

New issue

[Jump to bottom](#)

AddressSanitizer: stack-buffer-overflow in parse_table ps-pdf.cxx:6611:25 #416

🔒 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 stack-buffer-overflow in parse_table() ps-pdf.cxx:6611:25

- test platform
html doc 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:

```
html doc -f demo.pdf poc6.html
```

poc(zippped for update):
[poc6.zip](#)

```
=====
==38215==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7fff47945520 at pc 0x000000589ba7 bp 0x7fff47941170 sp 0x7fff47941168
READ of size 4 at 0x7fff47945520 thread T0
#0 0x589ba6 in parse_table(tree_str*, float, float, float, float, float*, float*, int*, int) /home//html doc_sani/html doc/ps-pdf.cxx:6611:25
#1 0x558013 in parse_doc(tree_str*, float*, float*, float*, float*, float*, int*, tree_str*, int*) /home//html doc_sani/html doc/ps-pdf.cxx:4167:11
#2 0x556c54 in parse_doc(tree_str*, float*, float*, float*, float*, float*, int*, tree_str*, int*) /home//html doc_sani/html doc/ps-pdf.cxx:4081:9
#3 0x556c54 in parse_doc(tree_str*, float*, float*, float*, float*, float*, int*, tree_str*, int*) /home//html doc_sani/html doc/ps-pdf.cxx:4081:9
#4 0x54f90e in pspdf_export /home//html doc_sani/html doc/ps-pdf.cxx:803:3
#5 0x53c845 in main /home//html doc_sani/html doc/html doc.cxx:1291:3
#6 0x7ffbd5cfc0b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/csu/../csu/libc-start.c:308:16
#7 0x41f8bd in _start (/home//html doc_sani/html doc/html doc+0x41f8bd)



Address 0x7fff47945520 is located in stack of thread T0 at offset 17312 in frame
#0 0x585052 in parse_table(tree_str*, float, float, float, float, float*, float*, int*, int) /home//html doc_sani/html doc/ps-pdf.cxx:6308

This frame has 12 object(s):
[32, 36) 'left.addr'
[48, 52) 'right.addr'
[64, 17312) 'table' (line 6317) <== Memory access at offset 17312 overflows this variable
[17568, 17572) 'col_min' (line 6318)
[17584, 17588) 'col_pref' (line 6318)
[17600, 17604) 'col_height' (line 6318)
[17616, 17620) 'temp_bottom' (line 6318)
[17632, 17636) 'temp_top' (line 6318)
[17648, 17652) 'temp_page' (line 6335)
[17664, 17676) 'bgrgb' (line 6346)
[17696, 17951) 'table_text' (line 6980)
[18016, 18020) 'temp_y' (line 7092)
HINT: this may be a false positive if your program uses some custom stack unwind mechanism, swapcontext or vfork
(longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow /home//html doc_sani/html doc/ps-pdf.cxx:6611:25 in parse_table(tree_str*, float, float, float, float, float*, float*, int*, int)
Shadow bytes around the buggy address:
 0x100068f20a50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x100068f20a60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x100068f20a70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x100068f20a80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x100068f20a90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x100068f20aa0: 00 00 00 00[f2]f2 f2 f2 f2 f2 f2 f2 f2 f2 f2
0x100068f20ab0: f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2
0x100068f20ac0: f2 f2 f2 f2 04 f2 04 f2 04 f2 04 f2 04 f2 04 f2
0x100068f20ad0: 00 04 f2 f2 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8
0x100068f20ae0: f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8
0x100068f20af0: f8 f8 f8 f8 f2 f2 f2 f2 f2 f2 f2 f2 f8 f3 f3
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
==38215==ABORTING
```

```
- source:ps-pdf.cxx+6611 -----
6606         table.col_smins[col] = col_min;
6607
6608         temp_width = col_width / colspan;
6609         for (int i = 0; i < colspan; i++)
6610         {
6611             // i=0x1d0a, col=0x1
6612             if (temp_width > table.col_widths[col + i])
6613                 table.col_widths[col + i] = temp_width;
6614         }
6615     }
6616 }
- threads -----
[#0] Id 1, Name: "html doc", stopped 0x4238fa in parse_table (), reason: SIGSEGV
- trace -----
[#0] 0x4238fa → parse_table(t=0x918b60, left=0, right=487, bottom=22, top=698, x=<optimized out>, y=0x7fffffffb674, page=<optimized out>, needspace=0x1)
[#1] 0x4157c0 → parse_doc(t=0x918b60, left=0x7fffffffb6e8, right=0x7fffffffb6e4, bottom=0x7fffffffb6ac, top=<optimized out>, x=<optimized out>, y=0x7fffffffb674,
page=0x7fffffffb684, cpara=0x917d10, needspace=0x7fffffffb6d4)
[#2] 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>)
[#3] 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>)
[#4] 0x411980 → pspdf_export(document=<optimized out>, toc=<optimized out>)
[#5] 0x408e89 → main(argc=<optimized out>, argv=<optimized out>)
```

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

 Fix crash bugs with bogus table attributes (Issue [#416](#))


✖ ba61a3e

michaelrsweet commented on Apr 1, 2021

Owner

[master [ba61a3e](#)] Fix crash bugs with bogus table attributes (Issue [#416](#))

More range checking for colspan and rowspan.

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

chibataiki commented on Feb 21

Author

[CVE-2021-23206](#) assigned

Assignees

 **michaelrsweet**

Labels

bug **priority-high**

Projects

None yet

Milestone

Stable

Development

No branches or pull requests

2 participants

