Bug 701818 - heap-use-after-free at devices/vector/gdevxps.c:1431 in xps_finish_image_path

Status: RESOLVED FIXED

Alias: None

Product: Ghostscript
Component: General (show other bugs)
Version: master
Hardware: PC Linux

Importance: P4 normal
Assignee: Chris Liddell (chrisl)

URL:
Keywords:

Depends on:
Blocks:

Reported: 2019-10-31 18:10 UTC by Suhwan
Modified: 2021-10-30 08:16 UTC (History)
CC List: 2 users (show)

See Also:
Customer:
Word Size: ---

Attachments	
poc (117.73 KB, application/pdf) 2019-10-31 18:10 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 2019-10-31 18:10:53 UTC Description

```
Created attachment 18402 [details]
poc

Hello

I found a heap-use-after-free bug in GhostScript.
Please confirm.
Thanks.

OS:        Ubuntu 18.04 64bit
Version:   commit b5bc53eb7223f8999882a5d8e2e35c27fe7a0b57

Steps to reproduce:
1. Download the .POC files.
2. Compile the source code with "make sanitize" using gcc.
3. Run following cmd.



gs -dBATCH -dNOPAUSE -dSAFER -dNOTRANSPARENCY -sOutputFile=tmp -sDEVICE=xpswrite
$PoC

Here's ASAN report.

==26756==ERROR: AddressSanitizer: heap-use-after-free on address 0x62a0005164b4 at
pc 0x557316b12cee bp 0x7ffe0d999530 sp 0x7ffe0d999520
READ of size 4 at 0x62a0005164b4 thread T0
    #0 0x557316b12ced in xps_finish_image_path devices/vector/gdevxps.c:1431
    #1 0x557316b13633 in xps_dorect devices/vector/gdevxps.c:1490
    #2 0x5573169df4ab in gdev_vector_dopath base/gdevvec.c:92
    #3 0x5573169e61b4 in gdev_vector_write_clip_path base/gdevvec.c:774
    #4 0x5573169e65ef in gdev_vector_update_clip_path base/gdevvec.c:807
    #5 0x5573169e9abe in gdev_vector_fill_path base/gdevvec.c:1171
    #6 0x557316b13ac4 in gdev_xps_fill_path devices/vector/gdevxps.c:1534
    #7 0x55731714488e in gx_fill_path base/gxpaint.c:53
    #8 0x557316dec6fe in do_fill base/gspaint.c:322
    #9 0x557316dec979 in fill_with_rule base/gspaint.c:356
    #10 0x557316deca2a in gs_fill base/gspaint.c:367
    #11 0x5573173e69f1 in zfill psi/zpaint.c:27
    #12 0x5573172e80e2 in do_call_operator psi/interp.c:86
    #13 0x5573172f296a in interp psi/interp.c:1410
    #14 0x5573172e9c2f in gs_call_interp psi/interp.c:520
    #15 0x5573172e92d4 in gs_interpret psi/interp.c:477
    #16 0x5573172bd82b in gs_main_interpret psi/imain.c:253
    #17 0x5573172c0ce0 in gs_main_run_string_end psi/imain.c:791
    #18 0x5573172c06a5 in gs_main_run_string_with_length psi/imain.c:735
    #19 0x5573172c0617 in gs_main_run_string psi/imain.c:716
    #20 0x5573172cd2db in run_string psi/imainarg.c:1117
    #21 0x5573172cd07e in runarg psi/imainarg.c:1086
    #22 0x5573172cc8fd in argproc psi/imainarg.c:1008
    #23 0x5573172c70c9 in gs_main_init_with_args01 psi/imainarg.c:241
    #24 0x5573172c752d in gs_main_init_with_args psi/imainarg.c:288
    #25 0x5573172d2a5d in psapi_init_with_args psi/psapi.c:272
    #26 0x5573174a207c in gsapi_init_with_args psi/iapi.c:148
    #27 0x5573160731d8 in main psi/gs.c:95
    #28 0x7f2e1bfd9b96 in __libc_start_main (/lib/x86_64-linux-
gnu/libc.so.6+0x21b96)
    #29 0x557316072f79 in _start (gs+0x36bf79)

0x62a0005164b4 is located 692 bytes inside of 22536-byte region
[0x62a000516200,0x62a00051ba08)
freed by thread T0 here:
    #0 0x7f2e1d8c37b8 in __interceptor_free (/usr/lib/x86_64-linux-
gnu/libasan.so.4+0xde7b8)
    #1 0x557316dd276c in gs_heap_free_object base/gsmalloc.c:358
    #2 0x557316d42315 in alloc_free_clump base/gsalloc.c:2636
    #3 0x557316d37ee5 in free_all_not_allocator base/gsalloc.c:994
    #4 0x557316d351ef in clump_splay_app base/gsalloc.c:606
    #5 0x557316d3818b in i_free_all base/gsalloc.c:1030
    #6 0x557317409d7e in restore_free psi/isave.c:989
    #7 0x557317408d3f in restore_space psi/isave.c:847
    #8 0x557317408723 in alloc_restore_step_in psi/isave.c:784
    #9 0x55731736d27e in dorestore psi/zvmem.c:173
    #10 0x5573172b36f0 in z2restore psi/zdevice2.c:373
    #11 0x5573172e80e2 in do_call_operator psi/interp.c:86
    #12 0x5573172f4d13 in interp psi/interp.c:1674
    #13 0x5573172e9c2f in gs_call_interp psi/interp.c:520
    #14 0x5573172e92d4 in gs_interpret psi/interp.c:477
    #15 0x5573172bd82b in gs_main_interpret psi/imain.c:253
    #16 0x5573172c0ce0 in gs_main_run_string_end psi/imain.c:791
    #17 0x5573172c06a5 in gs_main_run_string_with_length psi/imain.c:735
    #18 0x5573172c0617 in gs_main_run_string psi/imain.c:716
    #19 0x5573172cd2db in run_string psi/imainarg.c:1117
    #20 0x5573172cd07e in runarg psi/imainarg.c:1086
    #21 0x5573172cc8fd in argproc psi/imainarg.c:1008
    #22 0x5573172c70c9 in gs_main_init_with_args01 psi/imainarg.c:241
    #23 0x5573172c752d in gs_main_init_with_args psi/imainarg.c:288
    #24 0x5573172d2a5d in psapi_init_with_args psi/psapi.c:272
    #25 0x5573174a207c in gsapi_init_with_args psi/iapi.c:148
    #26 0x5573160731d8 in main psi/gs.c:95
    #27 0x7f2e1bfd9b96 in __libc_start_main (/lib/x86_64-linux-
gnu/libc.so.6+0x21b96)

previously allocated by thread T0 here:
    #0 0x7f2e1d8c3b50 in __interceptor_malloc (/usr/lib/x86_64-linux-
gnu/libasan.so.4+0xdeb50)
    #1 0x557316dd1826 in gs_heap_alloc_bytes base/gsmalloc.c:193
    #2 0x557316d4117b in alloc_acquire_clump base/gsalloc.c:2485
    #3 0x557316d3e422 in alloc_obj base/gsalloc.c:1948
    #4 0x557316d399d9 in i_alloc_struct base/gsalloc.c:1231
    #5 0x557316b15b1d in xps_begin_image devices/vector/gdevxps.c:1835
    #6 0x5573171a104b in gx_default_begin_typed_image base/gdevddrw.c:1044
    #7 0x557316dba1ab in gs_image_begin_typed base/gsimage.c:258
    #8 0x5573173de8ea in zimage_setup psi/zimage.c:180
    #9 0x5573173deff9 in zimage1 psi/zimage.c:243
    #10 0x5573172e80e2 in do_call_operator psi/interp.c:86
    #11 0x5573172f1861 in interp psi/interp.c:1300
    #12 0x5573172e9c2f in gs_call_interp psi/interp.c:520
    #13 0x5573172e92d4 in gs_interpret psi/interp.c:477
    #14 0x5573172bd82b in gs_main_interpret psi/imain.c:253
    #15 0x5573172c0ce0 in gs_main_run_string_end psi/imain.c:791
    #16 0x5573172c06a5 in gs_main_run_string_with_length psi/imain.c:735
    #17 0x5573172c0617 in gs_main_run_string psi/imain.c:716
    #18 0x5573172cd2db in run_string psi/imainarg.c:1117
    #19 0x5573172cd07e in runarg psi/imainarg.c:1086
    #20 0x5573172cc8fd in argproc psi/imainarg.c:1008
    #21 0x5573172c70c9 in gs_main_init_with_args01 psi/imainarg.c:241
    #22 0x5573172c752d in gs_main_init_with_args psi/imainarg.c:288
    #23 0x5573172d2a5d in psapi_init_with_args psi/psapi.c:272
    #24 0x5573174a207c in gsapi_init_with_args psi/iapi.c:148
    #25 0x5573160731d8 in main psi/gs.c:95
    #26 0x7f2e1bfd9b96 in __libc_start_main (/lib/x86_64-linux-
gnu/libc.so.6+0x21b96)

SUMMARY: AddressSanitizer: heap-use-after-free devices/vector/gdevxps.c:1431 in
xps_finish_image_path
Shadow bytes around the buggy address:
  0x0c548009ac40: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c548009ac50: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c548009ac60: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c548009ac70: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c548009ac80: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
=>0x0c548009ac90: fd fd fd fd fd fd[fd]fd fd fd fd fd fd fd fd fd
  0x0c548009aca0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c548009acb0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c548009acc0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c548009acd0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c548009ace0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
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
```

Ken Sharp 2019-10-31 19:15:12 UTC Comment 1

This might need to be re-assigned ot either myself or Henry. This assignmen is just for an initial triage and to make suer it doesn't get forgotten.

Chris Liddell (chrisl) 2019-11-05 11:00:31 UTC Comment 2

Fixed in:

https://git.ghostscript.com/?p=ghostpdl.git;a=commitdiff;h=94d8955cb77

Todd 2020-08-26 19:20:26 UTC Comment 3

@Ken @Chris note that I tested this on ghostscript-9.25 with the PoC here and I got an entirely different backtrace:

```
==1298203==ERROR: AddressSanitizer: heap-use-after-free on address 0x62a000678250
at pc 0x000002664563 bp 0x7ffc94c166a0 sp 0x7ffc94c16690
[16/7539]
READ of size 4 at 0x62a000678250 thread T0
    #0 0x2664562 in igc_reloc_struct_ptr psi/igc.c:1279
    #1 0x1ccd294 in basic_reloc_ptrs base/gsmemory.c:347
    #2 0x26683fc in gc_do_reloc psi/igc.c:1246
    #3 0x266c017 in gs_gc_reclaim psi/igc.c:450
    #4 0x27764da in context_reclaim psi/zcontext.c:290
    #5 0x2518dcc in gs_vmreclaim psi/ireclaim.c:163
    #6 0x2518dcc in ireclaim psi/ireclaim.c:80
    #7 0x24f2b8c in interp_reclaim psi/interp.c:447
    #8 0x24bd784 in gs_main_finit psi/imain.c:914
    #9 0x53174e in main psi/gs.c:138
    #10 0x7f29b4ca71a2 in __libc_start_main ../csu/libc-start.c:308
    #11 0x53c28d in _start (/home/moveax41h/analysis/dist-
git/ghostscript/ghostscript-9.25/bin/gs+0x53c28d)

0x62a000678250 is located 80 bytes inside of 22536-byte region
[0x62a000678200,0x62a00067da08)
freed by thread T0 here:
    #0 0x7f29b5c9291f in __interceptor_free (/lib64/libasan.so.5+0x10d91f)
    #1 0x1bd5720 in alloc_free_clump base/gsalloc.c:2599

previously allocated by thread T0 here:
    #0 0x7f29b5c92d18 in __interceptor_malloc (/lib64/libasan.so.5+0x10dd18)
    #1 0x1cb97ae in gs_heap_alloc_bytes base/gsmalloc.c:193

SUMMARY: AddressSanitizer: heap-use-after-free psi/igc.c:1279 in
igc_reloc_struct_ptr
Shadow bytes around the buggy address:
  0x0c54800c6ff0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c54800c7000: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c54800c7010: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c54800c7020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c54800c7030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c54800c7040: fd fd fd fd fd fd fd fd fd fd[fd]fd fd fd fd fd
  0x0c54800c7050: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c54800c7060: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c54800c7070: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c54800c7080: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c54800c7090: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
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
==1298203==ABORTING
```

I'm not yet sure why or if the implications here are related or coincidental and this is an entirely separate use-after-free.

Chris Liddell (chrisl)   2020-08-27 07:48:56 UTC   **Comment 4**

(In reply to Todd from comment #3)
<SNIP>

> I'm not yet sure why or if the implications here are related or coincidental
> and this is an entirely separate use-after-free.

It's almost certainly a different problem, so please don't add new problems to existing (and especially closed) bugs.

FWIW, I cannot reproduce what you see with the current code, 9.52 nor the pending 9.53 release code, building with clang 10.

Format For Printing - XML - Clone This Bug - Top of page