saitoha / libsixel Public

<> Code ⊙ Issues 43 Pull requests 7 ▷ Actions Projects Wiki ···

New issue

Jump to bottom

# Address Sanitizer: invalid read at stb_image.h:5669 #74

Closed **hongxuchen** opened this issue on Jul 28, 2018 · 3 comments

Assignees

**hongxuchen** commented on Jul 28, 2018

Our fuzzer detected several crashes when converting PSD file against `2df6437` (compiled with Address Sanitizer). The command to trigger that is `img2sixel $POC -o /tmp/test.six` where $POC is:

https://github.com/ntu-sec/pocs/blob/master/libsixel-2df6437/crashes/read_stb_image.h%3A5669_1.psd

gdb output:

```
Reading symbols from /home/hongxu/FOT/libsixel-fuzz/install/bin/img2sixel...done.
Starting program: /home/hongxu/FOT/libsixel-fuzz/install/bin/img2sixel read_stb_image.h:5669_1.psd -o /dev/null
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff7a75bbc in stbi__psd_decode_rle (p=<optimized out>, pixelCount=0x1e000096, s=<optimized out>) at ./stb_image.h:5669
5669          len = stbi__get8(s);
#0  0x00007ffff7a75bbc in stbi__psd_decode_rle (p=<optimized out>, pixelCount=0x1e000096, s=<optimized out>) at ./stb_image.h:5669
#1  stbi__psd_load (s=0x7fffffffb990, x=<optimized out>, y=<optimized out>, comp=0x7fffffffbab0, req_comp=0x3, ri=<optimized out>, bpc=<optimized out>) at ./stb_image.h:5809
#2  stbi__load_main (s=<optimized out>, x=<optimized out>, y=<optimized out>, comp=0x7fffffffbab0, req_comp=0x3, ri=0x7fffffffb200, bpc=<optimized out>) at ./stb_image.h:992
#3  0x00007ffff7a29fa9 in stbi__load_and_postprocess_8bit (s=0x7fffffffb990, x=0x607000000038, y=0x60700000003c, comp=0x7fffffffbab0, req_comp=0x3) at ./stb_image.h:1090
#4  0x00007ffff7a4eb05 in load_with_builtin (pchunk=<optimized out>, fstatic=0x0, fuse_palette=0x1, loop_control=0x0, context=<optimized out>, reqcolors=<optimized out>, bgcolor=
<optimized out>, fn_load=<optimized out>) at loader.c:882
#5  sixel_helper_load_image_file (filename=0x7fffffffb830 " ", fstatic=<optimized out>, fuse_palette=<optimized out>, reqcolors=<optimized out>, bgcolor=<optimized out>,
loop_control=<optimized out>, fn_load=<optimized out>, finsecure=<optimized out>, cancel_flag=<optimized out>, context=<optimized out>, allocator=<optimized out>) at loader.c:1352
#6  0x00007ffff7b5be07 in sixel_encoder_encode (encoder=0x610000000040, filename=0x7fffffffc9e3 "read_stb_image.h:5669_1.psd") at encoder.c:1737
#7 0x0000000000515390 in main (argc=0x4, argv=0x7fffffffc478) at img2sixel.c:457
```

**saitoha** self-assigned this on Aug 4, 2018

**saitoha** added a commit that referenced this issue on Dec 23, 2019

`Introduce SIXEL_ALLOCATE_BYTES_MAX macro and limit allocation size to…` ··· `0b1e0b3`

**saitoha** commented on Dec 23, 2019 Owner

This problem seems to be caused when libsixel is compiled with `-fsanitize=address` flag.

with `-fsanitize=address` :

```
$ (CFLAGS="-O0 -g -fsanitize=address" ./configure && make) 2>&1 > /dev/null && converters/img2sixel https://github.com/ntu-sec/pocs/raw/master/libsixel-
2df6437/crashes/read_stb_image.h:5669_1.psd 2>&1 | head
ar: `u' modifier ignored since `D' is the default (see `U')
==6630==ERROR: AddressSanitizer failed to allocate 0x78003000 (2013278208) bytes of LargeMmapAllocator (error code: 12)
==6630==Process memory map follows:
        0x00007fff7000-0x00008fff7000
        0x00008fff7000-0x02008fff7000
        0x02008fff7000-0x10007fff8000
        0x55be9fd65000-0x55be9fd6e000   /home/vagrant/libsixel/converters/.libs/img2sixel
        0x55be9ff6d000-0x55be9ff6e000   /home/vagrant/libsixel/converters/.libs/img2sixel
        0x55be9ff6e000-0x55be9ff6f000   /home/vagrant/libsixel/converters/.libs/img2sixel
        0x600000000000-0x602000000000
        0x602000000000-0x602000050000
```

without `-fsanitize=address` :

```
$ (CFLAGS="-O0 -g" ./configure && make) 2>&1 > /dev/null && converters/img2sixel https://github.com/ntu-sec/pocs/raw/master/libsixel-2df6437/crashes/read_stb_image.h:5669_1.psd
2>&1 | head
ar: `u' modifier ignored since `D' is the default (see `U')
stb_image error
outofmem
```

**saitoha** commented on Dec 23, 2019 Owner

`0b1e0b3` avoids SEGV by limiting the allocation size to 128MB.

**saitoha** commented on Jan 2, 2020 Owner

Fixed on v1.8.5. Thanks!

**saitoha** closed this as completed on Jan 2, 2020

Assignees

saitoha

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants