xctf BABYRE

原创

菜逼的ctf之路 于 2020-10-10 19:05:09 发布 207 收藏 1

xctf BABYRE

第一次做smc类型题目，利用idapython把源代码补全
首先确保idapro里有python插件(一般都有的)，打开文件

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char s; // [rsp+0h] [rbp-20h]
  int v5; // [rsp+18h] [rbp-8h]
  int i; // [rsp+1Ch] [rbp-4h]

  for ( i = 0; i <= 181; ++i )
  {
    envp = (const char **)(*((unsigned __int8 *)judge + i) ^ 0xCu);
    *((_BYTE *)judge + i) ^= 0xCu;
  }
  printf("Please input flag:", argv, envp);
  __isoc99_scanf("%20s", &s);
  v5 = strlen(&s);
  if ( v5 == 14 && (unsigned int)judge((__int64)&s) )
    puts("Right!");
  else
    puts("Wrong!");
  return 0;
}
```

看代码逻辑，判断需要进入judge函数，然而发现这个函数有问题，看前面发现他把从judge函数开始后的181字节异或 0xc,其实也就是说从judge函数开始，异或0xc，然后形成的数据才能形成一个有效的代码段，这种方法可以给静态调试一定的困难(如果用动态调试。可以看到这里内存的变化，而且不用这么麻烦，但是我为了学习这种手法，所以选择静态调试)，所以这里需要一个idapython（不懂的百度）其实就是利用python脚本把judge处的代码补全，先贴上补丁脚本

```
import sys
from idautils import *
from idc import *
import idaapi
if __name__ =="__main__":
 start_addr=0x600B00
 for i in range(182):
  PatchByte(start_addr+i,Byte(start_addr+i)^0xC)
```

之后f5，重新反汇编，生成c语言，可以看到judge函数就变成了代码段

```
signed __int64 __fastcall judge(__int64 a1)
{
  char v2; // [rsp+8h] [rbp-20h]
  char v3; // [rsp+9h] [rbp-1Fh]
  char v4; // [rsp+Ah] [rbp-1Eh]
  char v5; // [rsp+Bh] [rbp-1Dh]
  char v6; // [rsp+Ch] [rbp-1Ch]
  char v7; // [rsp+Dh] [rbp-1Bh]
  char v8; // [rsp+Eh] [rbp-1Ah]
  char v9; // [rsp+Fh] [rbp-19h]
  char v10; // [rsp+10h] [rbp-18h]
  char v11; // [rsp+11h] [rbp-17h]
  char v12; // [rsp+12h] [rbp-16h]
  char v13; // [rsp+13h] [rbp-15h]
  char v14; // [rsp+14h] [rbp-14h]
  char v15; // [rsp+15h] [rbp-13h]
  int i; // [rsp+24h] [rbp-4h]

  v2 = 102;
  v3 = 109;
  v4 = 99;
  v5 = 100;
  v6 = 127;
  v7 = 107;
  v8 = 55;
  v9 = 100;
  v10 = 59;
  v11 = 86;
  v12 = 96;
  v13 = 59;
  v14 = 110;
  v15 = 112;
  for ( i = 0; i <= 13; ++i )
    *(_BYTE *)(i + a1) ^= i;
  for ( i = 0; i <= 13; ++i )
  {
    if ( *(_BYTE *)(i + a1) != *(&v2 + i) )
      return 0LL;
  }
  return 1LL;
}
```

如果和上图不一样，请自行百度ida中 p，c，d，这三个快捷键的用法，你会发现新天地，之后的逻辑就很简单了
直接贴脚本了

```
text=[102,109,99,100,127,107,55,100,59,86,96,59,110,112]
s=''
for i in range(len(text)):
    s+=chr(text[i]^i)
print(s)
```