第五季极客挑战赛 逆向部分writeup

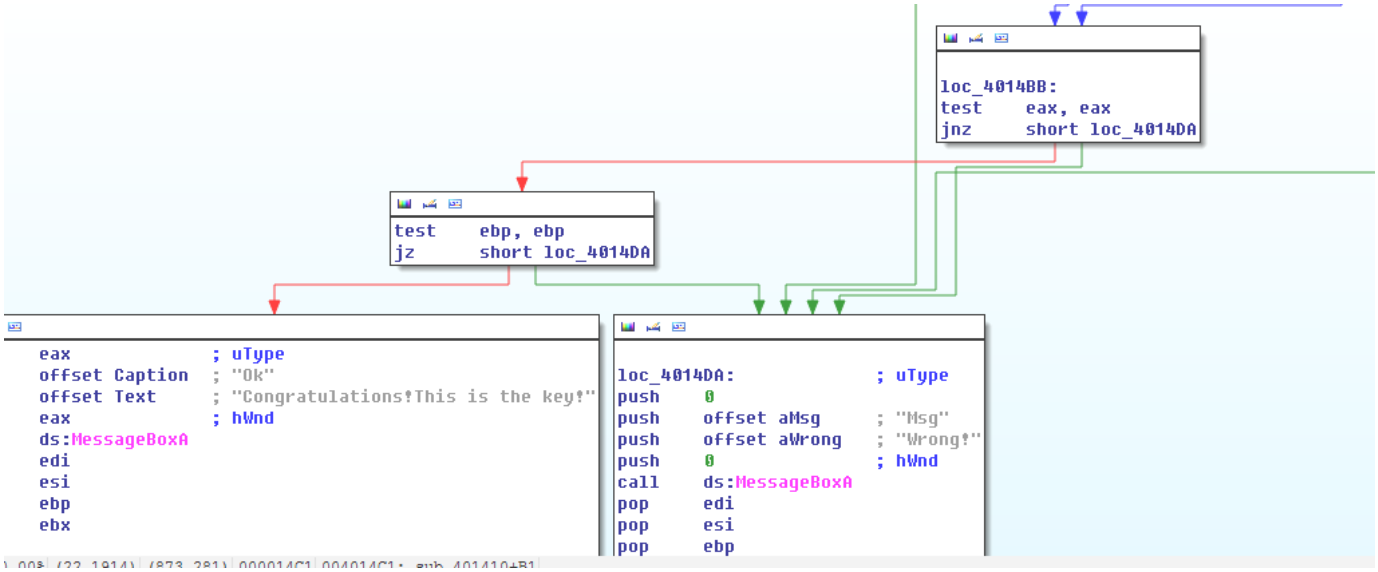
原创

dreaming_waiting 于 2015-03-19 17:08:24 发布 628 收藏

分类专栏： CTF--逆向 文章标签： python ctf reverse

本文链接：https://blog.csdn.net/lxx_nico/article/details/44460761

1. Reverse-ruhua




```
  {
    v5 = NULL;
  }
  CWnd::GetWindowTextA((char *)v1 + 160, v3, 20);
  CWnd::GetWindowTextA((char *)v1 + 96, v5, 20);
  v6 = strlen(v3);
  v7 = strlen(v5);
  if ( v6 > 0xA || v7 > 0xA || (sub_401500(v3), sub_401530(v5), strcmp(v3, v5)) || !v6 )
    result = MessageBoxA(NULL, "Wrong!", "Msg", 0);
  else
    result = MessageBoxA(NULL, "Congratulations!This is the key!", "Ok", 0);
  return result;
}
```

明显，我们要分析的是sub_401500和sub_401530

```
unsigned int __cdecl sub_401500(const char *a1)
{
  unsigned int result; // eax@1
  unsigned int v2; // kr04_4@1

  result = 0;
  v2 = strlen(a1) + 1;
  if ( v2 != 1 )
  {
```

```
  `
    do
    {
      a1[result] = (a1[result] ^ 3) - 20;
      ++result;
    }
    while ( result < v2 - 1 );
  }
  return result;
}
```

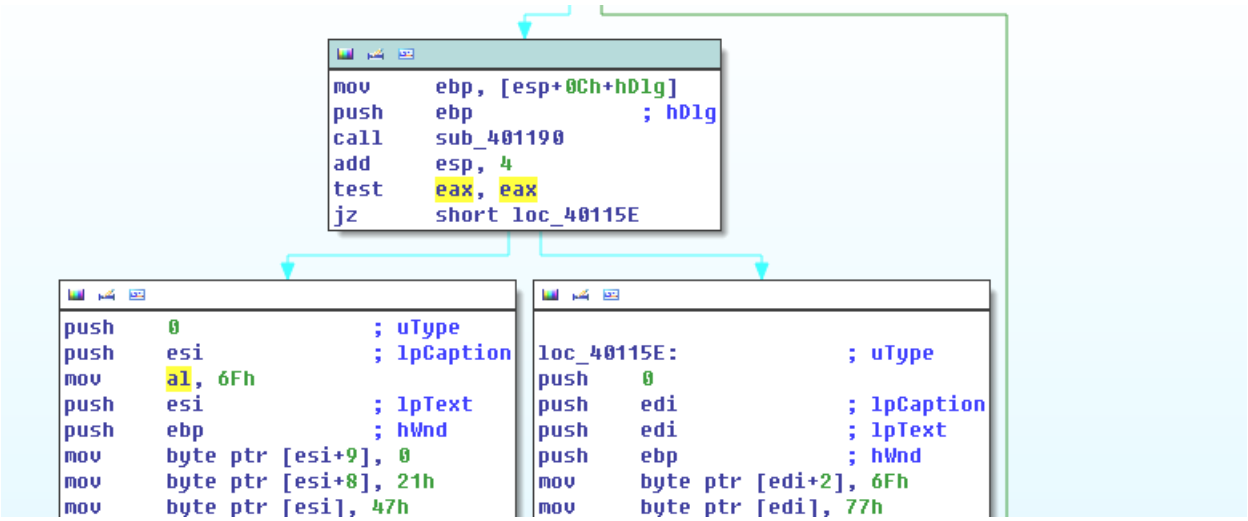
```
unsigned int __cdecl sub_401530(const char *a1)
{
  unsigned int result; // eax@1
  unsigned int v2; // kr04_4@1

  result = 0;
  v2 = strlen(a1) + 1;
  if ( v2 != 1 )
  {
    do
    {
      a1[result] = (a1[result] + 2) ^ 0x10;
      ++result;
    }
    while ( result < v2 - 1 );
  }
  return result;
}
```

看懂就用python算出来

```
Python:
username = raw_input('Username:')
password = ''
for i in range(len(username)):
    password += chr((((ord(username[i])^3)-20)^16)-2)
print "Password:" + password
```

1. reverse –cm2



Sub_401190是关键!

```
signed int v2; // ecx@2
char v3; // al@3
signed int v4; // eax@15
CHAR String; // [sp+10h] [bp-18h]@1
int flag[1]; // [sp+11h] [bp-17h]@1
int flag[5]; // [sp+15h] [bp-13h]@1
int flag[9]; // [sp+19h] [bp-Fh]@1
int flag[12]; // [sp+1Dh] [bp-Bh]@1
int v11; // [sp+21h] [bp-7h]@1
__int16 v12; // [sp+25h] [bp-3h]@1
char v13; // [sp+27h] [bp-1h]@1

flag[1] = 0;
flag[5] = 0;
flag[9] = 0;
```

针对这里的原来IDA自动定义的v5,v6,v7改过来，分别是flag[1],flag[2]。。等等的，然后我们就可以得到一系列的等式，最后直接用笔推算就没问题了。