新手ctf逆向-----BUU [ACTF新生赛2020]rome WP

原创

N1N3:-1 于 2020-09-13 11:55:34 发布 258 收藏

丢进ida

丢进去，找到主要函数

```
int func()
{
  int result; // eax@1
  int v1; // [sp+14h] [bp-44h]@8
  int v2; // [sp+18h] [bp-40h]@8
  int v3; // [sp+1Ch] [bp-3Ch]@8
  int v4; // [sp+20h] [bp-38h]@8
  unsigned __int8 v5; // [sp+24h] [bp-34h]@1
  unsigned __int8 v6; // [sp+25h] [bp-33h]@2
  unsigned __int8 v7; // [sp+26h] [bp-32h]@3
  unsigned __int8 v8; // [sp+27h] [bp-31h]@4
  unsigned __int8 v9; // [sp+28h] [bp-30h]@5
  int v10; // [sp+29h] [bp-2Fh]@8
  int v11; // [sp+2Dh] [bp-2Bh]@8
  int v12; // [sp+31h] [bp-27h]@8
  int v13; // [sp+35h] [bp-23h]@8
  unsigned __int8 v14; // [sp+39h] [bp-1Fh]@7
  char v15; // [sp+3Bh] [bp-1Dh]@1
  char v16; // [sp+3Ch] [bp-1Ch]@1
  char v17; // [sp+3Dh] [bp-1Bh]@1
  char v18; // [sp+3Eh] [bp-1Ah]@1
  char v19; // [sp+3Fh] [bp-19h]@1
  char v20; // [sp+40h] [bp-18h]@1
  char v21; // [sp+41h] [bp-17h]@1
  char v22; // [sp+42h] [bp-16h]@1
  char v23; // [sp+43h] [bp-15h]@1
  char v24; // [sp+44h] [bp-14h]@1
  char v25; // [sp+45h] [bp-13h]@1
  char v26; // [sp+46h] [bp-12h]@1
  char v27; // [sp+47h] [bp-11h]@1
  char v28; // [sp+48h] [bp-10h]@1
  char v29; // [sp+49h] [bp-Fh]@1
  char v30; // [sp+4Ah] [bp-Eh]@1
  char v31; // [sp+4Bh] [bp-Dh]@1
  int i; // [sp+4Ch] [bp-Ch]@8

  v15 = 'Q';
  v16 = 's';
  v17 = 'w';
  v18 = '3';
  v19 = 's';
  v20 = 'j';
  v21 = '_';
  v22 = 'l';
  v23 = 'z';
  v24 = '4';
  v25 = '_';
  v26 = 'U';
  v27 = 'j';
  v28 = 'w';
  v29 = '@';
  v30 = 'l';
  v31 = '\0';
  printf("Please input:");
  scanf("%s", &v5);
  result = v5;
  if ( v5 == 'A' )
  {
    result = v6;
    if ( v6 == 'C' )
    {
      result = v7;
      if ( v7 == 'T' )
      {
        result = v8;
        if ( v8 == 'F' )
        {
          result = v9;
          if ( v9 == '{' )
          {
            result = v14;
            if ( v14 == '}' )
            {
              v1 = v10;
      if ( v7 == 'T' )
      {
        result = v8;
        if ( v8 == 'F' )
        {
          result = v9;
          if ( v9 == '{' )
          {
            result = v14;
            if ( v14 == '}' )
            {
              v1 = v10;
              v2 = v11;
              v3 = v12;
              v4 = v13;
              for ( i = 0; i <= 15; ++i )
              {
                if ( *(&v1 + i) > 64 && *(&v1 + i) <= 90 )
                  *(&v1 + i) = (*(&v1 + i) - 51) % 26 + 65;
                if ( *(&v1 + i) > 96 && *(&v1 + i) <= 122 )
                  *(&v1 + i) = (*(&v1 + i) - 79) % 26 + 97;
              }
              for ( i = 0; i <= 15; ++i )
              {
                result = *(&v15 + i);
                if ( *(&v1 + i) != result )
                  return result;
              }
              result = printf("You are correct!");
            }
          }
        }
      }
    }
  }
  return result;
}
00000882 func:82
```

主要运算和判断就在这一块了

分析分析

在分析这一块，我自己一直被绕了进去，主要问题出现在v10到v13，我一直认为是四个字符，但是下面运算的循环有16次，以下是我错误的分析：
78行的第一个循环，是从v10开始的，我在想有16次循环，但未知的v10,v11,v12,v13只有四个，那剩下的12次就要对已有的v14~v26进行运算了，但85行的第二个循环确从v10和v15开始比较，这就很奇怪了，v10之后的16个字符经过了运算要和v15之后16个字符进行比较，俩串16个字符互相重叠，v15之后有的字符还没运算，这脚本该咋写。。。。
错误的分析到此为止，我在这卡了很久，一直在想这脚本该咋写，心里却觉得不咋对劲，这题不应该这么难啊，于是我进数据段看了看，发现了自己的憨批之处

```
int v10; // [sp+29h] [bp-2Fh]@8
int v11; // [sp+2Dh] [bp-2Bh]@8
int v12; // [sp+31h] [bp-27h]@8
int v13; // [sp+35h] [bp-23h]@8
```

这四个变量是int型的。。。。。每个都占了四个字节，也就是说每个变量都能放四个char，而其他的变量都是char型只能放一个字符，4*4=16，好吧，做运算的只有这16个未知字符

写脚本

这样一来，这脚本就好写多了呀，直接逐个爆破，爽滴很~

```
a='ACTF{'
v15='Qsw3sj_lz4_Ujw@l'
s=''
for k in range(len(v15)):
    for s in range(128):
        i = s
        if i > 64 and i <= 90:
            i = (i-51) % 26 + 65
        if i > 96 and i <= 122:
            i = (i-79) % 26 + 97
        if i == ord(v15[k]):
            a=a+chr(s)
print(a+'}')
```

```
终端  调试控制台  问题 100  输出  2: Python Debug Consc
da3\python.exe' 'c:\Users\92024\.vscode\extensions\ms-python.python-2020.8.108011\pythonFiles\lib
\python\debugpy\launcher' '50432' '--' 'd:\myfolder\1.py'
ACTF{Cae3ar_th4_cre@t}
PS D:\VSCode\Microsoft VS Code\myfolder> cd 'd:\VSCode\Microsoft VS Code\myfolder'; & 'D:\Anacon
da3\python.exe' 'c:\Users\92024\.vscode\extensions\ms-python.python-2020.8.108011\pythonFiles\lib
\python\debugpy\launcher' '50438' '--' 'd:\myfolder\1.py'
ACTF{Cae3ar_th4_cre@t}
PS D:\VSCode\Microsoft VS Code\myfolder> cd 'd:\VSCode\Microsoft VS Code\myfolder'; & 'D:\Anacon
da3\python.exe' 'c:\Users\92024\.vscode\extensions\ms-python.python-2020.8.108011\pythonFiles\lib
\python\debugpy\launcher' '50454' '--' 'd:\myfolder\1.py'
ACTF{Cae3ar_th4_Gre@t}
PS D:\VSCode\Microsoft VS Code\myfolder>
```



```
a='ACTF{' v15='Qsw3sj_lz4_Ujw@l' s='' for k in range(len(v15)): for s in range(128): i = s if i > 64 and i <= 90: i = (i-51) % 26 + 65 if i > 96 and i <= 122: i = (i-79) % 26 + 97 if i == ord(v15[k]): a=a+chr(s) print(a+'}')
```

虽说算法很简单，但我在python恶心的缩进上还是浪费了时间，真心觉得还没大括号好用。。。。。（新人吐槽，巨巨勿喷）

> flag{Cae3ar_th4_Gre@t}

总结

因为我的愚蠢，就因为个数据类型不清楚，太想当然，导致在这个并不难的题目上浪费了大量的时间，不应该不应该。。。。写个博客以示警告！