Bug 701787 - heap-buffer-overflow at base/gdevm32.c:102 in mem_true32_fill_rectangle

Status: RESOLVED FIXED
Alias: None
Product: Ghostscript
Component: General (show other bugs)
Version: master
Hardware: PC Linux
Importance: P4 normal
Assignee: Ray Johnston
URL:
Keywords:
Depends on:
Blocks:

Reported: 2019-10-26 05:36 UTC by Suhwan
Modified: 2019-11-04 16:02 UTC (History)
CC List: 1 user (show)
See Also:
Customer:
Word Size: ---

Attachments	
poc (48.53 KB, application/pdf) 2019-10-26 05:36 UTC, Suhwan	Details
Add an attachment (proposed patch, testcase, etc.)	

Note
You need to log in before you can comment on or make changes to this bug.

Suhwan 2019-10-26 05:36:48 UTC Description

Created attachment 18371 [details]
poc

Hello.

I found a heap-buffer-overflow bug in GhostScript.

Please confirm.

Thanks.

OS: Ubuntu 18.04 64bit

Steps to reproduce:
1. Download the .POC files.
2. Compile the source code with ASan.
3. Run following cmd.
gs -dNOPAUSE -r345 -dFitPage -sPAPERSIZE=legal -sOutputFile=tmp -sDEVICE=cdj970
 $PoC

Here's ASAN report

```
==20883==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x7f78e6309130 at
pc 0x00000050e6d0 bp 0x7fff0d310810 sp 0x7fff0d30ffc0
WRITE of size 20400 at 0x7f78e6309130 thread T0
    #0 0x50e6cf in __interceptor_memset.part.39 (gs+0x50e6cf)
    #1 0x2c1e054 in mem_true32_fill_rectangle ghostpdl/./base/gdevm32.c:102:13
    #2 0x295395e in gx_dc_pure_fill_rectangle ghostpdl/./base/gxdcolor.c:800:16
    #3 0x2b7b84f in gx_default_fillpage ghostpdl/./base/gdevddrw.c:1091:16
    #4 0x14a38cb in clist_playback_band ghostpdl/./base/gxclrast.c:2156:29
    #5 0x14cd97b in clist_playback_file_bands ghostpdl/./base/gxclread.c:920:16
    #6 0x14d5c63 in clist_render_rectangle ghostpdl/./base/gxclread.c:854:16
    #7 0x14d413f in clist_rasterize_lines ghostpdl/./base/gxclread.c:743:20
    #8 0x14d2154 in clist_get_bits_rectangle ghostpdl/./base/gxclread.c:632:12
    #9 0x1596512 in clist_get_bits_rect_mt ghostpdl/./base/gxclthrd.c:860:13
    #10 0x2bae551 in gx_default_get_bits ghostpdl/./base/gdevdgbr.c:54:12
    #11 0x13f6b97 in gdev_prn_get_bits ghostpdl/./base/gdevprn.c:1687:16
    #12 0x13f6b97 in gdev_prn_copy_scan_lines ghostpdl/./base/gdevprn.c:1712
    #13 0x1e6ca6d in GetScanLine ghostpdl/./contrib/gdevdj9.c:928:5
    #14 0x1e52b91 in send_scan_lines ghostpdl/./contrib/gdevdj9.c:1014:12
    #15 0x1e52b91 in cdj970_print_page ghostpdl/./contrib/gdevdj9.c:891
    #16 0x13f07d9 in gx_default_print_page_copies ghostpdl/./base/gdevprn.c:1231:12
    #17 0x13ef028 in gdev_prn_output_page_aux ghostpdl/./base/gdevprn.c:1133:27
    #18 0x22b6f20 in gs_output_page ghostpdl/./base/gsdevice.c:212:17
    #19 0x3054b9f in zoutputpage ghostpdl/./psi/zdevice.c:416:12
    #20 0x2e8bdb6 in interp ghostpdl/./psi/interp.c:1300:28
    #21 0x2e8bdb6 in gs_call_interp ghostpdl/./psi/interp.c:520
    #22 0x2e8bdb6 in gs_interpret ghostpdl/./psi/interp.c:477
    #23 0x2e3f451 in gs_main_interpret ghostpdl/./psi/imain.c:253:12
    #24 0x2e3f451 in gs_main_run_string_end ghostpdl/./psi/imain.c:791
    #25 0x2e3f451 in gs_main_run_string_with_length ghostpdl/./psi/imain.c:735
    #26 0x2e548f0 in run_string ghostpdl/./psi/imainarg.c:1117:12
    #27 0x2e548f0 in runarg ghostpdl/./psi/imainarg.c:1086
    #28 0x2e5302a in argproc ghostpdl/./psi/imainarg.c:1008:16
    #29 0x2e479f7 in gs_main_init_with_args01 ghostpdl/./psi/imainarg.c:241:24
    #30 0x2e539d0 in gs_main_init_with_args ghostpdl/./psi/imainarg.c:288:16
    #31 0x57b86f in main ghostpdl/./psi/gs.c:95:16
    #32 0x7f78edd99b96 in __libc_start_main /build/glibc-OTsEL5/glibc-
2.27/csu/../csu/libc-start.c:310
    #33 0x482e79 in _start (gs+0x482e79)

0x7f78e6309130 is located 0 bytes to the right of 4000048-byte region
[0x7f78e5f38800,0x7f78e6309130)
allocated by thread T0 here:
    #0 0x542d30 in __interceptor_malloc (gs+0x542d30)
    #1 0x23640fd in gs_heap_alloc_bytes ghostpdl/./base/gsmalloc.c:193:34

SUMMARY: AddressSanitizer: heap-buffer-overflow (gs+0x50e6cf) in
__interceptor_memset.part.39
Shadow bytes around the buggy address:
  0x0fef9cc591d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fef9cc591e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fef9cc591f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fef9cc59200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fef9cc59210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0fef9cc59220: 00 00 00 00 00 00[fa]fa fa fa fa fa fa fa fa fa
  0x0fef9cc59230: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0fef9cc59240: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0fef9cc59250: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0fef9cc59260: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0fef9cc59270: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
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
==20883==ABORTING
```

Ken Sharp 2019-10-29 14:04:46 UTC Comment 1

Julian had a look at this one, there's some discussion in teh #artifex IRC logs on 29th October 2019 at around 12:42 which might be (somewhat) illuminating.

Robin's conclusion at 13:48 is that the bandheight is being calculated incorrectly, but that may not be the end of the story.

Robin Watts 2019-10-29 15:17:40 UTC Comment 2

clist_init_data is called several times with the device having a width of 2933. This (I believe) calculates the band height.

Then gx_device_set_hwsize_from_media is called from cdj970_one_time_initialisation, within cdj970_print_page and that changes the width to 5100.

When we then try to fill the page according to the width etc, we overrun the buffer.

Ray Johnston 2019-11-04 16:02:00 UTC Comment 3

Fixed in commit 4f73e8b4d578e69a17f452fa60d2130c5faaefd6