Optical Character Recognition (GOCR) Bugs

Status: Alpha
Brought to you by: joerg10

#41 A use-after-free in pgm2asc.c:2817:39

Status: open **Owner:** nobody **Labels:** bug (3)
Priority: 5
Updated: 2020-08-03 **Created:** 2020-08-03 **Creator:** zhouan **Private:** No

System info

Ubuntu X64, gcc (Ubuntu 5.5.0-12ubuntu1), gocr (latest jocr-dev 0.53-20200802)

Configure

CFLAGS="-g -fsanitize=address" LDFLAGS="-fsanitize=address" ./configure

Command line

./src/gocr -m 4 ./use-after-free-context_correction-pgm2asc-2817

AddressSanitizer output

```
=================================================================
==33563==ERROR: AddressSanitizer: heap-use-after-free on address 0x61a000001ea4 at pc 0x000
READ of size 4 at 0x61a000001ea4 thread T0
    #0 0x54552a in context_correction /home/seviezhou/AlphaFuzz/targets/jocr/src/pgm2asc.c:
    #1 0x54a291 in pgm2asc /home/seviezhou/AlphaFuzz/targets/jocr/src/pgm2asc.c:3427:28
    #2 0x518776 in main /home/seviezhou/AlphaFuzz/targets/jocr/src/gocr.c:350:5
    #3 0x7f394175f83f in __libc_start_main /build/glibc-e6zv40/glibc-2.23/csu/../csu/libc-s
    #4 0x41a768 in _start (/home/seviezhou/AlphaFuzz/targets/gocr/src/gocr+0x41a768)

0x61a000001ea4 is located 36 bytes inside of 1376-byte region [0x61a000001e80,0x61a0000023e
freed by thread T0 here:
    #0 0x4de7b8 in __interceptor_cfree.localalias.0 /home/seviezhou/llvm-6.0.0/projects/com
    #1 0x54c1a7 in free_box /home/seviezhou/AlphaFuzz/targets/jocr/src/box.c:122:3

previously allocated by thread T0 here:
    #0 0x4de978 in __interceptor_malloc /home/seviezhou/llvm-6.0.0/projects/compiler-rt/lib
    #1 0x54bdc2 in malloc_box /home/seviezhou/AlphaFuzz/targets/jocr/src/box.c:96:24

SUMMARY: AddressSanitizer: heap-use-after-free /home/seviezhou/AlphaFuzz/targets/jocr/src/p
Shadow bytes around the buggy address:
  0x0c347fff8380: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c347fff8390: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c347fff83a0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c347fff83b0: fd fd fd fd fd fd fd fd fd fd fd fd fa fa fa fa
  0x0c347fff83c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c347fff83d0: fd fd fd fd[fd]fd fd fd fd fd fd fd fd fd fd fd
  0x0c347fff83e0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c347fff83f0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c347fff8400: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c347fff8410: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c347fff8420: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
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
==33563==ABORTING
```

1 Attachments

use-after-free-context_correction-pgm2asc-2817.zip

Discussion

Log in to post a comment.