Bug 702851 - use-after-free vulnerability in igc_reloc_struct_ptr() from PDF file

Status: RESOLVED WORKSFORME

Alias: None

Product: Ghostscript
Component: Security (public) (show other bugs)
Version: 9.25
Hardware: PC Linux

Importance: P4 normal
Assignee: Chris Liddell (chrisl)

URL:
Keywords:

Depends on:
Blocks:

Reported: 2020-08-28 19:59 UTC by Todd
Modified: 2020-09-02 07:08 UTC (History)
CC List: 7 users (show)

See Also:
Customer:
Word Size: ---

Attachments

Add an attachment (proposed patch, testcase, etc.)

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

Todd 2020-08-28 19:59:55 UTC Description

```
Overview:

When testing for BZ#701818 in ghostscript-9.25, I found a use-after-free which
looks unrelated to that bug. It occurs when the crafted PDF PoC file is provided as
input to ghostscript.

Steps to reproduce:

1. Download https://bugs.ghostscript.com/attachment.cgi?id=18402
2. Compile ghostscript with AddressSanitizer
3. run:

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

Expected result:

ghostscript displays an error and exits, or otherwise handles the input

Actual result:

ASAN shows that ghostscript performs a use-after-free:

==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


Environmental information:
OS: Fedora 31
Software: ghostscript-9.25
Compiler: GCC 9.3.1

Notes:

I'm aware this is an old version of ghostscript, but wanted to share this for
```

informational purposes. I did see BZ#695318, but the backtrace is a bit different there and looks like it could be a different issue.

Ray Johnston 2020-08-28 20:44:48 UTC **Comment 1**

Please re-test with the 9.53.0 Release Candidate:

https://github.com/ArtifexSoftware/ghostpdl-downloads/releases/tag/ghostpdl-9.53.0rc2

If it is reproducible, then re-open this bug (change the "Status" to confirmed)

We really don't appreciate someone submitting bugs for old versions -- it takes time, even if it just asking you to test with the latest available (in this case the 9.53.0 Release candidate). Wouldn't you rather have us spending time on things that are actually still problems.

chris-liddell released Ghostscript/GhostPDL 9.25 on Sep 13, 2018

Ken Sharp 2020-08-29 12:56:35 UTC **Comment 2**

(In reply to Todd from comment #0)

> When testing for BZ#701818 in ghostscript-9.25, I found a use-after-free
> which looks unrelated to that bug. It occurs when the crafted PDF PoC file
> is provided as input to ghostscript.
>
> Steps to reproduce:
>
> 1. Download https://bugs.ghostscript.com/attachment.cgi?id=18402
> 2. Compile ghostscript with AddressSanitizer
> 3. run:
>
> gs -dBATCH -dNOPAUSE -dSAFER -dNOTRANSPARENCY -sOutputFile=tmp
> -sDEVICE=xpswrite $PoC

In order to reproduce the problem it is necessary to use the 9.25 released code, later releases do not exhibit the issue. The 9.25 release was the 13th September 2018.

In ~~bug 701818~~ we can see from the address sanitizer stack that this particular use-after-free isn't present. Now that report was against a SHA representing a commit between releases, in fact from 31st October 2019.

So this tells us that the problem had already been fixed by this point. Using Git bisect I find that the relevant commit was this one:

https://git.ghostscript.com/?p=ghostpdl.git;a=commit;h=ece5cbbd9979cd35737b00e68267762d72feb2ea

This was not *intended* to fix this specific problem, but it clearly does.

I would highly recommend using Git bisect in cases like this, its quick (on Linux) it only took me an hour or so to locate this commit, and will find cases like this where a problem was unwittingly fixed as a consequence of something else.

Obviously its up to you whether you choose to take on that commit in order to resolve the use-after-free problem, from our point of view the problem has already been fixed in our current code base.

I've changed the resolution to 'worksforme' as there was a problem here, but its already been resolved.

Todd 2020-09-01 16:13:10 UTC **Comment 3**

Thanks for looking into it. My intention of course was mostly to inform about the bug for documentation purposes, knowing that this is not even close to the current version.

Appreciate the reference to Git Bisect as this is a useful tool I can add to my toolbox, and the specific commit you referenced.

Just to be comprehensive, I checked it with 9.53.0RC2 in my environment and it did not trigger a use-after-free. Instead, I got this output:

```
==========================================
~/Downloads/ghostpdl-9.53.0rc2/sanbin $ ./gs -dBATCH -dNOPAUSE -dSAFER -
dNOTRANSPARENCY -sOutputFile=tmp -sDEVICE=xpswrite ~/Downloads/xps_finish_poc.pdf
GPL Ghostscript RELEASE CANDIDATE 2 9.53.0 (2020-08-27)
Copyright (C) 2020 Artifex Software, Inc.  All rights reserved.
This software is supplied under the GNU AGPLv3 and comes with NO WARRANTY:
see the file COPYING for details.
Processing pages 1 through 3.
Page 1
   **** Error: Error reading font stream, attempting to load the font using its
name
               Output may be incorrect.
Querying operating system for font files...
Substituting font Times-Roman for EPKNCP+TimesNewRomanPSMT.
Loading NimbusRoman-Regular font from %rom%Resource/Font/NimbusRoman-Regular...
4467604 2911804 4167968 2833129 4 done.
Page 2
Substituting font Helvetica for Tahoma.
Loading NimbusSans-Regular font from %rom%Resource/Font/NimbusSans-Regular...
4533716 3085400 4369968 2999819 4 done.
Substituting font Helvetica for ArialMT.
   **** Error reading a content stream. The page may be incomplete.
               Output may be incorrect.
   **** Error reading a content stream. The page may be incomplete.
               Output may be incorrect.
Page 3
Substituting font Helvetica for Tahoma.
   **** Error reading a content stream. The page may be incomplete.
               Output may be incorrect.

   **** This file had errors that were repaired or ignored.
   **** The file was produced by:
   **** >>>> Acrobat Distiller 9.5.2 (Windows) <<<<
   **** Please notify the author of the software that produced this
   **** file that it does not conform to Adobe's published PDF
   **** specification.

   **** The rendered output from this file may be incorrect.

====================================
```

So ghostscript of course properly catches the problem in 9.53RC2 as you stated.

Is it ok to make this public at this point?

Ken Sharp 2020-09-01 16:20:49 UTC **Comment 4**

(In reply to Todd from comment #3)
> Thanks for looking into it. My intention of course was mostly to inform
> about the bug for documentation purposes, knowing that this is not even
> close to the current version.

Technically it is (I think) *just* still supported for customers (not quite 2 years yet).

> Is it ok to make this public at this point?

Well its been 'fixed', even if its incidentally, for 18 months so I can't see any reason why not. Might be nice to wait for Chris to give it his blessing, seeing as how he owns the bug, but he's on vacation today. Do you mind waiting until tomorrow for a response from him ?

Todd 2020-09-01 16:39:53 UTC **Comment 5**

(In reply to Ken Sharp from comment #4)
> Do you mind waiting until tomorrow for a response from him ?

Absolutely can wait. Thanks.

Chris Liddell (chrisl) 2020-09-02 07:08:22 UTC **Comment 6**

I don't see any issue making it public.