SEGV (NULL pointer dereference) on DCTStream::getChar #31

Open **strongcourage** opened this issue on May 28, 2019 · 0 comments

strongcourage commented on May 28, 2019

Hi,

Our fuzzer found a crash due to a NULL pointer dereference bug on the function DCTStream::getChar (the latest commit `b671b64` on master - version 0.70).

PoC: https://github.com/strongcourage/PoCs/blob/master/pdf2json_b671b64/PoC_npd_DCTStream::getChar

```
valgrind pdf2json $PoC /dev/null
==20313== Memcheck, a memory error detector
==20313== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==20313== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==20313== Command: ./pdf2json PoC_npd_DCTStream::getChar /dev/null
==20313==
Error (13143): Badly formatted number
Error: PDF file is damaged - attempting to reconstruct xref table...
Error (13759): Illegal character '>'
Error (7369): Dictionary key must be a name object
Error (7379): Dictionary key must be a name object
Error (7380): Illegal character '>'
Error (7380): Dictionary key must be a name object
Error (7388): Dictionary key must be a name object
Error (7394): Dictionary key must be a name object
Error (7853): Dictionary key must be a name object
Error (7903): Dictionary key must be a name object
Error (7913): Dictionary key must be a name object
Error (7920): Dictionary key must be a name object
Error (7922): Dictionary key must be a name object
Error (7924): Dictionary key must be a name object
Error (7928): Dictionary key must be a name object
Error (7933): Dictionary key must be a name object
Error (7940): Dictionary key must be a name object
Error (7943): Dictionary key must be a name object
Error (7945): Dictionary key must be a name object
Error (7949): Dictionary key must be a name object
Error (7960): Dictionary key must be a name object
Error (7972): Dictionary key must be a name object
Error (7976): Dictionary key must be a name object
Error (7979): Dictionary key must be a name object
Error (7987): Dictionary key must be a name object
==20313== Invalid read of size 1
==20313==    at 0x433044: DCTStream::getChar() (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x40947A: Object::streamGetChar() (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x48796F: Lexer::getChar() (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x487A83: Lexer::getObj(Object*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x4890DF: Parser::Parser(XRef*, Lexer*, int) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x4542F8: Gfx::display(Object*, int) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x42A59E: Page::displaySlice(OutputDev*, double, double, int, int, int, int, int, int, int, Links*, int, Catalog*, int (*)(void*), void*) (in
/home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x42A14A: Page::display(OutputDev*, double, double, int, int, int, Links*, int, Catalog*, int (*)(void*), void*) (in
/home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x42BCBD: PDFDoc::displayPage(OutputDev*, int, double, double, int, int, int, int, int (*)(void*), void*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x42BD48: PDFDoc::displayPages(OutputDev*, int, int, double, double, int, int, int, int, int (*)(void*), void*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x40269A: main (pdf2json.cc:275)
==20313==  Address 0x0 is not stack'd, malloc'd or (recently) free'd
==20313==
==20313==
==20313== Process terminating with default action of signal 11 (SIGSEGV)
==20313==  Access not within mapped region at address 0x0
==20313==    at 0x433044: DCTStream::getChar() (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x40947A: Object::streamGetChar() (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x48796F: Lexer::getChar() (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x487A83: Lexer::getObj(Object*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x4890DF: Parser::Parser(XRef*, Lexer*, int) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x4542F8: Gfx::display(Object*, int) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x42A59E: Page::displaySlice(OutputDev*, double, double, int, int, int, int, int, int, int, Links*, int, Catalog*, int (*)(void*), void*) (in
/home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x42A14A: Page::display(OutputDev*, double, double, int, int, int, Links*, int, Catalog*, int (*)(void*), void*) (in
/home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x42BCBD: PDFDoc::displayPage(OutputDev*, int, double, double, int, int, int, int, int (*)(void*), void*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x42BD48: PDFDoc::displayPages(OutputDev*, int, int, double, double, int, int, int, int, int (*)(void*), void*) (in /home/dungnguyen/PoCs/pdf2json_b671b64/pdf2json)
==20313==    by 0x40269A: main (pdf2json.cc:275)
==20313==  If you believe this happened as a result of a stack
==20313==  overflow in your program's main thread (unlikely but
==20313==  possible), you can try to increase the size of the
==20313==  main thread stack using the --main-stacksize= flag.
==20313==  The main thread stack size used in this run was 8388608.
==20313==
==20313== HEAP SUMMARY:
==20313==     in use at exit: 225,178 bytes in 1,772 blocks
==20313==   total heap usage: 2,047 allocs, 275 frees, 341,335 bytes allocated
==20313==
==20313== LEAK SUMMARY:
==20313==    definitely lost: 16 bytes in 1 blocks
==20313==    indirectly lost: 8 bytes in 1 blocks
==20313==      possibly lost: 0 bytes in 0 blocks
==20313==    still reachable: 225,154 bytes in 1,770 blocks
==20313==         suppressed: 0 bytes in 0 blocks
==20313== Rerun with --leak-check=full to see details of leaked memory
==20313==
==20313== For counts of detected and suppressed errors, rerun with: -v
==20313== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
Segmentation fault
```

Thanks,
Manh Dung

strongcourage changed the title ~~Segmentation fault (NULL pointer dereference) on DCTStream::getChar~~ **SEGV (NULL pointer dereference) on DCTStream::getChar** 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