AddressSanitizer: stack-buffer-overflow at fromgif.c:310 #75

Closed **hongxuchen** opened this issue on Jul 28, 2018 · 2 comments

Assignees

Labels bug

hongxuchen commented on Jul 28, 2018

Our fuzzer detected several crashes when converting gif file against `2df6437` (compiled with Address Sanitizer). The command to trigger that is `img2sixel $POC -o /tmp/test.six` where $POC can be:
https://github.com/ntu-sec/pocs/blob/master/libsixel-2df6437/crashes/so_fromgif.c%3A310_1.gif
https://github.com/ntu-sec/pocs/blob/master/libsixel-2df6437/crashes/so_fromgif.c%3A310_2.gif

gdb output is like:

```
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from /home/hongxu/FOT/libsixel/install/bin/img2sixel...done.
Starting program: /home/hongxu/FOT/libsixel/install/bin/img2sixel hbo_fromgif.c:310_1.gif -o /tmp/test.six
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
=================================================================
==17533==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7fffffffa7d8 at pc 0x7ffff7b5f160 bp 0x7fffffff5950 sp 0x7fffffff5948
WRITE of size 2 at 0x7fffffffa7d8 thread T0
    #0 0x7ffff7b5f15f in gif_process_raster /home/hongxu/FOT/libsixel/src/fromgif.c:310:31
    #1 0x7ffff7b5c303 in gif_load_next /home/hongxu/FOT/libsixel/src/fromgif.c:462:22
    #2 0x7ffff7b5a25e in load_gif /home/hongxu/FOT/libsixel/src/fromgif.c:599:22
    #3 0x7ffff7b11198 in load_with_builtin /home/hongxu/FOT/libsixel/src/loader.c:858:18
    #4 0x7ffff7b10116 in sixel_helper_load_image_file /home/hongxu/FOT/libsixel/src/loader.c:1352:18
    #5 0x7ffff7b69d98 in sixel_encoder_encode /home/hongxu/FOT/libsixel/src/encoder.c:1737:14
    #6 0x515787 in main /home/hongxu/FOT/libsixel/converters/img2sixel.c:457:22
    #7 0x7ffff61a6b96 in __libc_start_main /build/glibc-OTsEL5/glibc-2.27/csu/../csu/libc-start.c:310
    #8 0x41a239 in _start (/home/hongxu/FOT/libsixel/install/bin/img2sixel+0x41a239)

Address 0x7fffffffa7d8 is located in stack of thread T0 at offset 18296 in frame
    #0 0x7ffff7b5999f in load_gif /home/hongxu/FOT/libsixel/src/fromgif.c:555

  This frame has 4 object(s):
    [32, 208) 's' (line 556)
    [272, 18296) 'g' (line 557) <== Memory access at offset 18296 overflows this variable
    [18560, 18568) 'frame' (line 559)
    [18592, 18600) 'fnp' (line 560)
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 /home/hongxu/FOT/libsixel/src/fromgif.c:310:31 in gif_process_raster
Shadow bytes around the buggy address:
  0x10007fff74a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff74b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff74c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff74d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff74e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x10007fff74f0: 00 00 00 00 00 00 00 00 00 00 00[f2]f2 f2 f2 f2
  0x10007fff7500: f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2
  0x10007fff7510: f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 00 f2 f2 f2
  0x10007fff7520: 00 f3 f3 f3 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff7530: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff7540: 00 00 00 00 00 00 00 00 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
  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
==17533==ABORTING

Program received signal SIGABRT, Aborted.
__GI_raise (sig=sig@entry=0x6) at ../sysdeps/unix/sysv/linux/raise.c:51
51      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
#0  __GI_raise (sig=sig@entry=0x6) at ../sysdeps/unix/sysv/linux/raise.c:51
#1  0x00007ffff61c5801 in __GI_abort () at abort.c:79
```

```
#2  0x000000000050376b in __sanitizer::Abort() ()
#3  0x0000000000500a98 in __sanitizer::Die() ()
#4  0x00000000004e2d1d in __asan::ReportGenericError(unsigned long, unsigned long, unsigned long, unsigned long, bool, unsigned long, unsigned int, bool) ()
#5  0x00000000004e374b in __asan_report_store2 ()
#6  0x00007ffff7b5f160 in gif_process_raster (s=0x7fffffff6080, g=0x7fffffff6170) at fromgif.c:310
#7  0x00007ffff7b5c304 in gif_load_next (s=0x7fffffff6080, g=0x7fffffff6170, bgcolor=0x0) at fromgif.c:462
#8  0x00007ffff7b5a25f in load_gif (buffer=0x62d000000400 "GIF89a\f", size=0xca, bgcolor=0x0, reqcolors=0x100, fuse_palette=0x1, fstatic=0x0, loop_control=0x0,
fn_load=0x7ffff7b6a090 <load_image_callback>, context=0x610000000040, allocator=0x604000000190) at fromgif.c:599
#9  0x00007ffff7b11199 in load_with_builtin (pchunk=0x603000000e20, fstatic=0x0, fuse_palette=0x1, reqcolors=0x100, bgcolor=0x0, loop_control=0x0, fn_load=0x7ffff7b6a090
<load_image_callback>, context=0x610000000040) at loader.c:858
#10 0x00007ffff7b10117 in sixel_helper_load_image_file (filename=0x7fffffffc9ed "hbo_fromgif.c:310_1.gif", fstatic=0x0, fuse_palette=0x1, reqcolors=0x100, bgcolor=0x0,
loop_control=0x0, fn_load=0x7ffff7b6a090 <load_image_callback>, finsecure=0x0, cancel_flag=0x13b61c0 <signaled>, context=0x610000000040, allocator=0x604000000190) at loader.c:1352
#11 0x00007ffff7b69d99 in sixel_encoder_encode (encoder=0x610000000040, filename=0x7fffffffc9ed "hbo_fromgif.c:310_1.gif") at encoder.c:1737
#12 0x0000000000515788 in main (argc=0x4, argv=0x7fffffffc488) at img2sixel.c:457
```

👍 1

saitoha self-assigned this on Jul 30, 2018

saitoha added a commit that referenced this issue on Aug 1, 2018

`gif loader: check LZW code size (Issue #75)` 7808a06

saitoha commented on Aug 1, 2018 Owner

Thank you. I decide to restrict LZW code size for now.

saitoha added the bug label on Aug 2, 2018

saitoha added a commit that referenced this issue on Dec 14, 2019

`Merge the fix for #75, reported by @hongxuchen` 0887007

saitoha commented on Dec 15, 2019 Owner

v1.8.3 includes the fix for this problem, thanks.

saitoha closed this as completed on Dec 15, 2019

saitoha mentioned this issue on Dec 17, 2019

buffer overflow issue - OOB write #88

Closed

Assignees

saitoha

Labels

bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants