mz-automation / libiec61850 Public

Code Issues 107 Pull requests 25 Actions Projects Wiki ···

New issue

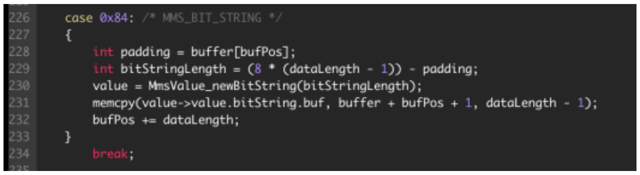
Jump to bottom

heap overflow when parsing `MMS_BIT_STRING` in MmsValue_decodeMmsData in mms/iso_mms/server/mms_access_result.c #200

Closed **sleicasper** opened this issue on Jan 12, 2020 · 3 comments

sleicasper commented on Jan 12, 2020

When libiec61850 parsing type `MMS_BIT_STRING` , it doesn't check variable `bufPos` . So we can provide a larger number for `bufPos` , then memory copy from `buffer + bufPos + 1` lead to heap overflow.



```
    case 0x84: /* MMS_BIT_STRING */
    {
        int padding = buffer[bufPos];
        int bitStringLength = (8 * (dataLength - 1)) - padding;
        value = MmsValue_newBitString(bitStringLength);
        memcpy(value->value.bitString.buf, buffer + bufPos + 1, dataLength - 1);
        bufPos += dataLength;
    }
        break;
```

poc:
poc.zip

result:

```
gdb-peda$ r  < ./poc
Starting program: /home/casper/targets/struct/libiec61850/afl/fuzzrun/fuzzmmsdata < poc
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
=================================================================
==24475==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000013 at pc 0x0000004a6864 bp 0x7fffffffe050 sp 0x7fffffffd800
WRITE of size 5 at 0x602000000013 thread T0
[New process 26870]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
process 26870 is executing new program: /home/casper/fuzz/fuzzdeps/llvm9/bin/llvm-symbolizer
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
    #0 0x4a6863 in __asan_memcpy /home/casper/fuzz/fuzzdeps/llvm-9.0.0.src/projects/compiler-rt/lib/asan/asan_interceptors_memintrinsics.cc:22
    #1 0x7ffff791103c in MmsValue_decodeMmsData /home/casper/targets/struct/libiec61850/afl/SRC/src/mms/iso_mms/server/mms_access_result.c:231:9
    #2 0x7ffff791174c in MmsValue_decodeMmsData /home/casper/targets/struct/libiec61850/afl/SRC/src/mms/iso_mms/server/mms_access_result.c:200:38
    #3 0x4ebb1e in main /home/casper/targets/struct/libiec61850/afl/../fuzzsrc/fuzzmmsdata.c:12:21
    #4 0x7ffff683db96 in __libc_start_main /build/glibc-OTsEL5/glibc-2.27/csu/../csu/libc-start.c:310
    #5 0x41ad59 in _start (/home/casper/targets/struct/libiec61850/afl/fuzzrun/fuzzmmsdata+0x41ad59)

0x602000000013 is located 0 bytes to the right of 3-byte region [0x602000000010,0x602000000013)
allocated by thread T0 here:
    #0 0x4a7a98 in calloc /home/casper/fuzz/fuzzdeps/llvm-9.0.0.src/projects/compiler-rt/lib/asan/asan_malloc_linux.cc:154
    #1 0x7ffff7b718ed in Memory_calloc /home/casper/targets/struct/libiec61850/afl/SRC/hal/memory/lib_memory.c:59:20

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/casper/fuzz/fuzzdeps/llvm-9.0.0.src/projects/compiler-rt/lib/asan/asan_interceptors_memintrinsics.cc:22 in __asan_memcpy
Shadow bytes around the buggy address:
  0x0c047fff7fb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c047fff8000: fa fa[03]fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8010: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
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
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  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
  Left alloca redzone:     ca
  Right alloca redzone:    cb
  Shadow gap:              cc
==24475==ABORTING
```

mzillgith added a commit that referenced this issue on Jan 13, 2020

`- MMS value parser: added plausibility check for bit-string padding v…` ··· `b4c7cef`

mzillgith commented on Jan 13, 2020 Contributor

Thanks for the hint. I added a plausibility check for the padding value.

mzillgith closed this as completed on Jan 13, 2020

sleicasper commented on Jan 13, 2020 Author

I think you should also check if `bufPos + buffer` reach end of `buffer`, because `bufPos` lead to heap overflow directly.

abergmann commented on Jan 15, 2020

CVE-2020-7054 was assigned to this issue.

DavidKorczynski mentioned this issue on Feb 22, 2021

libiec61850: initial integration google/oss-fuzz#5225

Merged

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants