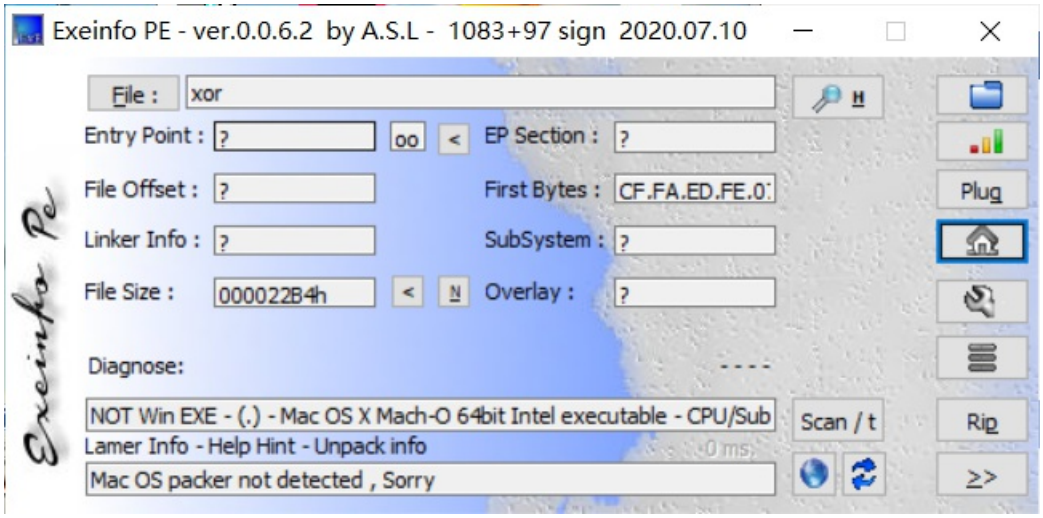
ctf xor题_BUUCTF-xor writeup

原创

weixin_39785524 于 2021-02-23 06:45:14 发布 134 收藏
文章标签： ctf xor题
版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/weixin_39785524/article/details/114492305
版权
将文件拖入ExEinfoPE



是64位的MACOSX 的可执行文件

拖进IDA中

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    char *v3; // rsi
4    int result; // eax
5    signed int i; // [rsp+2Ch] [rbp-124h]
6    char v6[264]; // [rsp+40h] [rbp-110h]
7    __int64 v7; // [rsp+148h] [rbp-8h]
8
9    memset(v6, 0, 0x100uLL);
10   v3 = (char *)256;
11   printf("Input your flag:\n", 0LL);
12   get_line(v6, 256LL);
13   if ( strlen(v6) != 33 )
14     goto LABEL_12;
15   for ( i = 1; i < 33; ++i )
16     v6[i] ^= v6[i - 1];
17   v3 = global;
18   if ( !strncmp(v6, global, 0x21uLL) )
19     printf("Success", v3);
20   else
21 LABEL_12:
22     printf("Failed", v3);
23   result = __stack_chk_guard;
24   if ( __stack_chk_guard == v7 )
25     result = 0;
26   return result;
27 }
```

程序比较简单，基本上所有步骤都在主函数内执行，v6是我们的输入内容

```
if ( strlen(v6) != 33 )
  goto LABEL_12;
```

可以观察知道，flag的长度为33

下面这部分对输入进行了一次处理，让输入的数组每一项与前一项异或，然后v6和v3进行比较

```
15  for ( i = 1; i < 33; ++i )
16    v6[i] ^= v6[i - 1];
17  v3 = global;
18  if ( !strncmp(v6, global, 0x21uLL) )
19    printf("Success", v3);
20  else
21LABEL_12:
22    printf("Failed", v3);
```

所以v3为处理后的内容

点global可以知道v3的内容，也可以在字符串表里看到

```
__data:0000000100001050 ; char *global
__data:0000000100001050 _global         dq offset aFKWOXZUPFVMDGH
__data:0000000100001050                                           ; DATA XREF: _main+10D↑r
__data:0000000100001050 __data          ends                      ; "f\nk\fw&O.@\x11x\rZ;U\x11p\x19F\x1Fv\"M"...
__data:0000000100001050
UNDE; ===========================================================================
UNDE
UNDE; Segment type: Pure data
UNDE__cstring       segment byte public 'DATA' use64                                  ========
UNDE                assume cs:__cstring
UNDE                ;org 100000F6Eh
UNDEaFKWOXZUPFVMDGH db 'f',0Ah              ; DATA XREF: __data:_global↓o
UNDE                db 'k',0Ch,'w&O.@',11h,'x',0Dh,'Z;U',11h,'p',19h,'F',1Fh,'v"M#D',0Eh,'g'
UNDE                db 6,'h',0Fh,'G2O',0                                              hk_guard_ptr↑o
```

```
's' __cstrin··· 00000022    C    f\nk\fw&O.@\x11x\rZ;U\x11p\x19F\x1Fv\"M#D\x0Eg\x06h\x0FG2O
```

所以只要对v3进行逆处理就可以得到flag

把v3转化为ASCII码为[102, 10, 107, 12, 119, 38, 79, 46, 64, 17, 120, 13, 90, 59, 85, 17, 112, 25, 70, 31, 118, 34, 77, 35, 68, 14, 103, 6, 104, 15, 71, 50, 79]

写脚本逆向处理(Tips: 一个数异或同一个数字两次可以得到它本身,比如 23 ^ 2 = 21, 21 ^ 2 = 23)

v6 = [102, 10, 107, 12, 119, 38, 79, 46, 64, 17, 120, 13, 90, 59, 85, 17, 112, 25, 70, 31, 118, 34, 77, 35, 68, 14, 103, 6, 104, 15, 71, 50, 79]

i= 32

while i>0:

v6[i]= v6[i] ^ v6[i-1]

i= i - 1i=0while i<33:print(chr(v6[i]), end="")

i= i + 1

运行得到flag

```
PS C:\Users\Kaguya\Desktop> python xor.py
flag{QianQiuWanDai_YiTongJiangHu}
```

flag{QianQiuWanDai_YiTongJiangHu}