XCTF easyGo

原创

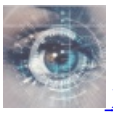
YenKoc 于 2020-04-16 11:36:00 发布 188 收藏

分类专栏： XCTF

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/YenKoc/article/details/105553121

版权

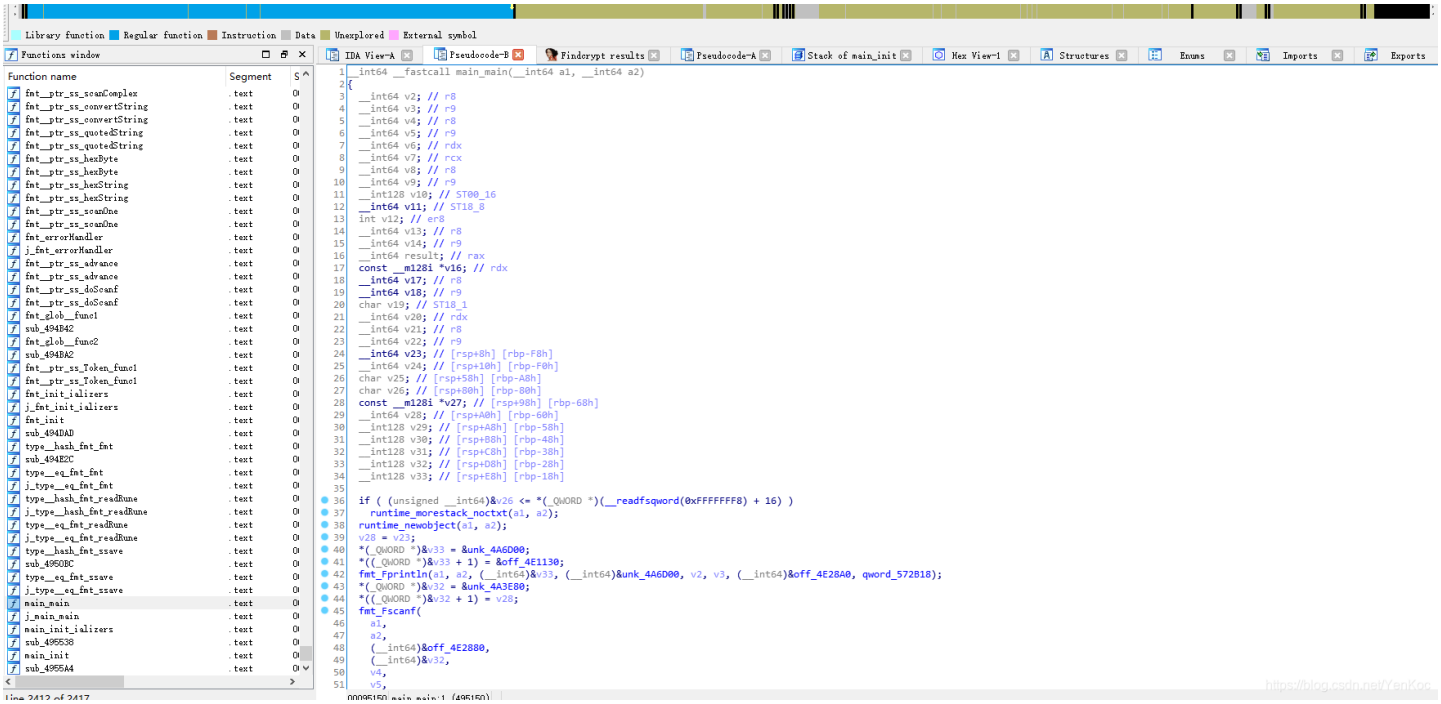


XCTF 专栏收录该内容

26 篇文章 2 订阅

订阅专栏

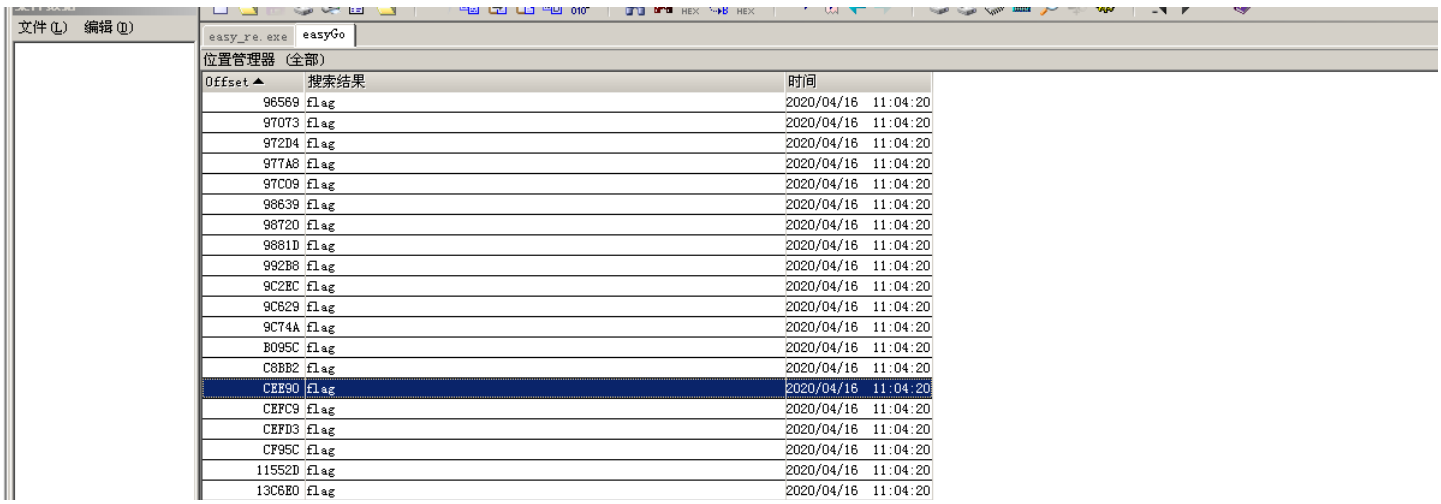
拖入ida，发现符号表需要还原一下，载入一个还原符号表的脚本。



```
__int64 __fastcall main_main(__int64 a1, __int64 a2)
{
  __int64 v2; // r8
  __int64 v3; // r9
  __int64 v4; // r8
  __int64 v5; // r9
  __int64 v6; // rdx
  __int64 v7; // rcx
  __int64 v8; // r8
  __int64 v9; // r9
  __int128 v10; // ST00_16
  __int64 v11; // ST18_8
  int v12; // er8
  __int64 v13; // r8
  __int64 v14; // r9
  __int64 result; // rax
  const __m128i *v16; // rdx
  __int64 v17; // r8
  __int64 v18; // r9
  char v19; // ST18_1
  __int64 v20; // rdx
  __int64 v21; // r8
  __int64 v22; // r9
  __int64 v23; // [rsp+8h] [rbp-F8h]
  __int64 v24; // [rsp+10h] [rbp-F0h]
  char v25; // [rsp+58h] [rbp-A8h]
  char v26; // [rsp+80h] [rbp-80h]
  const __m128i *v27; // [rsp+98h] [rbp-68h]
  __int64 v28; // [rsp+A0h] [rbp-60h]
  __int128 v29; // [rsp+A8h] [rbp-58h]
  __int128 v30; // [rsp+B8h] [rbp-48h]
  __int128 v31; // [rsp+C8h] [rbp-38h]
  __int128 v32; // [rsp+D8h] [rbp-28h]
  __int128 v33; // [rsp+E8h] [rbp-18h]

  if ( (unsigned __int64)&v26 <= *(_QWORD *)(__readfsqword(0xFFFFFFF8) + 16) )
    runtime_morestack_noctxt(a1, a2);
  runtime_newobject(a1, a2);
  v28 = v23;
  *(_QWORD *)&v33 = &unk_4A6D00;
  *((_QWORD *)&v33 + 1) = &off_4E1130;
  fmt_Fprintln(a1, a2, (__int64)&v33, (__int64)&unk_4A6D00, v2, v3, (__int64)&off_4E28A0, qword_572B18);
  *(_QWORD *)&v32 = &unk_4A3E80;
  *((_QWORD *)&v32 + 1) = v28;
  fmt_Fscanf(
    a1,
    a2,
    (__int64)&off_4E2880,
    (__int64)&v32,
    v4,
    v5,
```

go这个语言就有点恶心，字符串后面没有反斜杆零，ida识别出来，字符串就会挤在一堆，就很难看，看了某位师傅的wp，觉得这方法不错，就记录下，用winhex打开，根据文件的偏移转换成逻辑地址来看。

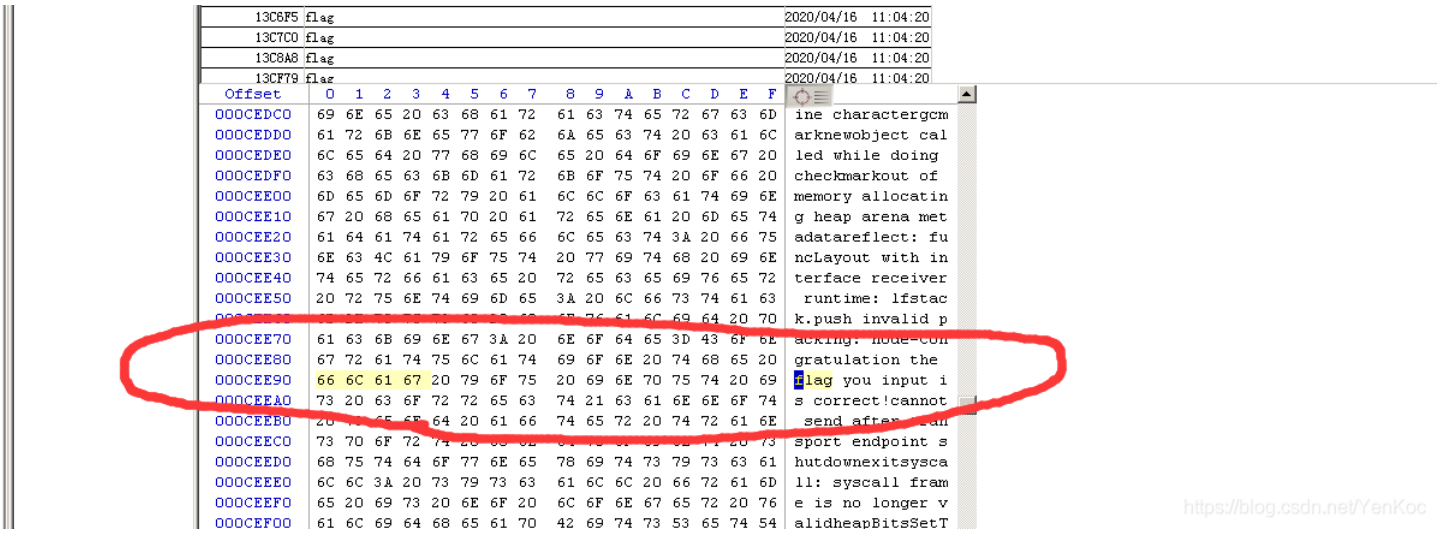


文件(L) 编辑(D)

easy_re.exe | easyGo

位置管理器 (全部)

Offset	搜索结果	时间
96569	flag	2020/04/16 11:04:20
97073	flag	2020/04/16 11:04:20
972D4	flag	2020/04/16 11:04:20
977A8	flag	2020/04/16 11:04:20
97C09	flag	2020/04/16 11:04:20
98639	flag	2020/04/16 11:04:20
98720	flag	2020/04/16 11:04:20
9881D	flag	2020/04/16 11:04:20
992B8	flag	2020/04/16 11:04:20
9C2EC	flag	2020/04/16 11:04:20
9C629	flag	2020/04/16 11:04:20
9C74A	flag	2020/04/16 11:04:20
B095C	flag	2020/04/16 11:04:20
C8BB2	flag	2020/04/16 11:04:20
CEE90	flag	2020/04/16 11:04:20
CEFC9	flag	2020/04/16 11:04:20
CEFD3	flag	2020/04/16 11:04:20
CF95C	flag	2020/04/16 11:04:20
11552D	flag	2020/04/16 11:04:20
13C6E0	flag	2020/04/16 11:04:20



找到了输入正确flag的提示信息。

```
.rodata:00000000004CEE7D unk_4CEE7D      db  43h ; C             ; DATA XREF: .rodata:off_4E1140↓o
.rodata:00000000004CEE7E                 db  6Fh ; o
.rodata:00000000004CEE7F                 db  6Eh ; n
.rodata:00000000004CEE80                 db  67h ; g
.rodata:00000000004CEE81                 db  72h ; r
.rodata:00000000004CEE82                 db  61h ; a
.rodata:00000000004CEE83                 db  74h ; t
.rodata:00000000004CEE84                 db  75h ; u
.rodata:00000000004CEE85                 db  6Ch ; l
.rodata:00000000004CEE86                 db  61h ; a
.rodata:00000000004CEE87                 db  74h ; t
.rodata:00000000004CEE88                 db  69h ; i
.rodata:00000000004CEE89                 db  6Fh ; o
.rodata:00000000004CEE8A                 db  6Eh ; n
.rodata:00000000004CEE8B                 db  20h
.rodata:00000000004CEE8C                 db  74h ; t
.rodata:00000000004CEE8D                 db  68h ; h
.rodata:00000000004CEE8E                 db  65h ; e
.rodata:00000000004CEE8F                 db  20h
.rodata:00000000004CEE90                 db  66h ; f
.rodata:00000000004CEE91                 db  6Ch ; l
.rodata:00000000004CEE92                 db  61h ; a
.rodata:00000000004CEE93                 db  67h ; g
.rodata:00000000004CEE94                 db  20h
.rodata:00000000004CEE95                 db  79h ; y
.rodata:00000000004CEE96                 db  6Fh ; o
.rodata:00000000004CEE97                 db  75h ; u
.rodata:00000000004CEE98                 db  20h
.rodata:00000000004CEE99                 db  69h ; i
.rodata:00000000004CEE9A                 db  6Eh ; n
.rodata:00000000004CEE9B                 db  70h ; p
.rodata:00000000004CEE9C                 db  75h ; u
.rodata:00000000004CEE9D                 db  74h ; t
.rodata:00000000004CEE9E                 db  20h
.rodata:00000000004CEE9F                 db  69h ; i
.rodata:00000000004CEEA0                 db  73h ; s
.rodata:00000000004CEEA1                 db  20h
.rodata:00000000004CEEA2                 db  63h ; c
.rodata:00000000004CEEA3                 db  6Fh ; o
.rodata:00000000004CEEA4                 db  72h ; r
.rodata:00000000004CEEA5                 db  72h ; r
.rodata:00000000004CEEA6                 db  65h ; e
.rodata:00000000004CEEA7                 db  63h ; c
.rodata:00000000004CEEA8                 db  74h ; t
.rodata:00000000004CEEA9                 db  21h ; !
.rodata:00000000004CEEAA                 db  63h ; c
```

https://blog.csdn.net/YenKoc

再查看它的引用，进一步找到关键函数

```
000495304                test    rcx, rcx
000495307                jnz     loc_49541B
00049530D
00049530D loc_49530D:                            ; CODE XREF: main_main+36A↓j
00049530D                mov     rax, [rsp+100h+var_60]
000495315                mov     rcx, [rax]
000495318                cmp     [rax+8], rbx
00049531C                jz      short loc_495393
00049531E
00049531E loc_49531E:                            ; CODE XREF: main_main+25B↓j
```

```
000049531E                xorps   xmm0, xmm0
0000495321                movups  [rsp+100h+var_58], xmm0
0000495329                lea     rax, unk_4A6D00
0000495330                mov     qword ptr [rsp+100h+var_58], rax
0000495338                lea     rax, off_4E1150
000049533F                mov     qword ptr [rsp+100h+var_58+8], rax
0000495347                nop
0000495348                mov     rax, cs:qword_572B18
000049534F                lea     rcx, off_4E28A0
0000495356                mov     [rsp+100h+var_100], rcx
000049535A                mov     [rsp+100h+var_F8], rax
000049535F                lea     rax, [rsp+100h+var_58]
0000495367                mov     [rsp+100h+var_F0], rax
000049536C                mov     [rsp+100h+var_E8], 1
0000495375                mov     [rsp+100h+var_E0], 1
000049537E                call    fmt_Fprintln
0000495383
0000495383 loc_495383:                            ; CODE XREF: main_main+2C6↓j
0000495383                mov     rbp, [rsp+100h+var_8]
000049538B                add     rsp, 100h
0000495392                retn
0000495393 ; ---------------------------------------------------------------------------
0000495393
0000495393 loc_495393:                            ; CODE XREF: main_main+1CC↑j
0000495393                mov     [rsp+100h+var_100], rdx
0000495397                mov     [rsp+100h+var_F8], rcx
000049539C                mov     [rsp+100h+var_F0], rbx
00004953A1                call    runtime_memequal
00004953A6                cmp     byte ptr [rsp+100h+var_E8], 0
```

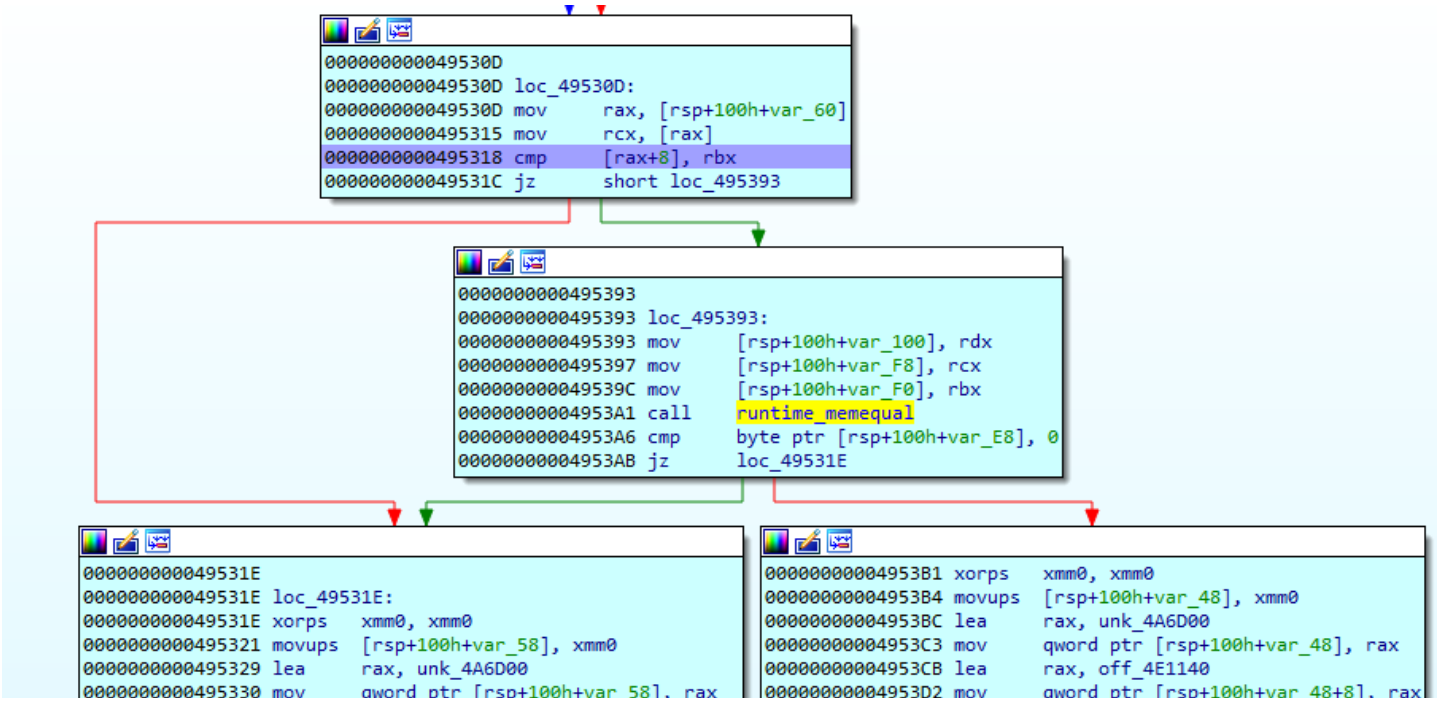
https://blog.csdn.net/YenKoc

那个是正确的跳转，下面这个肯定就是错误时的跳转了

```
                          jz      short loc_495393
000049531E
000049531E loc_49531E:                            ; CODE XREF: main_main+25B↓j
000049531E                xorps   xmm0, xmm0
0000495321                movups  [rsp+100h+var_58], xmm0
0000495329                lea     rax, unk_4A6D00
0000495330                mov     qword ptr [rsp+100h+var_58], rax
0000495338                lea     rax, off_4E1150
000049533F                mov     qword ptr [rsp+100h+var_58+8], rax
0000495347                nop
0000495348                mov     rax, cs:qword_572B18
000049534F                lea     rcx, off_4E28A0
0000495356                mov     [rsp+100h+var_100], rcx
000049535A                mov     [rsp+100h+var_F8], rax
000049535F                lea     rax, [rsp+100h+var_58]
0000495367                mov     [rsp+100h+var_F0], rax
000049536C                mov     [rsp+100h+var_E8], 1
0000495375                mov     [rsp+100h+var_E0], 1
000049537E                call    fmt_Fprintln
0000495383
```

https://blog.csdn.net/YenKoc

动调，到那个cmp那边，看看，估计是判断长度，再进一步比较



```
0000000000495338 lea     rax, off_4E1150
000000000049533F mov     qword ptr [rsp+100h+var_58+8], rax
0000000000495347 nop
0000000000495348 mov     rax, cs:qword_572B18
000000000049534F lea     rcx, off_4E28A0
```

```
00000000004953DA nop
00000000004953DB mov     rax, cs:qword_572B18
00000000004953E2 lea     rcx, off_4E28A0
00000000004953E9 mov     [rsp+100h+var_100], rcx
00000000004953ED mov     [rsp+100h+var_F8], rax
```

https://blog.csdn.net/YenKoc

在寄存器那里看到了flag，讲道理ida和gdb，各有千秋，gdb，很直观看到flag的。

```
RDX
RSI
R12
debug002:000000C0000140BE db  0Ah
debug002:000000C0000140BF db    0
debug002:000000C0000140C0 db  66h ; f
debug002:000000C0000140C1 db  6Ch ; l
debug002:000000C0000140C2 db  61h ; a
debug002:000000C0000140C3 db  67h ; g
debug002:000000C0000140C4 db  7Bh ; {
debug002:000000C0000140C5 db  39h ; 9
debug002:000000C0000140C6 db  32h ; 2
debug002:000000C0000140C7 db  30h ; 0
debug002:000000C0000140C8 db  39h ; 9
debug002:000000C0000140C9 db  34h ; 4
debug002:000000C0000140CA db  64h ; d
debug002:000000C0000140CB db  61h ; a
debug002:000000C0000140CC db  66h ; f
debug002:000000C0000140CD db  2Dh ; -
debug002:000000C0000140CE db  33h ; 3
debug002:000000C0000140CF db  33h ; 3
debug002:000000C0000140D0 db  63h ; c
debug002:000000C0000140D1 db  39h ; 9
debug002:000000C0000140D2 db  2Dh ; -
debug002:000000C0000140D3 db  34h ; 4
debug002:000000C0000140D4 db  33h ; 3
debug002:000000C0000140D5 db  31h ; 1
debug002:000000C0000140D6 db  65h ; e
debug002:000000C0000140D7 db  2Dh ; -
debug002:000000C0000140D8 db  61h ; a
debug002:000000C0000140D9 db  38h ; 8
debug002:000000C0000140DA db  35h ; 5
debug002:000000C0000140DB db  61h ; a
debug002:000000C0000140DC db  2Dh ; -
debug002:000000C0000140DD db  38h ; 8
debug002:000000C0000140DE db  62h ; b
debug002:000000C0000140DF db  66h ; f
debug002:000000C0000140E0 db  62h ; b
```

https://blog.csdn.net/YenKoc