AddressSanitizer: SEGV on unknown address 0x000000000014 #415

Closed chibataiki opened this issue on Jan 26, 2021 · 7 comments

Assignees

Labels bug priority-high

Milestone Stable

chibataiki commented on Jan 26, 2021 • edited

Hello, While fuzzing htmldoc , I found SEGV on unknown address

test platform
htmldoc Version 1.9.12 git [master `6898d0a`]
OS :Ubuntu 20.04.1 LTS x86_64
kernel: 5.4.0-53-generic
compiler: clang version 10.0.0-4ubuntu1
reproduced:

```
htmldoc -f demo.pdf   poc4.html
```

poc(zipped for update):

poc4.zip

```
=================================================================
==38160==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000014 (pc 0x7fd7b98ce259 bp 0x000000000001 sp 0x7ffcf67f15c0 T0)
==38160==The signal is caused by a WRITE memory access.
==38160==Hint: address points to the zero page.
    #0 0x7fd7b98ce258  (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x21258)
    #1 0x7fd7b98cbf1e  (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x1ef1e)
    #2 0x7fd7b98c3f2e in jpeg_consume_input (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x16f2e)
    #3 0x7fd7b98c41b1 in jpeg_read_header (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x171b1)
    #4 0x5c06dd in image_load_jpeg(image_t*, _IO_FILE*, int, int) /home//htmldoc_sani/htmldoc/image.cxx:1357:3
    #5 0x5c06dd in image_load /home//htmldoc_sani/htmldoc/image.cxx:824
    #6 0x5a8f6f in compute_size(tree_str*) /home//htmldoc_sani/htmldoc/htmllib.cxx:3239:11
    #7 0x5a1d63 in htmlReadFile /home//htmldoc_sani/htmldoc/htmllib.cxx:981:11
    #8 0x53eb98 in read_file(char const*, tree_str**, char const*) /home//htmldoc_sani/htmldoc/htmldoc.cxx:2492:9
    #9 0x539ce3 in main /home//htmldoc_sani/htmldoc/htmldoc.cxx:1177:7
    #10 0x7fd7b93610b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/csu/../csu/libc-start.c:308:16
    #11 0x41f8bd in _start (/home//htmldoc_sani/htmldoc/htmldoc+0x41f8bd)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x21258)
==38160==ABORTING
```

```
[#0] 0x7ffff7ef5259 → mov DWORD PTR [rbx+0x14], r14d
[#1] 0x7ffff7ef2f1f → mov r12d, eax
[#2] 0x7ffff7eeaf2f → jpeg_consume_input()
[#3] 0x7ffff7eeb1b2 → jpeg_read_header()
[#4] 0x5c06de → image_load_jpeg(img=0x619000000080, fp=<optimized out>, gray=<optimized out>, load_data=0x0)
[#5] 0x5c06de → image_load(filename=0x603000000190 "/var/tmp/041944.000001.tmp", gray=0x0, load_data=0x0)
[#6] 0x5a8f70 → compute_size(t=0x608000001c20)
[#7] 0x5a1d64 → htmlReadFile(parent=<optimized out>, fp=0x615000000300, base=0x7fffffffcbe0 "./pocs_htmldoc")
[#8] 0x53eb99 → read_file(filename=<optimized out>, document=0x7fffffffd180, path=<optimized out>)
[#9] 0x539ce4 → main(argc=0x4, argv=0x7fffffffdd38)
```

reporter: chiba of topsec alphalab

michaelrsweet self-assigned this on Jan 26, 2021

michaelrsweet added bug priority-high labels on Jan 26, 2021

michaelrsweet added this to the **Stable** milestone on Jan 26, 2021

michaelrsweet commented on Jan 26, 2021 Owner

Confirmed, investigating...

michaelrsweet commented on Apr 1, 2021 Owner

This crash is happening in libjpeg, so you need to provide the IJG a copy of the JPEG file so they can fix this.

I am also testing this against libjpeg-turbo, which will be in the next release of HTMLDOC as the embedded/local JPEG library...

michaelrsweet commented on Apr 1, 2021 Owner

... and it looks like the same issue is present in libjpeg-turbo as well.

michaelrsweet commented on Apr 1, 2021 Owner

OK, so for some reason the setjmp exception handling is not working - looking further because the library is throwing an error but not stopping processing.

michaelrsweet added a commit that referenced this issue on Apr 1, 2021

`Fix JPEG error handling (Issue #415)` ✕ `369b2ea`

michaelrsweet commented on Apr 1, 2021 Owner

OK, looks like I didn't actually implement the longjmp in the JPEG error handler. So both versions of libjpeg catch the error but allow the crash if you don't abort the read... fun...

[master `369b2ea`] Fix JPEG error handling (Issue #415)

michaelrsweet closed this as completed on Apr 1, 2021

---

michaelrsweet mentioned this issue on Apr 1, 2021

AddressSanitizer: double-free in function pspdf_export ps-pdf.cxx:945:7 #414

Closed

chibataiki commented on Apr 1, 2021 Author

Thanks for the fix.

chibataiki commented on Feb 21 Author

CVE-2021-23191 assigned

Assignees

michaelrsweet

Labels

bug priority-high

Projects

None yet

Milestone

Stable

Development

No branches or pull requests

2 participants