New issue

Jump to bottom

SEGV (stack overflow) on XRef::fetch #25

Open strongcourage opened this issue on May 27, 2019 · 0 comments

strongcourage commented on May 27, 2019 • edited

Hi,

Our fuzzer found a crash due to a stack overflow bug on the function XRef::fetch (the latest commit `b671b64` on master - version 0.70).

PoC_so_XRef::fetch: https://github.com/strongcourage/PoCs/blob/master/pdf2json_b671b64/PoC_so_XRef::fetch

Valgrind says:

```
valgrind pdf2json PoC_so_XRef\:\:fetch /dev/null
==17786== Memcheck, a memory error detector
==17786== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==17786== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==17786== Command: ./pdf2json PoC_so_XRef::fetch /dev/null
==17786==
==17786== Stack overflow in thread #1: can't grow stack to 0xffe801000
==17786==
==17786== Process terminating with default action of signal 11 (SIGSEGV)
==17786==  Access not within mapped region at address 0xFFE801FF8
==17786== Stack overflow in thread #1: can't grow stack to 0xffe801000
==17786==    at 0x4090A2: Object::Object() (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43F7AB: XRef::fetch(int, int, Object*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43CE5B: ObjectStream::ObjectStream(XRef*, int) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43FB1F: XRef::fetch(int, int, Object*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43CE5B: ObjectStream::ObjectStream(XRef*, int) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43FB1F: XRef::fetch(int, int, Object*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43CE5B: ObjectStream::ObjectStream(XRef*, int) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43FB1F: XRef::fetch(int, int, Object*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43CE5B: ObjectStream::ObjectStream(XRef*, int) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43FB1F: XRef::fetch(int, int, Object*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43CE5B: ObjectStream::ObjectStream(XRef*, int) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==    by 0x43FB1F: XRef::fetch(int, int, Object*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==17786==  If you believe this happened as a result of a stack
==17786==  overflow in your program's main thread (unlikely but
==17786==  possible), you can try to increase the size of the
==17786==  main thread stack using the --main-stacksize= flag.
==17786==  The main thread stack size used in this run was 8388608.
==17786== Stack overflow in thread #1: can't grow stack to 0xffe801000
==17786==
==17786== Process terminating with default action of signal 11 (SIGSEGV)
==17786==  Access not within mapped region at address 0xFFE801FF8
==17786== Stack overflow in thread #1: can't grow stack to 0xffe801000
==17786==    at 0x4A28680: _vgnU_freeres (in /usr/lib/valgrind/vgpreload_core-amd64-linux.so)
==17786==  If you believe this happened as a result of a stack
==17786==  overflow in your program's main thread (unlikely but
==17786==  possible), you can try to increase the size of the
==17786==  main thread stack using the --main-stacksize= flag.
==17786==  The main thread stack size used in this run was 8388608.
==17786==
==17786== HEAP SUMMARY:
==17786==     in use at exit: 836,551 bytes in 27,879 blocks
==17786==   total heap usage: 27,938 allocs, 59 frees, 946,615 bytes allocated
==17786==
==17786== LEAK SUMMARY:
==17786==    definitely lost: 0 bytes in 0 blocks
==17786==    indirectly lost: 0 bytes in 0 blocks
==17786==      possibly lost: 0 bytes in 0 blocks
==17786==    still reachable: 836,551 bytes in 27,879 blocks
==17786==         suppressed: 0 bytes in 0 blocks
==17786== Rerun with --leak-check=full to see details of leaked memory
==17786==
==17786== For counts of detected and suppressed errors, rerun with: -v
==17786== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
Segmentation fault
```

Thanks,
Manh Dung

strongcourage changed the title ~~Segmentation fault (stack overflow) on XRef::fetch~~ **SEGV (stack overflow) on XRef::fetch** on May 29, 2019

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant