Closed Issue created 8 months ago by **frokaikan**

heap-buffer-overflow on creating a face with strange file and invalid index

I compile freetype with ASAN and call `FT_New_Face` with the following code:

```
#include "ft2build.h"
#include FT_FREETYPE_H

int main (int argc, char **argv) {
    FT_Library lib;
    FT_Face face;

    FT_Init_FreeType(&lib);
    FT_New_Face(lib, argv[1], -4939615758108852224, &face);
}
```

and run with this file testface

I think `FT_New_Face` should return a non-zero value. However, ASAN reports like following:

```
ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000088 at pc 0x7fd51dd1048d bp 0x7f
READ of size 8 at 0x602000000088 thread T0
    #0 0x7fd51dd1048c in sfnt_init_face
    #1 0x7fd51dd60277 in tt_face_init
    #2 0x7fd51db84bf1 in open_face
    #3 0x7fd51db583c2 in ft_open_face_internal
    #4 0x7fd51db5764a in FT_New_Face
    #5 0x4c6c0b in main
    #6 0x7fd51d4f80b2 in __libc_start_main
    #7 0x41c2fd in _start

0x602000000088 is located 8 bytes to the left of 8-byte region [0x602000000090,0x602000000098)
allocated by thread T0 here:
    #0 0x494a3d in malloc
    #1 0x7fd51dbb8ea4 in ft_alloc
    #2 0x7fd51db7c1b1 in ft_mem_qalloc
    #3 0x7fd51db4cdb3 in ft_mem_alloc
    #4 0x7fd51dd298f3 in sfnt_open_font
    #5 0x7fd51dd1018c in sfnt_init_face
    #6 0x7fd51dd60277 in tt_face_init
    #7 0x7fd51db84bf1 in open_face
    #8 0x7fd51db583c2 in ft_open_face_internal
    #9 0x7fd51db5764a in FT_New_Face
    #10 0x4c6c0b in main
    #11 0x7fd51d4f80b2 in __libc_start_main

SUMMARY: AddressSanitizer: heap-buffer-overflow in sfnt_init_face
Shadow bytes around the buggy address:
  0x0c047fff7fc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff8000: fa fa 00 fa fa fa 00 fa fa fa 00 fa fa fa 00 fa
=>0x0c047fff8010: fa[fa]00 fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8060: 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
  Shadow gap:              cc
==3906462==ABORTING
```

Edited 8 months ago by frokaikan

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

Activity

frokaikan changed the description 8 months ago

Werner Lemberg closed via commit `53dfdcd8` 8 months ago

Werner Lemberg @wl · 8 months ago Owner

Thanks for the report, fixed.

frokaikan mentioned in issue #1139 (closed) 8 months ago

Werner Lemberg added Bug label 7 months ago

Brian Hop mentioned in issue #1152 (closed) 6 months ago

ABHISHEK PALIWAL @abhishpaliwal · 6 months ago

can anyone tell me the commit id, with which I can reproduce it? As I am trying with below steps but not able to reproduce it also not able to reproduce #1139 (closed) and #1140 (closed)

1. ./autogen.sh
2. ./configure CFLAGS="-fsanitize=address -g" CXXFLAGS="-fsanitize=address -g" LDFLAGS="-fsanitize=address -static-libasan -g"
3. make && make install
4. gcc poc.c -I /usr/local/include/freetype2 -L /usr/local/lib -lfreetype -fsanitize=address
5. ./a.out testface

After Step 5 getting the below logs:

$ ./a.out testface

===
==350188==ERROR: LeakSanitizer: detected memory leaks

Indirect leak of 1848 byte(s) in 28 object(s) allocated from: #0 0x7feb20648808 in __interceptor_malloc ../../../../src/libsanitizer/asan/asan_malloc_linux.cc:144 #1 (closed) 0x7feb2048fb48 (/usr/lib/x86_64-linux-gnu/libfreetype.so.6+0x13b48)

Indirect leak of 368 byte(s) in 1 object(s) allocated from: #0 0x7feb20648808 in __interceptor_malloc ../../../../src/libsanitizer/asan/asan_malloc_linux.cc:144 #1 (closed) 0x7feb2048fb48 (/usr/lib/x86_64-linux-gnu/libfreetype.so.6+0x13b48) #2 (closed) 0xe8ffc4342c417fff ()

SUMMARY: AddressSanitizer: 2216 byte(s) leaked in 29 allocation(s).

freetype version I am having is : VER-2-10-1

Regards, Abhishek

Edited by ABHISHEK PALIWAL 6 months ago

frokaikan @frokaikan · 6 months ago Author

The three bugs have been fixed in the newest version of freetype2... At least I can not reproduce all of them.

`53dfdcd8` fixes this bug, which previous commit is `1e2eb650`.

`22a0cccb` fixes #1139 (closed), which previous is `53dfdcd8`.

`0c2bdb01` fixes #1140 (closed), which previous is `d014387a`.

Edited by frokaikan 6 months ago

Alexei Podtelezhnikov @apodtele · 6 months ago Maintainer

FreeType 2.10.1 was released 3 years ago. Most repos moved on.

ABHISHEK PALIWAL @abhishpaliwal · 6 months ago

@frokaikan how can we check whether our version of freetype is vulnerable or not with these CVEs?

ABHISHEK PALIWAL @abhishpaliwal · 6 months ago

Any suggestion to verify?

Alexei Podtelezhnikov @apodtele · 6 months ago Maintainer

1. Assume it's verified because 2.10.1 < 2.12.1, or
2. Hire a consultant

Please register or sign in to reply

Please register or sign in to reply