New issue

Jump to bottom

heap-buffer-overflow_in_readHuffSym #3

Open **Cvjark** opened this issue on Aug 7 · 2 comments

Cvjark commented on Aug 7 • edited

Hi, in the lastest version of this code [ps: commit id ffaf11c] I found something unusual.

crash sample

8id0_heap-buffer-overflow_in_readHuffSym.zip

command to reproduce

```
./pdftops -q [crash sample] /dev/null
```

crash detail

```
=================================================================
==108391==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x620000001782 at pc
0x000000759029 bp 0x7ffd51edc550 sp 0x7ffd51edc548
READ of size 2 at 0x620000001782 thread T0
    #0 0x759028 in DCTStream::readHuffSym(DCTHuffTable*)
/home/bupt/Desktop/xpdf/xpdf/Stream.cc:3119:16
    #1 0x7548ba in DCTStream::readDataUnit(DCTHuffTable*, DCTHuffTable*, int*, int*)
/home/bupt/Desktop/xpdf/xpdf/Stream.cc:2624:17
    #2 0x751b27 in DCTStream::readMCURow() /home/bupt/Desktop/xpdf/xpdf/Stream.cc:2392:9
    #3 0x750d6e in DCTStream::getChar() /home/bupt/Desktop/xpdf/xpdf/Stream.cc:2316:12
    #4 0x6899e3 in Object::streamGetChar() /home/bupt/Desktop/xpdf/xpdf/./Object.h:288:20
    #5 0x6899e3 in Lexer::getChar() /home/bupt/Desktop/xpdf/xpdf/Lexer.cc:92:42
    #6 0x6899e3 in Lexer::getObj(Object*) /home/bupt/Desktop/xpdf/xpdf/Lexer.cc:124:14
    #7 0x6a8fc5 in Parser::Parser(XRef*, Lexer*, int) /home/bupt/Desktop/xpdf/xpdf/Parser.cc:33:10
    #8 0x581742 in Gfx::display(Object*, int) /home/bupt/Desktop/xpdf/xpdf/Gfx.cc:641:16
    #9 0x6a76a1 in Page::displaySlice(OutputDev*, double, double, int, int, int, int, int, int,
int, int, int (*)(void*), void*) /home/bupt/Desktop/xpdf/xpdf/Page.cc:360:10
    #10 0x6d5f6e in PSOutputDev::checkPageSlice(Page*, double, double, int, int, int, int, int,
int, int, int, int (*)(void*), void*) /home/bupt/Desktop/xpdf/xpdf/PSOutputDev.cc:3276:11
    #11 0x6a7172 in Page::displaySlice(OutputDev*, double, double, int, int, int, int, int, int,
int, int, int (*)(void*), void*) /home/bupt/Desktop/xpdf/xpdf/Page.cc:328:13
    #12 0x6a6f81 in Page::display(OutputDev*, double, double, int, int, int, int, int (*)(void*),
```

```
void*) /home/bupt/Desktop/xpdf/xpdf/Page.cc:308:3
    #13 0x6af9b4 in PDFDoc::displayPage(OutputDev*, int, double, double, int, int, int, int, int
(*)(void*), void*) /home/bupt/Desktop/xpdf/xpdf/PDFDoc.cc:384:27
    #14 0x6af9b4 in PDFDoc::displayPages(OutputDev*, int, int, double, double, int, int, int, int,
int (*)(void*), void*) /home/bupt/Desktop/xpdf/xpdf/PDFDoc.cc:397:5
    #15 0x796d81 in main /home/bupt/Desktop/xpdf/xpdf/pdftops.cc:342:10
    #16 0x7f3b180d3c86 in __libc_start_main /build/glibc-CVJwZb/glibc-2.27/csu/../csu/libc-
start.c:310
    #17 0x41d5d9 in _start (/home/bupt/Desktop/xpdf/xpdf/pdftops+0x41d5d9)

0x620000001782 is located 2314 bytes to the right of 3576-byte region
[0x620000000080,0x620000000e78)
allocated by thread T0 here:
    #0 0x4f5768 in operator new(unsigned long) /home/bupt/Desktop/tools/llvm-
12.0.1/llvm/projects/compiler-rt/lib/asan/asan_new_delete.cpp:99
    #1 0x7259bc in Stream::makeFilter(char*, Stream*, Object*, int)
/home/bupt/Desktop/xpdf/xpdf/Stream.cc:269:11
    #2 0x72459a in Stream::addFilters(Object*, int) /home/bupt/Desktop/xpdf/xpdf/Stream.cc:141:11
    #3 0x6ad41e in Parser::makeStream(Object*, unsigned char*, CryptAlgorithm, int, int, int, int)
/home/bupt/Desktop/xpdf/xpdf/Parser.cc:214:14
    #4 0x6ab6f6 in Parser::getObj(Object*, int, unsigned char*, CryptAlgorithm, int, int, int,
int) /home/bupt/Desktop/xpdf/xpdf/Parser.cc:101:18
    #5 0x781a3a in XRef::fetch(int, int, Object*, int)
/home/bupt/Desktop/xpdf/xpdf/XRef.cc:1028:13
    #6 0x6a7611 in Page::displaySlice(OutputDev*, double, double, int, int, int, int, int, int,
int, int, int (*)(void*), void*) /home/bupt/Desktop/xpdf/xpdf/Page.cc:357:12
    #7 0x6d5f6e in PSOutputDev::checkPageSlice(Page*, double, double, int, int, int, int, int,
int, int, int, int (*)(void*), void*) /home/bupt/Desktop/xpdf/xpdf/PSOutputDev.cc:3276:11

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/bupt/Desktop/xpdf/xpdf/Stream.cc:3119:16 in
DCTStream::readHuffSym(DCTHuffTable*)
Shadow bytes around the buggy address:
  0x0c407fff82a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c407fff82b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c407fff82c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c407fff82d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c407fff82e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c407fff82f0:[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c407fff8300: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c407fff8310: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c407fff8320: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c407fff8330: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c407fff8340: 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
==108391==ABORTING
```

ajakk commented on Aug 17

This doesn't appear to be the upstream repo for xpdf. Why did you fuzz it? This repo's last commit was in 2014 at xpdf-3.04, while xpdf has received many updates since then, and is currently at 4.04. Can you reproduce these issues on upstream's xpdf?

👍 1

Marceolv commented on Oct 26

Utilizando os conceitos de vetor , implemente um algoritmo que
1 - inclua até 1000 usuários;
2- edite um usuário;
3- exclua um usuário;
4- busque um usuário por e-mail;
5- imprima todos os usuários cadastrados;
6- faça backup dos usuários cadastrados;
7- faça restauração dos dados.

Dados do usuário
Id (Int)- preenchido automaticamente por números radomicos ;
Nome completo (string) ;
E-mail (string) - tem que aparecer o @;
Sexo (string);
Endereço (string);
Altura(double)- aceitar valores de 1 e 2 m;
Vacina (ex: tomou Vacina)

⬆️ Using vector concepts, implement an algorithm that:
1- include up to 1000 users;
2- edit a user;
3- Delete a user;
4- search for a user by email;
5- Print all registered users;
6- back up registered users;
7- Restore the data.

User data
Id (Int)- automatically filled in by radomic numbers;
Full name (string) ;
E-mail (string) - the @ has to appear;
Sex (string);
Address (string);
Height (double)- accept values of 1 and 2 m;
Vaccine (e.g. did you get a vaccine?).

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants