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

Closed sleicasper opened this issue on Apr 11, 2020 · 12 comments

sleicasper commented on Apr 11, 2020

`load_png` has a pointer `rows`, which should be set to NULL, otherwise `cleanup` code would use(calling `free`) it.

binary: img2sixel
file: loader.c
function: load_png

poc:
poc.zip

result:

```
#0  __GI_raise (sig=sig@entry=0x6) at ../sysdeps/unix/sysv/linux/raise.c:51
#1  0x00007ffff6fcf801 in __GI_abort () at abort.c:79
#2  0x00007ffff7018897 in __libc_message (action=action@entry=do_abort, fmt=fmt@entry=0x7ffff7145b9a "%s\n") at ../sysdeps/posix/libc_fatal.c:181
#3  0x00007ffff701f90a in malloc_printerr (str=str@entry=0x7ffff7143d88 "free(): invalid pointer") at malloc.c:5350
#4  0x00007ffff7026e1c in _int_free (have_lock=0x0, p=0x65a5b8, av=0x7ffff737ac40 <main_arena>) at malloc.c:4157
#5  __GI___libc_free (mem=0x65a5c8) at malloc.c:3124
#6  0x0000000000426701 in load_png (result=<optimized out>, buffer=<optimized out>, size=<optimized out>, psx=<optimized out>, psy=0x65a5dc, ppalette=<optimized out>,
pncolors=0x65a5e0, reqcolors=0x100, pixelformat=0x65a5e4,
    bgcolor=0x0, transparent=0x65a5f8, allocator=0x652260) at loader.c:633
#7  0x0000000000410e2b in load_with_builtin (pchunk=<optimized out>, fstatic=0x0, fuse_palette=0x1, reqcolors=<optimized out>, bgcolor=<optimized out>, loop_control=0x0, fn_load=
<optimized out>, context=<optimized out>) at loader.c:889
#8  sixel_helper_load_image_file (filename=<optimized out>, fstatic=0x0, fuse_palette=0x1, reqcolors=0x100, bgcolor=<optimized out>, loop_control=0x0, fn_load=<optimized out>,
finsecure=<optimized out>, cancel_flag=<optimized out>,
    context=<optimized out>, allocator=<optimized out>) at loader.c:1418
#9  0x0000000000403d93 in sixel_encoder_encode (encoder=0x652290, filename=0x7fffffffe634 "/tmp/crashes/id:000001,sig:11,src:000624,op:havoc,rep:4") at encoder.c:1743
#10 0x000000000040272d in main (argc=<optimized out>, argc@entry=0x2, argv=<optimized out>, argv@entry=0x7fffffffe388) at img2sixel.c:457
#11 0x00007ffff6fb0b97 in __libc_start_main (main=0x402010 <main>, argc=0x2, argv=0x7fffffffe388, init=<optimized out>, fini=<optimized out>, rtld_fini=<optimized out>,
stack_end=0x7fffffffe378) at ../csu/libc-start.c:310
#12 0x0000000000401f4a in _start ()
```

carnil commented on Apr 13, 2020

CVE-2020-11721 was assigned for this issue.

peanuts62 commented on Apr 16, 2020

@carnil Hello I find a problem and request a CVE id ,but no reply to me?

#136

dotlambda mentioned this issue on Feb 1, 2021

libsixel: mark as insecure NixOS/nixpkgs#111579

Merged

ctrlcctrlv commented on Jun 9, 2021

I'm not sure of the validity of this bug, **@carnil**, **@sleicasper**.

The maintainer of this project has been absent for over a year and now I'm *de facto* maintainer. (See #154.)

I've patched the other CVE, see libsixel#8.

This one doesn't seem valid to me though. I need more information. I don't get the crash. I get a `libpng` error. Could `libpng` have updated in the meanwhile? Is this a libpng issue masquerading as a libsixel issue?

I get:

```
Starting program: /home/fred/Workspace/libsixel/converters/.libs/img2sixel -8 id:000001,sig:11,src:000624,op:havoc,rep:4
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib/libthread_db.so.1".
libpng error
UU[00][00]: invalid chunk type
[Inferior 1 (process 1827699) exited with code 0377]
quit)
```

If you can still repro the bug, is this a sufficient fix?

```
diff --git a/src/loader.c b/src/loader.c
index abc27f8..ea93daa 100644
--- a/src/loader.c
+++ b/src/loader.c
@@ -630,7 +630,9 @@ load_png(unsigned char      /* out */ **result,

 cleanup:
     png_destroy_read_struct(&png_ptr, &info_ptr,(png_infopp)0);
-    sixel_allocator_free(allocator, rows);
+    if (rows != NULL) {
+        sixel_allocator_free(allocator, rows);
+    }

     return status;
 }
```

If not, please give more info. If it's fixed, no big deal, I'll request CVE mark the bug invalid...

ctrlcctrlv mentioned this issue 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

sleicasper commented on Jun 9, 2021 Author

@ctrlcctrlv
which version of libsixel are you reproducing?

This bug is reproducible at the time of reporting, which is v1.8.6.

ctrlcctrlv commented on Jun 9, 2021

Please give a full command line. I'm attempting reproduction at 1.8.6 plus a bunch of PR's that should be irrelevant, call it libsixel@ `50206ad` .

ctrlcctrlv commented on Jun 9, 2021

Also, what is your libpng version? Mine is 1.6.37 on Arch Linux. My working theory right now is that this is not our bug, it's a libpng bug that they fixed recently which had the side effect of causing this bug. However, please feel free to explain to me how to reproduce so I can solve it if I'm wrong and close this out.

ctrlcctrlv commented on Jun 9, 2021 • edited ▾

Wait. Is what you're saying that if `HAVE_SETJMP && HAVE_LONGJMP` *is true that* can trigger the `goto cleanup` , which then can free a wild pointer (`rows`)?

That doesn't seem to be the executed code path, but I can fix it. Is this what you mean **@sleicasper** ?

sleicasper commented on Jun 9, 2021 Author

@ctrlcctrlv yes, that is what I mean. 'rows' can be a wild pointer. After executing 'goto cleanup', it can be freed.

ctrlcctrlv commented on Jun 9, 2021

Except that's not the executed codepath for your PoC? I don't mind patching it, but I don't understand how you got that from your PoC.

ctrlcctrlv commented on Jun 9, 2021

See? I get a libpng error.

```
:!make && LD_LIBRARY_PATH=./converters/.libs/:src/:src/.libs gdb ./converters/.libs/img2sixel -ex 'run
 id:000001,sig:11,src:000624,op:havoc,rep:4'



GNU gdb (GDB) 10.2
Copyright (C) 2021 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-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://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 ./converters/.libs/img2sixel...
Starting program: /home/fred/Workspace/libsixel/converters/.libs/img2sixel id:000001,sig:11,src:000624,op:havoc,rep:4
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib/libthread_db.so.1".
libpng error
UU[00][00]: invalid chunk type
[Inferior 1 (process 1852108) exited with code 0377]
(gdb) break png_error
Breakpoint 1 at 0x7ffff7e4d220
(gdb) r
Starting program: /home/fred/Workspace/libsixel/converters/.libs/img2sixel id:000001,sig:11,src:000624,op:havoc,rep:4
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib/libthread_db.so.1".

Breakpoint 1, 0x00007ffff7e4d220 in png_error () from /usr/lib/libpng16.so.16
(gdb) bt
#0  0x00007ffff7e4d220 in png_error () from /usr/lib/libpng16.so.16
#1  0x00007ffff7e4d315 in png_chunk_error () from /usr/lib/libpng16.so.16
#2  0x00007ffff7e5ff23 in ?? () from /usr/lib/libpng16.so.16
#3  0x00007ffff7e6009b in ?? () from /usr/lib/libpng16.so.16
#4  0x00007ffff7e52339 in png_read_info () from /usr/lib/libpng16.so.16
#5  0x00007ffff7f8b62f in load_png (result=0x5555555645b8,
    buffer=0x55555555c3c0 "\211PNG\r\n\032\n", size=8, psx=0x5555555645c8, psy=0x5555555645cc,
    ppalette=0x5555555645c0, pncolors=0x5555555645d0, reqcolors=256, pixelformat=0x5555555645d4,
    bgcolor=0x0, transparent=0x5555555645e8, allocator=0x55555555c2a0) at loader.c:365
#6  0x00007ffff7f98213 in load_with_builtin (pchunk=0x55555555c390, fstatic=fstatic@entry=0,
    fuse_palette=fuse_palette@entry=1, reqcolors=reqcolors@entry=256, bgcolor=bgcolor@entry=0x0,
    loop_control=loop_control@entry=0, fn_load=0x7ffff7f9bda0 <load_image_callback>,
    context=0x55555555c2d0) at loader.c:889
#7  0x00007ffff7f99324 in sixel_helper_load_image_file (
    filename=filename@entry=0x7fffffffe652 "id:000001,sig:11,src:000624,op:havoc,rep:4", fstatic=0,
    fuse_palette=fuse_palette@entry=1, reqcolors=256, bgcolor=0x0, loop_control=0,
    fn_load=0x7ffff7f9bda0 <load_image_callback>, finsecure=0,
    cancel_flag=0x55555555b0f0 <signaled>, context=0x55555555c2d0, allocator=0x55555555c2a0)
    at loader.c:1418
#8  0x00007ffff7f9d722 in sixel_encoder_encode (encoder=0x55555555c2d0,
    filename=0x7fffffffe652 "id:000001,sig:11,src:000624,op:havoc,rep:4") at encoder.c:1748
#9  0x0000555555555944 in main (argc=2, argv=0x7fffffffe298) at img2sixel.c:459
(gdb) frame 5
#5  0x00007ffff7f8b62f in load_png (result=0x5555555645b8,
    buffer=0x55555555c3c0 "\211PNG\r\n\032\n", size=8, psx=0x5555555645c8, psy=0x5555555645cc,
    ppalette=0x5555555645c0, pncolors=0x5555555645d0, reqcolors=256, pixelformat=0x5555555645d4,
    bgcolor=0x0, transparent=0x5555555645e8, allocator=0x55555555c2a0) at loader.c:365
365         png_read_info(png_ptr, info_ptr);
(gdb)
```

ctrlcctrlv mentioned this issue on Jun 9, 2021

CVE-2020-11721: load_png in loader.c in libsixel.a in libsixel 1.8.6 has an uninitialized pointer leading to an invalid call to free, which can cause a denial of service.
libsixel/libsixel#9

Closed

ctrlcctrlv commented on Jun 9, 2021

Anyway **@sleicasper**, I've merged libsixel@ `e71aacc` which I consider as closing this. It closes libsixel#9 in the fork. I've released version 1.9.0.

sleicasper commented on Jun 10, 2021 Author

It seems to be a `libpng` error that triggers the `longjmp` , then `libsixel` frees the wild pointer.

However, it looks like `libpng` have fixed it somehow. I couldn't reproduce this issue using default version of `libpng` on ubuntu 20.04 anymore.

The patch looks good to me, it should prevent `libsixel` freeing wild pointer no matter `libpng` has error.

❤️ 1

sleicasper closed this as completed on Jun 10, 2021

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

4 participants