看雪ctf 流浪者 WP

原创

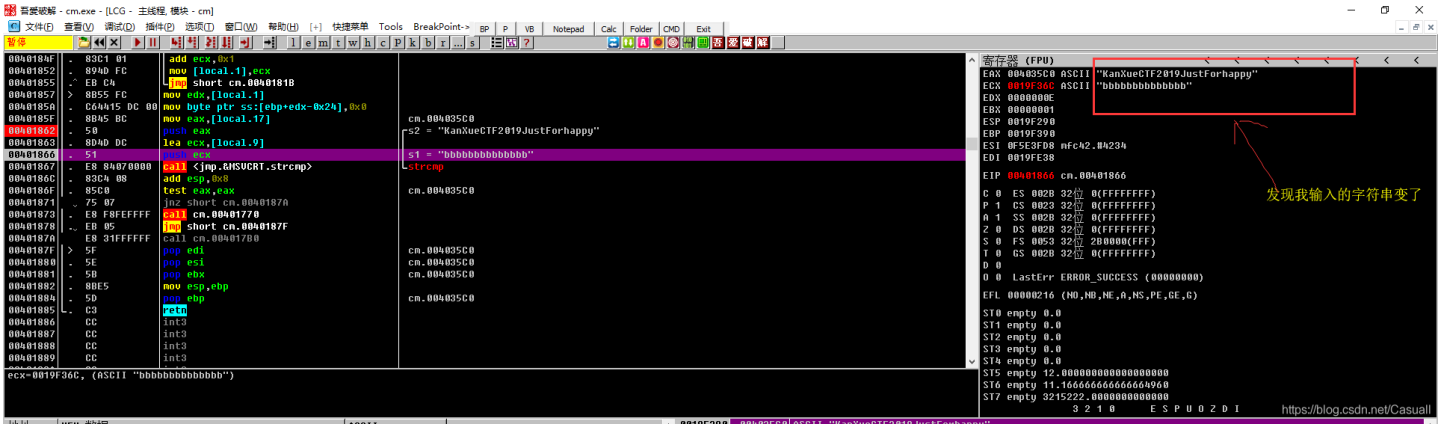
Casuall 于 2019-03-19 21:42:08 发布 659 收藏

分类专栏： Reverse 文章标签： 看雪ctf 流浪者 reverse

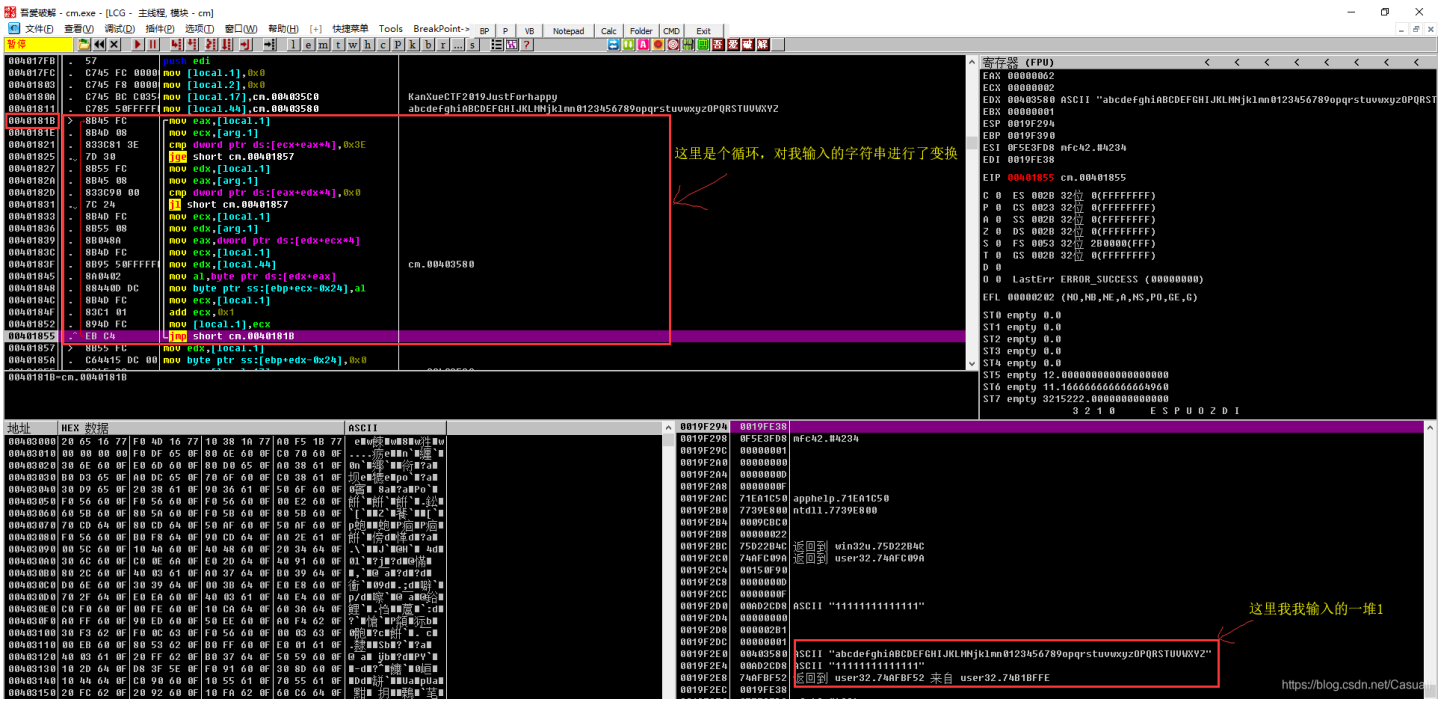
本文链接：https://blog.csdn.net/Casuall/article/details/88674344

看雪ctf 第一题 流浪者 wp

打开程序，提示要验证password，载入OD，按照签到题的思路，改了一个跳转，出现了提示框pass，但是没有flag，应该不是这种思路（我是一个刚入手的萌新，可能没改全，所以出不来flag，只能换一个思路了）。一步一步调试，发现程序比较的字符串和我输入的不一样，我输入了一堆1，但是发现最终比较的是一堆b，所以应该是中间做了一些处理，如图



，重新调试，发现了一个循环体，如图



，记录下循环体的地址`0x40181b`。

用IDA打开程序，跳转到`0x40181b`，查看伪代码，

```
int __cdecl ans(int a1)
{
  int result; // eax
  char Str1[28]; // [esp+D8h] [ebp-24h]
  int v3; // [esp+F4h] [ebp-8h]
  int v4; // [esp+F8h] [ebp-4h]

  v4 = 0;
  v3 = 0;
  while ( *(_DWORD *)(a1 + 4 * v4) < 62 && *(_DWORD *)(a1 + 4 * v4) >= 0 )
  {
    Str1[v4] = aAbcdefghiabcde[*(_DWORD *)(a1 + 4 * v4)];
    ++v4;
  }
  Str1[v4] = 0;
  if ( !strcmp(Str1, "KanXueCTF2019JustForhappy") )
    result = sub_401770();
  else
    result = sub_4017B0();
  return result;
}
```

这个名字是我自己改的

通过尝试，发现a1位一个数组，是偏移量的一个数组，括号里的意思应该就相当于往后移一位

,发现了**字符串比较函数**，如图中方框所示，如果`Str1`与`"KanXueCTF2019JustForhappy"`相等，则运行函数`sub_401770()`，双击进去查看，发现是通过的函数，然后往上看，`aAbcdefhiabcde`是一个字符串，完整的字符串为`abcdefghiABCDEFGHIJKLMNjklmn0123456789opqrstuvwxyzOPQRSTUVWXYZ`,然后查看传进来的`参数a1`，

IDA View-A | Pseudocode-A | Stack of ans | Strings window | Hex View-1 | Structures | Enums

```
int __thiscall ans2(CWnd *this)
{
  struct CString *v1; // ST08_4
  CWnd *v2; // eax
  int v3; // eax
  int v5[26]; // [esp+4Ch] [ebp-74h]
  int i; // [esp+B4h] [ebp-Ch]
  char *Str; // [esp+B8h] [ebp-8h]
  CWnd *v8; // [esp+BCh] [ebp-4h]

  v8 = this;
  v1 = (CWnd *)((char *)this + 100);
  v2 = CWnd::GetDlgItem(this, 1002);
  CWnd::GetWindowTextA(v2, v1);
  v3 = sub_401A30((char *)v8 + 100);
  Str = CString::GetBuffer((CWnd *)((char *)v8 + 100), v3);
  if ( !strlen(Str) )
    return CWnd::MessageBoxA(v8, "请输入pass!", 0, 0);
  for ( i = 0; Str[i]; ++i )
  {
    if ( Str[i] > 57 || Str[i] < 48 )
    {
      if ( Str[i] > 122 || Str[i] < 97 )
      {
        if ( Str[i] > 90 || Str[i] < 65 )
          sub_4017B0();
        else
          v5[i] = Str[i] - 29;
      }
      else
      {
        v5[i] = Str[i] - 87;
      }
    }
    else
    {
      v5[i] = Str[i] - 48;
    }
  }
  return ans((int)v5);
}
```

核心算法

,在**ans函数名**处按`x`查看调用ans函数的地方，跟过去看看，发现了另一个函数。

通过分析，整理一下思路，输入一个字符串，然后用图中核心算法生成一个`数组v5`，v5是一个偏移量的列表，然后将这个列表传入`ans（a1）`函数，得到Str1,相当于下面代码的赋值。

```
for(int i=0;i<25;i++){
    Str[i] = aAbcdefhiabcde[a[i]]
}
```

写一个python脚本是复原这个算法，就可以得到应该输入的字符串了

```
[19, 0, 27, 59, 44, 4, 11, 55, 14, 30, 28, 29, 37, 18, 44, 42, 43, 14, 38, 41, 7, 0, 39, 39, 48]
j0rXI4bTeustBiIGHeCF70DDM
[Finished in 0.2s]
```

```python
# coding=UTF-8

def getf(v5, i):
    global p1,flag
    if p < 25 and v5 == position[p1] :
     # print(content[i],'p1:',p1)
     flag += content[i]
     p1 += 1

def getPosition(KK,content,position):
    for i in KK:
     for j in range(0,len(content)):
      if i == content[j]:
       position.append(j)

def getflag(content,v5):          # 还原算法
    for m in range(0,25):         # 因为KK是长度为25的字符串，所以这里循环25次
     for i in range(0,len(content)):      # 遍历content,相当于爆破，找到v5(也就是a1)
      num = ord(content[i])
      if num > 57 or num < 48:
       if num > 122 or num < 97:
        if num > 90 or num < 65:
         pass
        else:
         v5 = num - 29
       else:
        v5 = num - 87
      else:
       v5 = num - 48
      getf(v5, i)


if __name__ == '__main__':
    content = 'abcdefghiABCDEFGHIJKLMNjklmn0123456789opqrstuvwxyzOPQRSTUVWXYZ'
    KK = 'KanXueCTF2019JustForhappy'
    p1 = 0
    position = []
    v5 = 0
    flag = ''
    getPosition(KK,content,position)      # 获取最终要比较的字符串KK的每个字符在content中的位置变换(也就是a1)
    print(position)
    getflag(content,v5)
    print(flag)
```

参考链接：

https://www.bilibili.com/video/av46108107

http://geekfz.cn/index.php/2019/03/19/reverse_liulangzhe/