Open Issue created 7 months ago by mqrsv

Tiff conversion to PS crashed due to incorrect memory size request

OS: ubuntu 20.04

LIBTIFF, Version 4.3.0

Command: ./tiff2ps -2 -a Poc.tiff

POC: example

ASAN Report: ==1860246==ERROR: AddressSanitizer: allocator is out of memory trying to allocate 0x2000000000 bytes #0 0x499d1d in __interceptor_malloc (/home/user/libtiff/tools/.libs/tiff2ps+0x499d1d) #1 0x7ffff7a46193 in _init (/lib/x86_64-linux-gnu/libjbig.so.0+0x1193) #2 (closed) 0x7ffff7db3407 (/home/user/libtiff/libtiff/.libs/libtiff.so.5+0x3407)

==1860246==HINT: if you don't care about these errors you may set allocator_may_return_null=1 SUMMARY: AddressSanitizer: out-of-memory (/home/user/libtiff/tools/.libs/tiff2ps+0x499d1d) in __interceptor_malloc ==1860246==ABORTING

Crashing thread backtrace:

#0 0x00007ffff78bb03b in __GI_raise (/lib/x86_64-linux-gnu/libc.so.6) at ../sysdeps/unix/sysv/linux/raise.c:50

#1 0x00007ffff789a859 in __GI_abort (/lib/x86_64-linux-gnu/libc.so.6) at abort.c:79

#2 (closed) 0x00007ffff7c5d1a3 in /lib/x86_64-linux-gnu/libjbig.so.0

#3 0x00007ffff7c62803 in jbg_dec_in (/lib/x86_64-linux-gnu/libjbig.so.0)

#4 (closed) 0x00000000004e3c85 in JBIGDecode (/home/user/libtiff/tools/tiff2ps) 49: int JBIGDecode(tif = (TIFF *)0x922bd0, buffer = (uint8_t *)*0x929940 "TZif2", size = (tmsize_t)147456, s = (uint16_t)) { ||: ||: / Local reference: int decodeStatus = 0; / ||: / Local reference: struct jbg_dec_state decoder = {d = 0, dl = 0, xd = 43136, yd = 3221258240, planes = 16, l0 = 49152, stripes = 65537, order = 0, options = 0, mx = 0, my = 0, dppriv = 0x0, ii = {0, 0, 0}, lhp = {0x925... / ||: / Local reference: TIFF * tif = 0x922bd0; / 76: #endif / HAVE_JBG_NEWLEN / 77: 78: decodeStatus = jbg_dec_in(&decoder, (unsigned char)tif->tif_rawcp, ||: --: } at tif_jbig.c:78

#5 0x0000000000660080 in TIFFReadEncodedStrip (/home/user/libtiff/tools/tiff2ps) 504: tmsize_t TIFFReadEncodedStrip(tif = (TIFF *)0x922bd0, strip = (uint32_t)0, buf = (void *)0x929940, size = (tmsize_t)147456) { |||: |||: / Local reference: TIFF * tif = 0x922bd0; / |||: / Local reference: uint32_t strip = 0; / |||: / Local reference: tmsize_t stripsize = ; / |||: / Local reference: uint16_t plane = 0; / |||: / Local reference: void * buf = 0x929940; */ 534: if (!TIFFFillStrip(tif,strip)) 535: return((tmsize_t)(-1)); 536: if ((*tif->tif_decodestrip)(tif,buf,stripsize,plane)<=0) |||: ---: } at tif_read.c:536

#6 0x00000000002c7ba1 in PS_Lvl2page (/home/user/libtiff/tools/tiff2ps) ????: int PS_Lvl2page(fd = (FILE *), tif = (TIFF *)0x922bd0, w = (uint32_t)0, h = (uint32_t)6) { ||||: ||||: / Local reference: tsize_t chunk_size = 147456; / ||||: / Local reference: tsize_t byte_count = ; / ||||: / Local reference: TIFF * tif = 0x922bd0; */ 2262: chunk_size); 2263: else 2264: byte_count = TIFFReadEncodedStrip(tif, ||||: ----: } at tiff2ps.c:2264

#7 (closed) 0x00000000002c7ba1 in PSpage (/home/user/libtiff/tools/tiff2ps) 2360: void PSpage(fd = (FILE *), tif = (TIFF *), w = (uint32_t), h = (uint32_t)) { ||||: ||||: / Local reference: char * imageOp = ; / ||||: / Local reference: FILE * fd = ; / ||||: / Local reference: TIFF * tif = ; / ||||: / Local reference: uint32_t w = ; / ||||: / Local reference: uint32_t h = ; */ 2365: imageOp = "imagemask"; 2366: 2367: if ((level2 || level3) && PS_Lvl2page(fd, tif, w, h)) ||||: ----: } at tiff2ps.c:2367

#8 (closed) 0x0000000000276ec1 in TIFF2PS (/home/user/libtiff/tools/tiff2ps) ????: int TIFF2PS(fd = (FILE *)0x7ffff7a656a0 <IO_2_1_stdout>, tif = (TIFF *)0x922bd0, pgwidth = (double)0, pgheight = (double)0, lm = (double)0, bm = (double)0, center = (int)) { ||||: ||||: / Local reference: double left_offset = 0; / ||||: / Local reference: FILE * fd = 0x7ffff7a656a0 <IO_2_1_stdout>; / ||||: / Local reference: double bottom_offset = 0; / ||||: / Local reference: double psheight = 6; / ||||: / Local reference: double scale = ; / ||||: / Local reference: double pswidth = 65535; */ 1079: case 180: fprintf (fd, "%f %f translate\n", left_offset ? left_offset : 0.0, 1080: bottom_offset ? bottom_offset : reqheight - (psheight * scale)); 1081: fprintf (fd, "%f %f scale\n1 1 translate 180 rotate\n", pswidth * scale, psheight * scale); ||||: ----: } at tiff2ps.c:1081

#9 (closed) 0x0000000000276ec1 in main (/home/user/libtiff/tools/tiff2ps) 260: int main(argc = (int)4, argv = (char *)*0x7fffffffdbe8) { |||: |||: / Local reference: int np = 32512; / |||: / Local reference: FILE * output = 0x7ffff7a656a0 <IO_2_1_stdout>; / |||: / Local reference: TIFF * tif = 0x922bd0; / |||: / Local reference: double pageWidth = 0; / |||: / Local reference: double pageHeight = 0; */ 500: return (EXIT_FAILURE); 501: } 502: np = TIFF2PS(output, tif, pageWidth, pageHeight, |||: ---: } at tiff2ps.c:502

Edited 7 months ago by mqrsv

Drag your designs here or click to upload.

Tasks 0

No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items 0

Link issues together to show that they're related or that one is blocking others. Learn more.

Activity

mqrsv changed the description 7 months ago ·

Su Laus @Su_Laus · 7 months ago — Developer

For me this is a bug in the "libjbig" library due to the corrupted example file and not a bug that could be fixed within libtiff.

Within jbig.c:2609 the value `s->yd` is set to `3221258240` and this big number is retrieved from the jbig coded data stream, which is corrupted in poc.tiff.

Petter Reinholdtsen @petterreinholdtsen · 5 months ago

For the record, this is https://security-tracker.debian.org/tracker/CVE-2022-1210 .

Please register or sign in to reply