New issue

Jump to bottom

Null Pointer dereference caused by incomplete check of the return value from libxml2 in ReadSVGImage svg.c:3621 #2624

Closed 5hadowblad3 opened this issue on Sep 25, 2020 · 1 comment

5hadowblad3 commented on Sep 25, 2020 • edited

Prerequisites

- [Y] I have written a descriptive issue title
- [Y] I have verified that I am using the latest version of ImageMagick
- [Y] I have searched open and closed issues to ensure it has not already been reported.

Description

There is a segmentation fault caused by the NPD in function ReadSVGImage, svg.c:3621 in ImageMagick 7.0.10.
ImageMagick does not check the nullity of the pointer returned from libxml2 and dereference it directly.
This directly leads to program crashes and segmentation fault.

Steps to Reproduce

1, To ensure reproduce, I use up space in the /tmp folder as a low-level privilege user.
For example, to facilitate the reproducation,

```
fallocate -l size_of_the_tmp_folder /tmp/test.img
```

2, Run:

```
magick convert poc ./test.ps
```

seg-svg3621.zip (unzip first)

Here is the trace reported by ASAN:

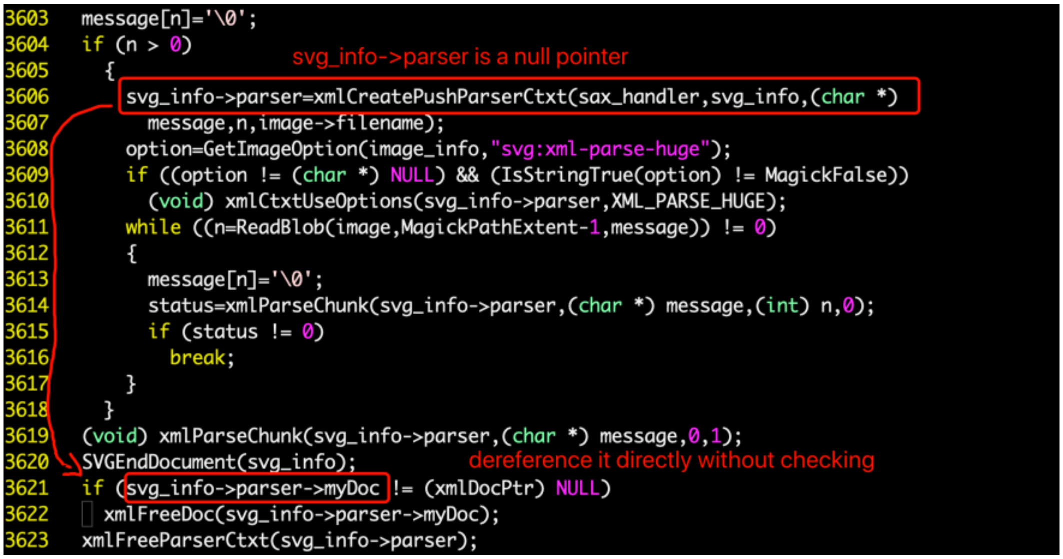
```
ASAN:SIGSEGV
=================================================================
==112350==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000010 (pc 0x0000009fec8c bp 0x62700001f900 sp 0x7ffd383e31c0 T0)
    #0 0x9fec8b in ReadSVGImage ../coders/svg.c:3621
    #1 0xc8ba0c in ReadImage ../MagickCore/constitute.c:553
    #2 0x8dfbc1 in ReadPESImage ../coders/pes.c:673
    #3 0xc8ba0c in ReadImage ../MagickCore/constitute.c:553
    #4 0xc8ecbc in ReadImages ../MagickCore/constitute.c:943
    #5 0x12bfaef in ConvertImageCommand ../MagickWand/convert.c:607
    #6 0x13fd865 in MagickCommandGenesis ../MagickWand/mogrify.c:191
    #7 0x43992d in MagickMain ../utilities/magick.c:149
    #8 0x7efd17fa982f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #9 0x439168 in _start (/mnt/data/playground/ImageMagick/build-asan/utilities/magick+0x439168)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV ../coders/svg.c:3621 ReadSVGImage
==112350==ABORTING
```

System Configuration

```
CFLAGS='-I/usr/include/libxml2 -I/usr/include/libpng12  -I/usr/include/openjpeg-2.1 -I/usr/include/freetype2 -I/usr/include/freetype2  -fopenmp -Wall -O0 -g -fsanitize=address  -mtune=broadwell -fexceptions -pthread -DMAGICKCORE_HDRI_ENABLE=1 -DMAGICKCORE_QUANTUM_DEPTH=16'
```

- ImageMagick version:
  Version: ImageMagick 7.0.10-31 Q16 x86_64 2020-09-22 https://imagemagick.org
  Copyright: © 1999-2020 ImageMagick Studio LLC
  License: https://imagemagick.org/script/license.php
  Features: Cipher DPC HDRI OpenMP(4.0)
  Delegates (built-in): bzlib fontconfig freetype jng jp2 jpeg lzma png x xml zlib
- Environment (Operating system, version and so on):
  DISTRIB_ID=Ubuntu
  DISTRIB_RELEASE=16.04
  DISTRIB_CODENAME=xenial
  DISTRIB_DESCRIPTION="Ubuntu 16.04.6 LTS"
- Additional information:



```
message[n]='\0';
if (n > 0)
  {
    svg_info->parser=xmlCreatePushParserCtxt(sax_handler,svg_info,(char *)
      message,n,image->filename);
    option=GetImageOption(image_info,"svg:xml-parse-huge");
    if ((option != (char *) NULL) && (IsStringTrue(option) != MagickFalse))
      (void) xmlCtxtUseOptions(svg_info->parser,XML_PARSE_HUGE);
    while ((n=ReadBlob(image,MagickPathExtent-1,message)) != 0)
    {
      message[n]='\0';
      status=xmlParseChunk(svg_info->parser,(char *) message,(int) n,0);
      if (status != 0)
        break;
    }
  }
(void) xmlParseChunk(svg_info->parser,(char *) message,0,1);
SVGEndDocument(svg_info);
if (svg_info->parser->myDoc != (xmlDocPtr) NULL)
  xmlFreeDoc(svg_info->parser->myDoc);
xmlFreeParserCtxt(svg_info->parser);
```

svg_info->parser is a null pointer

dereference it directly without checking

Here is the link for the function xmlCreatePushParserCtxt in libxml2,
which indicates the return value can be NULL if fails.
https://gitlab.gnome.org/GNOME/libxml2/-/blob/master/parser.c#L12375

```
/**
 * xmlCreatePushParserCtxt:
 * @sax:  a SAX handler
 * @user_data:  The user data returned on SAX callbacks
 * @chunk:  a pointer to an array of chars
 * @size:  number of chars in the array
 * @filename:  an optional file name or URI
 *
 * Create a parser context for using the XML parser in push mode.
 * If @buffer and @size are non-NULL, the data is used to detect
 * the encoding.  The remaining characters will be parsed so they
 * don't need to be fed in again through xmlParseChunk.
 * To allow content encoding detection, @size should be >= 4
 * The value of @filename is used for fetching external entities
 * and error/warning reports.
 *
 * Returns the new parser context or NULL
 */
```

dlemstra added a commit that referenced this issue on Sep 25, 2020

`Handle null pointer return from call to xmlCreatePushParserCtxt (#2624).` 43dfb18

dlemstra commented on Sep 25, 2020 Member

Thanks for the detailed report. We just pushed a patch to resolve this.

dlemstra closed this as completed on Sep 25, 2020

Assignees

No one assigned

Labels

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants