New issue

Jump to bottom

AddressSanitizer: heap-buffer-overflow in GetByteStr() at ngiflib.c:108 in NGIFLIB_NO_FILE mode #19

Closed **Marsman1996** opened this issue on Jun 29, 2021 · 1 comment

Marsman1996 commented on Jun 29, 2021

Similar to #18, this Overflow problem is because in NGIFLIB_NO_FILE mode, `GetByteStr()` copy memory buffer without checking the boundary.

Test Environment

Ubuntu 16.04, 64bit
ngiflib(master `0245fd4`)

How to trigger

1. Compile the program with AddressSanitizer in NGIFLIB_NO_FILE mode `CC="clang -fsanitize=address -g" CFLAGS+=-DNGIFLIB_NO_FILE make`
2. run the compiled program `$ ./gif2tga --outbase /dev/null $POC`

POC file

https://github.com/Marsman1996/pocs/raw/master/ngiflib/poc-ngiflib-0245fd4-GetByteStr-overflow

Details

ASAN report

```
==19652==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x606000000118 at pc 0x0000004dd09d bp 0x7ffd61fa7590 sp 0x7ffd61fa6d40
READ of size 132 at 0x606000000118 thread T0
    #0 0x4dd09c in __asan_memcpy /home/mcgrady/wyh/llvm/llvm-6.0.0.src/projects/compiler-rt/lib/asan/asan_interceptors_memintrinsics.cc:23
    #1 0x51771a in GetByteStr /opt/disk/marsman/test/ngiflib/build_asan/ngiflib.c:108:3
    #2 0x51771a in LoadGif /opt/disk/marsman/test/ngiflib/build_asan/ngiflib.c:716
    #3 0x5161b5 in main /opt/disk/marsman/test/ngiflib/build_asan/gif2tga.c:95:10
    #4 0x7f3de58ea83f in __libc_start_main /build/glibc-S7Ft5T/glibc-2.23/csu/../csu/libc-start.c:291
    #5 0x419fe8 in _start (/opt/disk/marsman/test/ngiflib/bin_asan/bin/gif2tga+0x419fe8)

0x606000000118 is located 0 bytes to the right of 56-byte region [0x6060000000e0,0x606000000118)
allocated by thread T0 here:
    #0 0x4de218 in __interceptor_malloc /home/mcgrady/wyh/llvm/llvm-6.0.0.src/projects/compiler-rt/lib/asan/asan_malloc_linux.cc:88
    #1 0x516021 in main /opt/disk/marsman/test/ngiflib/build_asan/gif2tga.c:75:11
    #2 0x7f3de58ea83f in __libc_start_main /build/glibc-S7Ft5T/glibc-2.23/csu/../csu/libc-start.c:291

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/mcgrady/wyh/llvm/llvm-6.0.0.src/projects/compiler-rt/lib/asan/asan_interceptors_memintrinsics.cc:23 in __asan_memcpy
Shadow bytes around the buggy address:
  0x0c0c7fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c0c7fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c0c7fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c0c7fff8000: fa fa fa fa fd fd fd fd fd fd fd fa fa fa fa fa
  0x0c0c7fff8010: 00 00 00 00 00 00 00 01 fa fa fa fa 00 00 00 00
=>0x0c0c7fff8020: 00 00 00[fa]fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c7fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c7fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c7fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c7fff8060: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c7fff8070: 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
==19652==ABORTING
```

miniupnp commented on Aug 11, 2021 Owner

fixed with `19913ae`

miniupnp closed this as completed on Aug 11, 2021

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants