

New issue

Jump to bottom

## AddressSanitizer: heap-buffer-overflow on render\_table\_row() ps-pdf.cxx:6123:34 #417

Closed

chibataiki opened this issue on Jan 26, 2021 · 3 comments

Assignees



Labels

bug

priority-high

Milestone

Stable

chibataiki commented on Jan 26, 2021 • edited

Hello, While fuzzing htmldoc, I found a heap-buffer-overflow in the render\_table\_row() ps-pdf.cxx:6123:34

- test platform  
htmldoc Version 1.9.12 git [master 6898d8a]  
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 poc7.html

poc(zipped for update):

[poc7.zip](#)

```
=====
==38248==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x625000002100 at pc 0x00000059260e bp 0x7fffa3362670 sp 0x7fffa3362668
READ of size 8 at 0x625000002100 thread T0
#0 0x59260d in render_table_row(hdtable_t&, tree_str***, int, unsigned char*, float, float, float, float, float*, float*, int*) /home/htmldoc_sani/htmldoc/ps-pdf.cxx:6123:34
#1 0x588630 in parse_table(tree_str*, float, float, float, float, float*, float*, int*, int) /home/htmldoc_sani/htmldoc/ps-pdf.cxx:7081:5
#2 0x558013 in parse_doc(tree_str*, float*, float*, float*, float*, float*, int*, tree_str*, int*) /home/htmldoc_sani/htmldoc/ps-pdf.cxx:4167:11
#3 0x556c54 in parse_doc(tree_str*, float*, float*, float*, float*, float*, int*, tree_str*, int*) /home/htmldoc_sani/htmldoc/ps-pdf.cxx:4081:9
#4 0x556c54 in parse_doc(tree_str*, float*, float*, float*, float*, float*, int*, tree_str*, int*) /home/htmldoc_sani/htmldoc/ps-pdf.cxx:4081:9
#5 0x54f90e in pspdf_export /home/htmldoc_sani/htmldoc/ps-pdf.cxx:803:3
#6 0x53c845 in main /home/htmldoc_sani/htmldoc/htmldoc.cxx:1291:3
#7 0x7f52a6b3e0b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/csu/../csu/libc-start.c:308:16
#8 0x41f8bd in _start (/home/htmldoc_sani/htmldoc/htmldoc+0x41f8bd)

0x625000002100 is located 32 bytes to the right of 8160-byte region [0x625000000100,0x6250000020e0)
allocated by thread T0 here:
#0 0x4ee4e in realloc /home/goushi/work/libfuzzer-workshop/src/llvm/projects/compiler-rt/lib/asan/asan_malloc_linux.cc:165
#1 0x55d96b in check_pages(int) /home/htmldoc_sani/htmldoc/ps-pdf.cxx:8804:24
```

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/htmldoc\_sani/htmldoc/ps-pdf.cxx:6123:34 in render\_table\_row(hdtable\_t&, tree\_str\*\*\*, int, unsigned char\*, float, float, float, float, float\*, float\*, int\*)

Shadow bytes around the buggy address:

```
0x0c4a7fff83d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c4a7fff83e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c4a7fff83f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c4a7fff8400: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c4a7fff8410: 00 00 00 00 00 00 00 00 00 00 00 00 00 fa fa fa
=>0x0c4a7fff8420:[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c4a7fff8430: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c4a7fff8440: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c4a7fff8450: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c4a7fff8460: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c4a7fff8470: fa 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
==38248==ABORTING
```


```
—— source:ps-pdf.cxx:8754 ——
8749     break;
8750 }
8751
8752 if (insert)
8753 {
+ 8754     if (insert->prev)
8755         insert->prev->next = r;
8756     else
8757         pages[page].start = r;
8758
8759     r->prev = insert->prev;
- threads ——
[#0] Id 1, Name: "htmldoc", stopped 0x415e8c in new_render (), reason: SIGSEGV
- trace ——
```

```
[#0] 0x415e8c → new_render(page=0x14, type=0x2, x=0, y=-6.3660128850113321e+24, width=1.2732025770022664e+25, height=6.3660128850113321e+24, data=0x7ffffff6a40,
insert=0x682881c800000000)
[#1] 0x4267e2 → render_table_row(table=@0x7ffffff6d98, cells=<optimized out>, row=<optimized out>, height_var=<optimized out>, left=0, right=0, bottom=<optimized out>, top=
<optimized out>, x=<optimized out>, y=<optimized out>, page=<optimized out>)
[#2] 0x424519 → parse_table(t=<optimized out>, left=<optimized out>, right=<optimized out>, bottom=<optimized out>, top=<optimized out>, x=<optimized out>, y=<optimized out>, page=
<optimized out>, needspace=<optimized out>)
[#3] 0x4157c0 → parse_doc(t=0x918c20, left=0x7fffffffb6e8, right=0x7fffffffb6e4, bottom=0x7fffffffb6ac, top=<optimized out>, x=<optimized out>, y=0x7fffffffb674,
page=0x7fffffffb684, cpara=0x917cc0, needspace=0x7fffffffb6d4)
[#4] 0x414964 → parse_doc(t=0x918390, left=<optimized out>, right=<optimized out>, bottom=<optimized out>, top=0x7fffffffb69c, x=0x7fffffffb6ec, y=<optimized out>, page=<optimized
out>, cpara=<optimized out>, needspace=<optimized out>)
[#5] 0x414964 → parse_doc(t=0x9171d0, left=<optimized out>, right=<optimized out>, bottom=<optimized out>, top=0x7fffffffb69c, x=0x7fffffffb6ec, y=<optimized out>, page=<optimized
out>, cpara=<optimized out>, needspace=<optimized out>)
[#6] 0x411980 → pspdf_export(document=<optimized out>, toc=<optimized out>)
[#7] 0x408e89 → main(argc=<optimized out>, argv=<optimized out>)
—
```

reporter: chiba of topsec alphalab

 **michaelsweet** self-assigned this on Jan 26, 2021


 **michaelsweet** added **bug** **priority-high** labels on Jan 26, 2021


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

**michaelsweet** commented on Jan 26, 2021

Owner

Confirmed, although the backtrace I get is a little different on macOS... Investigating...

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

 Fix a crash bug with bogus table attributes (Issue [#417](#))


✖ [0ddab26](#)

**michaelsweet** commented on Apr 1, 2021

Owner

[master [0ddab26](#)] Fix a crash bug with bogus table attributes (Issue [#417](#))

The issue was the bogus border value, which was larger than a page. Added some range checking to limit to sane values.

 **michaelsweet** closed this as completed on Apr 1, 2021

**chibataiki** commented on Feb 21

Author

[CVE-2021-26259](#) assigned

#### Assignees

 **michaelsweet**

#### Labels

**bug** **priority-high**

#### Projects

None yet

#### Milestone

Stable

#### Development

No branches or pull requests

2 participants

