New issue

Jump to bottom

# Fix the stack buffer overflow issue #184

Merged dov merged 1 commit into `fribidi:master` from `tagoh:issues/181` on Feb 17



**tagoh** commented on Feb 17 Contributor

strlen() could returns 0. Without a conditional check for len,
accessing S_ pointer with len - 1 may causes a stack buffer overflow.

AddressSanitizer reports this like:
==1219243==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7ffdce043c1f at pc
0x000000403547 bp 0x7ffdce0
43b30 sp 0x7ffdce043b28
READ of size 1 at 0x7ffdce043c1f thread T0
#0 0x403546 in main ../bin/fribidi-main.c:393
#1 0x7f226804e58f in __libc_start_call_main (/lib64/libc.so.6+0x2d58f)
#2 0x7f226804e648 in __libc_start_main_impl (/lib64/libc.so.6+0x2d648)
#3 0x4036f4 in _start (/tmp/fribidi/build/bin/fribidi+0x4036f4)

Address 0x7ffdce043c1f is located in stack of thread T0 at offset 63 in frame
#0 0x4022bf in main ../bin/fribidi-main.c:193

This frame has 5 object(s):
[32, 36) 'option_index' (line 233)
[48, 52) 'base' (line 386)
[64, 65064) 'S_' (line 375) <== Memory access at offset 63 underflows this variable
[65328, 130328) 'outstring' (line 385)
[130592, 390592) 'logical' (line 384)

This fixes #181

`Fix the stack buffer overflow issue` ... `ad3a19e`

**dov** merged commit **`cffa304`** into `fribidi:master` on Feb 17

Reviewers

No reviews

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

**stack-buffer-overflow on address 0x7ffda2c0112f at pc 0x5580929d7ab5 bp 0x7ffda2bc1820 sp 0x7ffda2bc1810**

2 participants