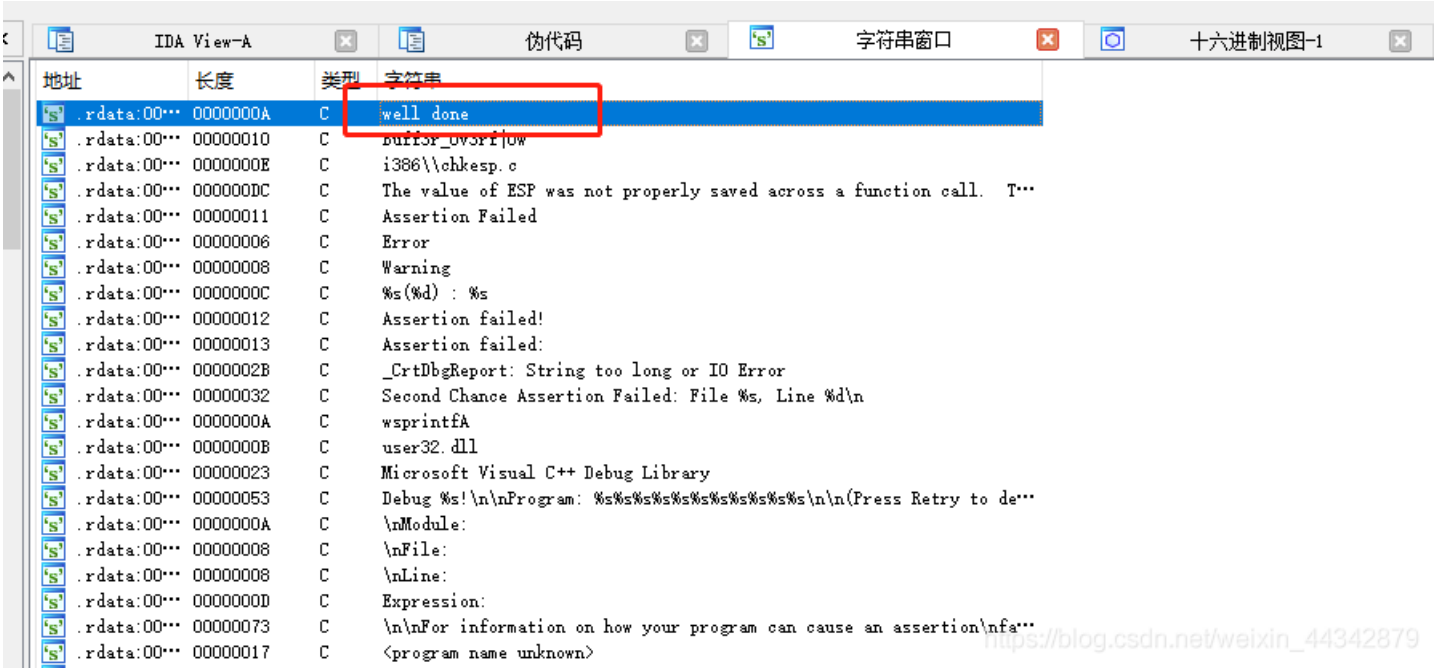
mysterious

下载附件

首先题目给的是一个exe文件，对于这样的一半现在运行一下，结果如下：

利用ida分析

对于这种需要输入口令的一半拖入ida直接分析即可。打开ida后，先查找一下字符串



IDA View-A | 伪代码 | 字符串窗口 | 十六进制视图-1

地址	长度	类型	字符串
.rdata:00…	0000000A	C	well done
.rdata:00…	00000010	C	Buff3r_0v3rf|0w
.rdata:00…	0000000E	C	i386\\chkesp.c
.rdata:00…	000000DC	C	The value of ESP was not properly saved across a function call. T…
.rdata:00…	00000011	C	Assertion Failed
.rdata:00…	00000006	C	Error
.rdata:00…	00000008	C	Warning
.rdata:00…	0000000C	C	%s(%d) : %s
.rdata:00…	00000012	C	Assertion failed!
.rdata:00…	00000013	C	Assertion failed:
.rdata:00…	0000002B	C	_CrtDbgReport: String too long or IO Error
.rdata:00…	00000032	C	Second Chance Assertion Failed: File %s, Line %d\n
.rdata:00…	0000000A	C	wsprintfA
.rdata:00…	0000000B	C	user32.dll
.rdata:00…	00000023	C	Microsoft Visual C++ Debug Library
.rdata:00…	00000053	C	Debug %s!\n\nProgram: %s%s%s%s%s%s%s%s%s%s\n\n(Press Retry to de…
.rdata:00…	0000000A	C	\nModule:
.rdata:00…	00000008	C	\nFile:
.rdata:00…	00000008	C	\nLine:
.rdata:00…	0000000D	C	Expression:
.rdata:00…	00000073	C	\n\nFor information on how your program can cause an assertion\nfa…
.rdata:00…	00000017	C	<program name unknown>

看到有well done字符串，找到其中的交叉引用函数，进行F5分析

```
int __stdcall sub_401090(HWND hWnd, int a2, int a3, int a4)
{
  char v5; // [esp+50h] [ebp-310h]
  CHAR Text[4]; // [esp+154h] [ebp-20Ch]
  char v7; // [esp+159h] [ebp-207h]
  __int16 v8; // [esp+255h] [ebp-10Bh]
  char v9; // [esp+257h] [ebp-109h]
  int v10; // [esp+258h] [ebp-108h]
  CHAR String; // [esp+25Ch] [ebp-104h] 注意string为char类型，也可以理解为跟下面的v12、v13、v14为一个字符数组
  char v12; // [esp+25Fh] [ebp-101h]
  char v13; // [esp+260h] [ebp-100h]
  char v14; // [esp+261h] [ebp-FFh]

  memset(&String, 0, 0x104u);
  v10 = 0;
  if ( a2 == 16 )
  {
    DestroyWindow(hWnd);
    PostQuitMessage(0);
  }
  else if ( a2 == 273 )
  {
    if ( a3 == 1000 )
    {
      GetDlgItemTextA(hWnd, 1002, &String, 260);// 获取输入，输入的存放在以string为首地址的连续空间里
      strlen(&String);
      if ( strlen(&String) > 6 )                // 判读输入长度，不能超过6，这样一般也就是6
        ExitProcess(0);
      v10 = atoi(&String) + 1;                  // atoi函数为将字符型的数字转换为int型的数字，如果没有的话返回为0，比如char 100，经过此函数就变为int 的100
      if ( v10 == 123 && v12 == 'x' && v14 == 'z' && v13 == 'y' )// 进行一个判断，也就是对输入的一个判断，
                                                //
      {
        strcpy(Text, "flag");
        memset(&v7, 0, 0xFCu);
        v8 = 0;
        v9 = 0;
        _itoa(v10, &v5, 10);
        strcat(Text, "{");
        strcat(Text, &v5);
        strcat(Text, "_");
        strcat(Text, "Buff3r_0v3rf|0w");
        strcat(Text, "}");
        MessageBoxA(0, Text, "well done", 0);
      }
      SetTimer(hWnd, 1u, 0x3E8u, TimerFunc);
    }
    if ( a3 == 1001 )
      KillTimer(hWnd, 1u);
  }
  return 0;
}
```


这样的就知道了输入122xyz，就可以得到flag

手撕得到flag

```
v10 = atoi(&String) + 1;                // atoi函数为将字符型的数字转换为int型的数字，如果没有的话题目为0，比如
if ( v10 == 123 && v12 == 'x' && v14 == 'z' && v13 == 'y' )// 进行一个判断，也就是对输入的一个判断，
                                        //
{
  strcpy(Text, "flag");
  memset(&v7, 0, 0xFCu);
  v8 = 0;
  v9 = 0;
  _itoa(v10, &v5, 10);
  strcat(Text, "{");
  strcat(Text, &v5);
  strcat(Text, "_");
  strcat(Text, "Buff3r_0v3rf|0w");
  strcat(Text, "}");
  MessageBoxA(0, Text, "well done", 0);
}
SetTimer(hWnd, 1u, 0x3E8u, TimerFunc);
}
if ( a3 == 1001 )
  KillTimer(hWnd, 1u);
```


如图所示，strcat显示的大概率为flag，这样就只需找到v5是什么就行，v10为123，经过itoa函数后，v5为字符串123，这样就直接得到的了flag