libconfuse / **libconfuse** Public

Code · Issues 12 · Pull requests 3 · Actions · Projects · Wiki · ···

New issue · Jump to bottom

[BUG] heap buffer overflow in cfg_tilde_expand #163

Closed **kdsjZh** opened this issue on Sep 2 · 0 comments

kdsjZh commented on Sep 2 • edited

short summary

Hello, I was testing my fuzzer and found a heap buffer overflow in cfg_tilde_expand, src/confuse.c:1909. A heap buffer overflow can be triggered when parsing a crafted file. As shown in the attachment

Step to reproduce

```
CC="gcc -fsanitize=address -g " CXX="g++ -fsanitize=address -g" ./autogen.sh && ./configure --
disable-shared && make -j$(nproc)
./examples/cfgtest $POC
```

Environment

- Ubuntu 22.04 (docker image)
- gcc 11.2.0
- libconfuse latest commit `0325922`

ASan Log

```
Using libConfuse 3.3 by Martin Hedenfalk <martin@bzero.se>

=================================================================
==418268==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000572 at pc
0x7f94939440ab bp 0x7ffcd3b274a0 sp 0x7ffcd3b26c48
READ of size 3 at 0x602000000572 thread T0
    #0 0x7f94939440aa in __interceptor_getpwnam
../../../../src/libsanitizer/sanitizer_common/sanitizer_common_interceptors.inc:1922
    #1 0x555a8791cb33 in cfg_tilde_expand /benchmark/libconfuse/src/confuse.c:1909
    #2 0x555a8792a97f in cfg_lexer_include /benchmark/libconfuse/src/lexer.l:332
    #3 0x555a8791d6e0 in call_function /benchmark/libconfuse/src/confuse.c:1274
    #4 0x555a87921a3e in cfg_parse_internal /benchmark/libconfuse/src/confuse.c:1566
    #5 0x555a87922523 in cfg_parse_fp /benchmark/libconfuse/src/confuse.c:1701
    #6 0x555a87922523 in cfg_parse_fp /benchmark/libconfuse/src/confuse.c:1685
    #7 0x555a8792266f in cfg_parse /benchmark/libconfuse/src/confuse.c:1802
    #8 0x555a8791750a in main /benchmark/libconfuse/examples/cfgtest.c:129
    #9 0x7f94936bad8f in __libc_start_call_main ../sysdeps/nptl/libc_start_call_main.h:58
    #10 0x7f94936bae3f in __libc_start_main_impl ../csu/libc-start.c:392
    #11 0x555a87917d04 in _start (/benchmark/libconfuse/examples/cfgtest+0x7d04)

0x602000000572 is located 0 bytes to the right of 2-byte region [0x602000000570,0x602000000572)
allocated by thread T0 here:
    #0 0x7f949396d867 in __interceptor_malloc
../../../../src/libsanitizer/asan/asan_malloc_linux.cpp:145
    #1 0x555a8791cb0d in cfg_tilde_expand /benchmark/libconfuse/src/confuse.c:1904

SUMMARY: AddressSanitizer: heap-buffer-overflow
../../../../src/libsanitizer/sanitizer_common/sanitizer_common_interceptors.inc:1922 in
__interceptor_getpwnam
Shadow bytes around the buggy address:
  0x0c047fff8050: fa fa 00 fa fa fa 00 fa fa fa 00 fa fa fa 00 fa
  0x0c047fff8060: fa fa 00 fa fa fa 05 fa fa fa fd fa fa fa fd fa
  0x0c047fff8070: fa fa 00 fa fa fa fd fd fa fa 00 fa fa fa 00 fa
  0x0c047fff8080: fa fa 00 fa fa fa 00 fa fa fa fd fa fa fa 00 fa
  0x0c047fff8090: fa fa fd fd fa fa 00 fa fa fa 00 fa fa fa 00 fa
=>0x0c047fff80a0: fa fa 00 fa fa fa 03 fa fa fa 00 fa fa fa[02]fa
  0x0c047fff80b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff80c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff80d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff80e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff80f0: 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
==418268==ABORTING
```

Credit

Han Zheng (NCNIPC of China, Hexhive)

POC

poc0.zip

👀 1

troglobit closed this as completed in `d73777c` on Sep 2

troglobit added a commit that referenced this issue on Sep 2

`Credit Han Zheng for finding issue #163` … `b3841f0`

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant