xctf reverse simple-check-100

原创

doudoudedi 于 2019-09-30 14:25:48 发布 129 收藏 1

分类专栏： 题目 文章标签： xctf reverse

本文链接：https://blog.csdn.net/qq_37433000/article/details/101770623

题目 专栏收录该内容

83 篇文章 2 订阅

订阅专栏

emem今天也是水题的一天记一道逆向这题其实是一道破解的题目但是用ollydbg调试会出现乱码估计是程序的问题

```
70  v3 = alloca(32);
71  v4 = alloca(32);
72  v9 = &v7;
73  printf("Key: ");
74  v6 = v9;
75  scanf("%s", v9);
76  if ( check_key(v9) )
77    interesting_function(&v8);                // 这里是关键
78  else
79    puts("Wrong");
80  return 0;
81}
```

逻辑是这个函数然后我们看他的汇编代码

```
text:0040159C                 mov     [esp+48h+Format], eax
text:0040159F                 call    _check_key
text:004015A4                 test    eax, eax
text:004015A6                 jz      short loc_4015B5
text:004015A8 ; 76:     interesting_function(&v8);                //
text:004015A8                 lea     eax, [ebp+var_2D]
text:004015AB                 mov     [esp+48h+Format], eax
text:004015AE                 call    _interesting_function
```

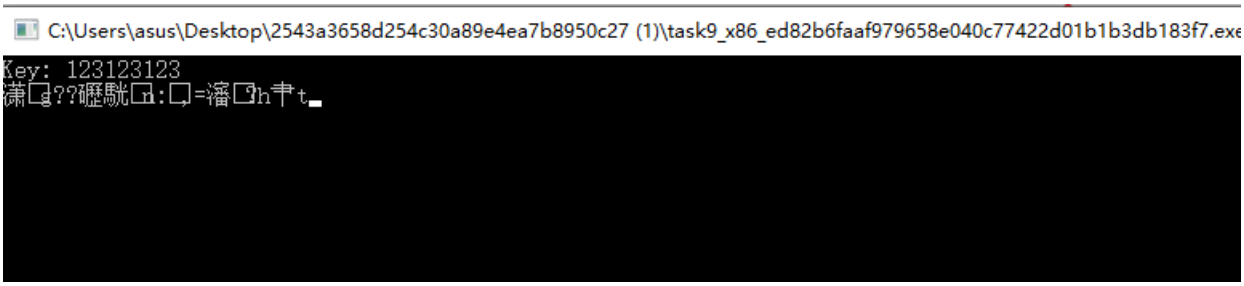
```
test eax,eax
jz address
```

这里就是汇编的知识了我们需要将eax置为1然后就会跳过jz执行之后的代码(或者直接NOP)

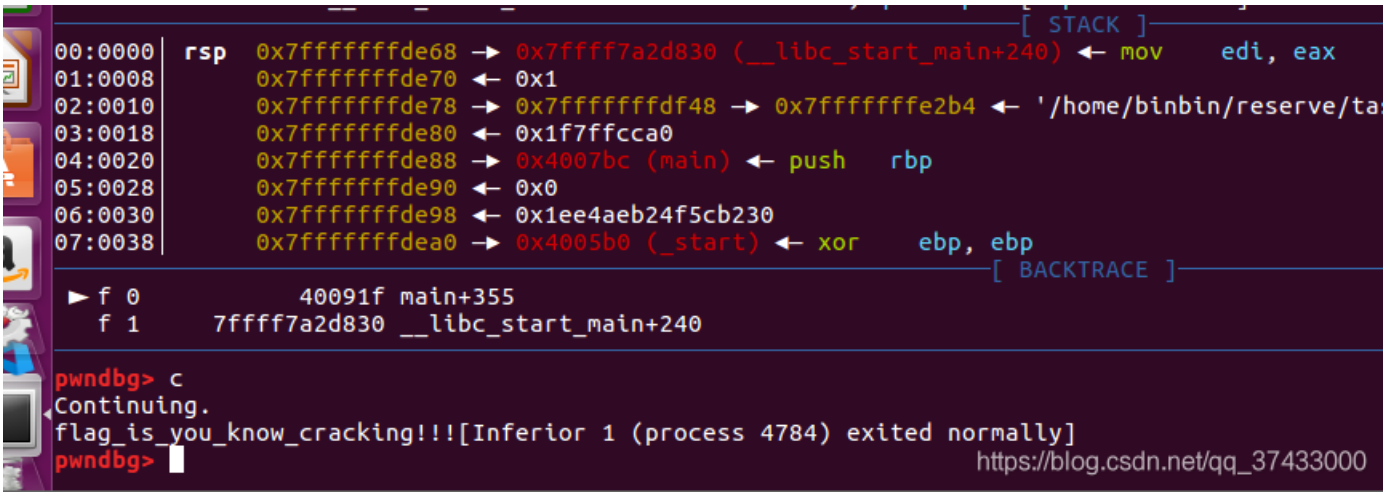
```
int *__cdecl interesting_function(int a1)
{
  int *result; // eax
  unsigned int v2; // [esp+1Ch] [ebp-1Ch]
  int *v3; // [esp+20h] [ebp-18h]
  int v4; // [esp+24h] [ebp-14h]
  int j; // [esp+28h] [ebp-10h]
  int i; // [esp+2Ch] [ebp-Ch]

  result = a1;
  v4 = a1;
  for ( i = 0; i <= 6; ++i )
  {
    v2 = *(4 * i + v4) ^ 0xDEADBEEF;
    result = &v2;
    v3 = &v2;
    for ( j = 3; j >= 0; --j )
      result = putchar((*(v3 + j) ^ flag_data[4 * i + j]));
  }
  return result;
}
```


这其实就是一个打印flag的函数我用od调出来是乱码



用gdb调出来flag



水题真开心