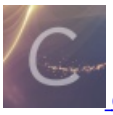
攻防世界Reverse进阶区-Mysterious-writeup

原创

y4ung 于 2020-09-24 23:13:35 发布 530 收藏 2

分类专栏： ctf 文章标签： ctf



1. 介绍

本题是xctf攻防世界中Reverse的进阶区的题Mysterious

题目来源： BUUCTF-2019
题目描述：自从报名了CTF竞赛后，小明就辗转于各大论坛，但是对于逆向题目仍是一知半解。有一天，一个论坛老鸟给小明发了一个神秘的盒子，里面有开启逆向思维的秘密。小明如获至宝，三天三夜，终于解答出来了，聪明的你能搞定这个神秘盒子么？(答案为flag{XXX}形式)

2. 分析

运行，随便输入一个字符串，点击Crack，没反应

查壳：

扔进IDA里看一下。Strings window查看，没找到运行程序时窗口里的字符串。不过看到一个 `well done` ，跟进去发现是在sub_401090函数中被引用。

找到`well done`所在的基本块，条件是`v10 == 123 && v12 == 120 && v14 == 122 && v13 == 121`，注意到在该基本块的开头调用了strcpy函数将`flag`赋值给Text，同时下面也有一些对Text拼接字符串的操作，看起来这个Text保存的就是flag。

观察Text的赋值，除了v5，其它的都是已知的字符，需要解出v5的值。其中，v5是执行itoa函数以后得到的：`_itoa(v10, &v5, 10);` 。

itoa函数能把一个整数转换为字符串：

- v10是需要转换成字符串的数字
- v5是转换后保存字符串的变量
- 10是进制

因此，我们需要把v10解出来。v10等于多少呢？if条件里已经告诉我们了，等于123，因此flag为：`flag{123_Buff3r_0v3rf|0w}`

```
14  memset(&String, 0, 0x104u);
15  v10 = 0;
16  if ( a2 == 16 )
17  {
18    DestroyWindow(hWnd);
19    PostQuitMessage(0);
20  }
21  else if ( a2 == 273 )
22  {
23    if ( a3 == 1000 )
24    {
25      GetDlgItemTextA(hWnd, 1002, &String, 260);
26      strlen(&String);
27      if ( strlen(&String) > 6 )
28        ExitProcess(0);
29      v10 = atoi(&String) + 1;
30      if ( v10 == 123 && v12 == 120 && v14 == 122 && v13 == 121 )
31      {
32        strcpy(Text, "flag");                   // Text保存的是flag
33        memset(&v7, 0, 252u);
34        v8 = 0;
35        v9 = 0;
36        _itoa(v10, &v5, 10);                    // 把一个整数转换为字符串:
37                                                // v10是需要转换成字符串的数字
38                                                // v5是转换后保存字符串的变量
39                                                // 10是进制
40                                                //
41                                                // 因此需要把v10解出来
42        strcat(Text, "{");
43        strcat(Text, &v5);                      // 需要把v5解出来
44        strcat(Text, "_");
45        strcat(Text, "Buff3r_0v3rf|0w");
46        strcat(Text, "}");
47        MessageBoxA(0, Text, "well done", 0);
48      }
49      SetTimer(hWnd, 1u, 1000u, TimerFunc);
50    }
51    if ( a3 == 1001 )
52      KillTimer(hWnd, 1u);
53  }
54  return 0;
55}
0000113E sub_401090:14 (40113E)
```


3. 溢出

上面的方法感觉是这道题比较特殊，刚好可以直接逆出来v10的值。让我们来看看运行后的程序里，用户的输入应该是啥吧。应该考察的是溢出。

第25行中，读取用户输入保存到String，读取的最大长度为260字节，并且没有对String做长度校验，可能存在溢出的问题。

```
GetDlgItemTextA(hWnd, 1002, &String, 260);
```

然后对String进行了长度校验，必须要大于6才行。

接下来 `v10 = atoi(&String) + 1;` ，也就是用户输入是个数字的字符串，通过atoi转成数字以后再加一，将结果赋值给v10。因此输入的前三个字符应该为：`"123"`。

重头戏来了，还是这个if条件，对v10，v12，v13和v14进行了判断。

```
if ( v10 == 123 && v12 == 120 && v14 == 122 && v13 == 121 )
{
    ...
}
```

让我们看看String~v14在栈中的情况。

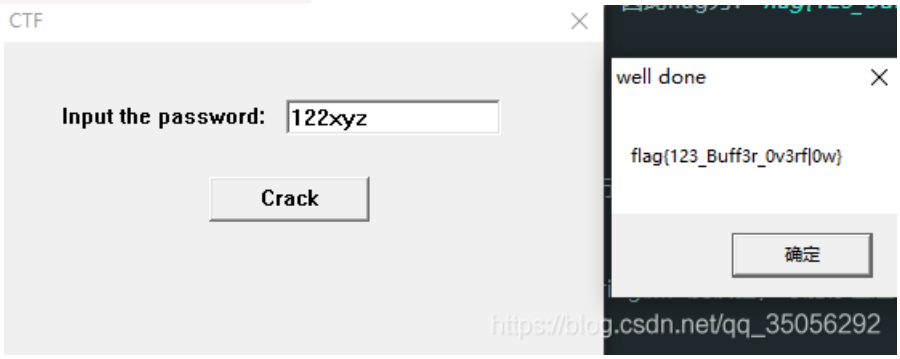
```
int __stdcall sub_401090(HWND hWnd, int a2, int a3, int a4)
{
  char v5; // [esp+50h] [ebp-310h]
  CHAR Text[4]; // [esp+154h] [ebp-20Ch]
  char v7; // [esp+159h] [ebp-207h]
  __int16 v8; // [esp+255h] [ebp-10Bh]
  char v9; // [esp+257h] [ebp-109h]
  int v10; // [esp+258h] [ebp-108h]
  CHAR String; // [esp+25Ch] [ebp-104h]
  char v12; // [esp+25Fh] [ebp-101h]
  char v13; // [esp+260h] [ebp-100h]
  char v14; // [esp+261h] [ebp-FFh]
```

String这三个字节填充的是122无疑，v12、v13、v14的ASCII值分别为120、121、122，即对应的字符为x、y、z

内存地址	数据	长度（字节）	具体内容
ebp-255	v14	1	122=>z
ebp-256	v13	1	121=> y
ebp-257	v12	1	120=>x
ebp-260	String	3	122

因此用户的输入应该为：`122xyz`

输入以后，得到flag：`flag{123_Buff3r_0v3rf|0w}`



4. 总结

这道题和攻防世界进阶区-srm-50都考察了溢出。

可以把用户输入保存的变量以及一些重要条件判断中的变量用一个栈（高地址在上方，低地址在下方）画出来，看一下他们的长度以及分布情况，从而计算出应该输入什么才能从用户输入覆盖到那些条件判断的变量。

> 画栈的时候，可以参考一下IDA提示的变量在栈中的位置，比如本题中String为 ebp-104h，v12为ebp-101h，那么String在栈中占的位置为 `ebp-104h，ebp-103h，ebp-102h` 一共三个字节。

比如本题的String是保存用户输入的变量，`v12~v14`是输出flag的基本块的条件判断变量。并且在读取用户输入的时候未做长度校验，因此可以通过String直接覆盖上去，把`v12~v14`修改成我们想要的值。