New issue

Jump to bottom

stack-buffer-overflow at IEC10x/Iec104.c #14

Open **umxyz** opened this issue on Sep 23, 2019 · 0 comments

umxyz commented on Sep 23, 2019 • edited

I used gcc 5.4.0 with *CFLAGS=-g -fsanitize=address CXXFLAGS=-g -fsanitize=address LDFLAGS=-fsanitize=address to compile the IEC104*, and use *LD_PRELOAD=/root/preeny/x86_64-linux-gnu/desock.so ./iec104_monitor -m server -n 1 < test_case* to run the program, whilel found a **stack-buffer-overflow** in IEC10x/Iec104.c, Iec104_Deal_I

Snip **Iec104.c:1175**

```
    /* check asdu addrest */
    if(Iec10x_Sta_Addr != asdu->_addr){
        LOG("-%s-, error asdu addr(%x)(%x) \n" ,__FUNCTION__ ,Iec10x_Sta_Addr,asdu->_addr);
        return RET_ERROR;
    }
```

It looks like you do not check the value of Iec10x_Sta_Addr, when it's value become unexpect, there will be a stack-buffer-overflow, which cause the program exit, it is advisble to ensure the value of Iec10x_Sta_Addr limited in a safe range.

ASAN OUTPUT

```
==10033==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7fff99ddd2dd at pc 0x0000004b95c1 bp 0x7fff99ddca80 sp 0x7fff99ddca70  READ of size 2 at 0x7fff99ddd2dd thread
T0                                                                                                                                  #0 0x4b95c0 in Iec104_Deal_I ../IEC10X/Iec104.c:1175
    #1 0x4b9c11 in Iex104_Receive ../IEC10X/Iec104.c:1307
    #2 0x4be985 in Iec104_main /root/iec/Polar_104/test/main.c:423
    #3 0x405e53 in main /root/iec/Polar_104/test/main.c:629
    #4 0x7fe09fcb782f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #5 0x406238 in _start (/root/temp/iec/have_tested/Polar_104/test/iec104_monitor+0x406238)

Address 0x7fff99ddd2dd is located in stack of thread T0 at offset 1789 in frame                                                                  #0 0x4be39f in Iec104_main
/root/iec/Polar_104/test/main.c:255

  This frame has 5 object(s):
    [32, 36) 'sin_size'
    [96, 100) 'on'
    [160, 176) 's_add'
    [224, 240) 'c_add'
    [288, 1788) 'Iec104_RecvBuf' <== Memory access at offset 1789 overflows this variable                                                  HINT: this may be a false positive if
your program uses some custom stack unwind mechanism or swapcontext
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow ../IEC10X/Iec104.c:1175 Iec104_Deal_I
Shadow bytes around the buggy address:
  0x1000733b3a00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000733b3a10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000733b3a20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000733b3a30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000733b3a40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x1000733b3a50: 00 00 00 00 00 00 00 00 00 00 00[04]f3 f3 f3 f3                                                                  0x1000733b3a60: f3 f3 f3 f3 00 00 00
00 00 00 00 00 00 00 00 00                                                                                  0x1000733b3a70: f1 f1 f1 f1 04 f4 f4 f4 f2 f2 f2 f2 04 f4 f4 f4
0x1000733b3a80: f2 f2 f2 f2 00 f4 f4 f4 f2 f2 f2 f2 00 f4 f4 f4                                                                    0x1000733b3a90: f2 f2 f2 f2 00 04 f4 f4
f2 f2 f2 f2 00 00 00 00                                                                                 0x1000733b3aa0: f3 f3 f3 f3 f3 f3 f3 f3 00 00 00 00 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa                                                                                                          Heap right redzone:      fb
Freed heap region:       fd                                                                                                          Stack left redzone:      f1
Stack mid redzone:       f2
  Stack right redzone:     f3                                                                                                          Stack partial redzone:   f4
Stack after return:      f5                                                                                                          Stack use after scope:   f8
Global redzone:          f9                                                                                                          Global init order:       f6
Poisoned by user:        f7                                                                                                          Container overflow:      fc
Array cookie:            ac                                                                                                          Intra object redzone:    bb
ASan internal:           fe                                                                                                        ==10033==ABORTING
```

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant