New issue

Jump to bottom

Unverified indexs into the array lead to out of bound access in fromgif.c:283 #136

Open **peanuts62** opened this issue on Apr 15, 2020 · 4 comments · Fixed by libsixel/libsixel#8

peanuts62 commented on Apr 15, 2020

run_cmd

```
img2sixel -8 array_overflow
```

poc
the asan log

```
ASAN:DEADLYSIGNAL
=================================================================
==21151==ERROR: AddressSanitizer: SEGV on unknown address 0x10007fffb228 (pc 0x7ffff6bb5a03 bp 0x7fffffff86f0 sp 0x7fffffff86e0 T0)
==21151==The signal is caused by a READ memory access.
    #0 0x7ffff6bb5a02 in gif_out_code /home/parallels/Desktop/libsixel-master/src/fromgif.c:283
    #1 0x7ffff6bb5ab8 in gif_out_code /home/parallels/Desktop/libsixel-master/src/fromgif.c:284
    #2 0x7ffff6bb66ff in gif_process_raster /home/parallels/Desktop/libsixel-master/src/fromgif.c:393
    #3 0x7ffff6bb72d6 in gif_load_next /home/parallels/Desktop/libsixel-master/src/fromgif.c:502
    #4 0x7ffff6bb8130 in load_gif /home/parallels/Desktop/libsixel-master/src/fromgif.c:656
    #5 0x7ffff6bb1d26 in load_with_builtin /home/parallels/Desktop/libsixel-master/src/loader.c:908
    #6 0x7ffff6bb26d0 in sixel_helper_load_image_file /home/parallels/Desktop/libsixel-master/src/loader.c:1418
    #7 0x7ffff6bc22cb in sixel_encoder_encode /home/parallels/Desktop/libsixel-master/src/encoder.c:1743
    #8 0x55555555830e in main /home/parallels/Desktop/libsixel-master/converters/img2sixel.c:457
    #9 0x7ffff674bb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #10 0x555555555c59 in _start (/home/parallels/Desktop/libsixel-master/converters/.libs/img2sixel+0x1c59)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV /home/parallels/Desktop/libsixel-master/src/fromgif.c:283 in gif_out_code
==21151==ABORTING
```

analyse :

I use the gdb to debug the bug. I found in the fromgif.c:283 ,the `code = 0x7fff` is larger than the structure of g which define as 4096. so the crash occur !

source code is here:

```
enum {
   gif_lzw_max_code_size = 12
};

typedef struct
{
   int w, h;
   unsigned char *out;  /* output buffer (always 4 components) */
   int flags, bgindex, ratio, transparent, eflags;
   unsigned char pal[256][3];
   unsigned char lpal[256][3];
   gif_lzw codes[1 << gif_lzw_max_code_size];
   unsigned char *color_table;
   int parse, step;
   int lflags;
   int start_x, start_y;
   int max_x, max_y;
   int cur_x, cur_y;
   int actual_width, actual_height;
   int line_size;
   int loop_count;
   int delay;
   int is_multiframe;
   int is_terminated;
} gif_t;
```

bug position:

```
static void
gif_out_code(
    gif_t           /* in */ *g,
    unsigned short  /* in */ code
)
{
    /* recurse to decode the prefixes, since the linked-list is backwards,
       and working backwards through an interleaved image would be nasty */
    if (g->codes[code].prefix >= 0) {
        gif_out_code(g, (unsigned short)g->codes[code].prefix);
    }

    if (g->cur_y >= g->max_y) {
        return;
    }
```

gdb log :

```
In file: /home/parallels/Desktop/libsixel-master/src/fromgif.c
   278     unsigned short  /* in */ code
   279 )
   280 {
   281     /* recurse to decode the prefixes, since the linked-list is backwards,
   282        and working backwards through an interleaved image would be nasty */
 ► 283     if (g->codes[code].prefix >= 0) {
   284         gif_out_code(g, (unsigned short)g->codes[code].prefix);
```

```
   285     }
   286
   287     if (g->cur_y >= g->max_y) {
   288         return;
──────────────────────────────────────────[ STACK ]──────────────────────────────────────────
00:0000| rsp  0x7fffffff8670 ◂— 0x7fff00000000
01:0008|      0x7fffffff8678 —▸ 0x7fffffff8ab0 —▸ 0xfa0000007d0 ◂— 0x0
02:0010| rbp  0x7fffffff8680 —▸ 0x7fffffff86a0 —▸ 0x7fffffff8700 —▸ 0x7fffffff88d0 —▸ 0x7fffffffd2b0 ◂— ...
03:0018|      0x7fffffff8688 —▸ 0x7ffff6bb5ab9 (gif_out_code+246) ◂— mov    rax, qword ptr [rbp - 8]
04:0020|      0x7fffffff8690 ◂— 0x100200000000
05:0028|      0x7fffffff8698 —▸ 0x7fffffff8ab0 —▸ 0xfa0000007d0 ◂— 0x0
06:0030|      0x7fffffff86a0 —▸ 0x7fffffff8700 —▸ 0x7fffffff88d0 —▸ 0x7fffffffd2b0 —▸ 0x7fffffffd680 ◂— ...
07:0038|      0x7fffffff86a8 —▸ 0x7ffff6bb6700 (gif_process_raster+1391) ◂— mov    eax, dword ptr [rbp - 0x1c]
──────────────────────────────────────────[ BACKTRACE ]──────────────────────────────────────────
 ▸ f 0     7ffff6bb5a03 gif_out_code+64
   f 1     7ffff6bb5ab9 gif_out_code+246
   f 2     7ffff6bb6700 gif_process_raster+1391
   f 3     7ffff6bb72d7 gif_load_next+2953
   f 4     7ffff6bb8131 load_gif+1590
   f 5     7ffff6bb1d27 load_with_builtin+2481
   f 6     7ffff6bb26d1 sixel_helper_load_image_file+854
   f 7     7ffff6bc22cc sixel_encoder_encode+1121
   f 8     55555555830f main+9378
   f 9     7ffff674bb97 __libc_start_main+231
Program received signal SIGSEGV (fault address 0x10007fffb21a)
pwndbg> p code
$1 = 32767
```

version:

```
➜  libsixel-master ./converters/img2sixel --version
img2sixel 1.8.6

configured with:
  libcurl: no
  libpng: yes
  libjpeg: yes
  gdk-pixbuf2: no
  GD: no

Copyright (C) 2014-2018 Hayaki Saito <saitoha@me.com>.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

complies command

```
./configure CC="gcc" CXX="g++" CFLAGS="-g -O0 -fsanitize=address"
```

👍 1

peanuts62 mentioned this issue on Apr 16, 2020

Invalid free wild pointer lead to DOS in load_png in loader.c #134

Closed

carnil commented on Nov 22, 2020

CVE-2020-19668 was assigned for this issue.

NicoleG25 commented on Dec 1, 2020

Hi **@saitoha**
Do you happen to know if there is any plan to address this issue ?

Thanks in advance!

👍 2

dotlambda mentioned this issue on Feb 1, 2021

libsixel: mark as insecure NixOS/nixpkgs#111579

Merged

JISakuya commented on May 19, 2021 • edited

LZW Minimum Code Size determines the initial number of bits used for LZW codes in the image data, and 2**12+2 is more than 4096. So the 12-bits limitation is not for this, and the condition in the fromgif.c:328 is wrong.

This was referenced on Jun 9, 2021

Notification of fork at libsixel/libsixel (libsixel/libsixelのフォークのお知らせ). This project is unmaintained. (管理者 @saitoha が不在です。) #154

Open

CVE-2020-19668: Unverified indexes into array lead to out of bound access in the gif_out_code function in fromgif.c in libsixel 1.8.6. libsixel/libsixel#7

Closed

ctrlcctrlv added a commit to libsixel/libsixel that referenced this issue on Jun 9, 2021

`[SECURITY] Verify LZW code fits in 12 bits before we use it` ··· `07438fb`

ctrlcctrlv mentioned this issue on Jun 9, 2021

[SECURITY] Verify LZW code fits in 12 bits before we use it libsixel/libsixel#8

Merged

ctrlcctrlv added a commit to libsixel/libsixel that referenced this issue on Jun 9, 2021

`[SECURITY] Verify LZW code fits in 12 bits before we use it` ··· `05e5d21`

ctrlcctrlv commented on Jun 9, 2021

This issue has been patched in the fork. See libsixel#8, PR.

This repository has an absent maintainer. It's unlikely the maintainer will ever return, therefore the fork effort is described in #154. Distributions, users, and all other stakeholders are encouraged to switch to the fork.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

[SECURITY] Verify LZW code fits in 12 bits before we use it
libsixel/libsixel

5 participants