

[BUG] use-after-free in pdfimages,xpdf-4.04

[Post Reply](#) 

Search this topic...




3 posts • Page 1 of 1

hzheng



[BUG] use-after-free in pdfimages,xpdf-4.04

 Fri Aug 12, 2022 8:07 am

Hello, I was testing my fuzzer and found a use-after-free in xpdf-4.04, can be triggered via a crafted pdf file.

Environment

Ubuntu 22.04 (docker)

gcc-11.2.0

xpdf-4.04

step to reproduce

...

```
cd $PATH_TO_XPDF && mkdir build && pushd build
```

```
cmake -DCMAKE_BUILD_TYPE=Release -DCMAKE_C_FLAGS="-fsanitize=address -g" -DCMAKE_CXX_FLAGS="-fsanitize=address -g" ../
```

```
./xpdf/pdfimages $POC /dev/null
```

...

ASan Log

...

```
=====
==1164636==ERROR: AddressSanitizer: heap-use-after-free on address 0x603000012970 at pc 0x55c1c290fa1a bp
0x7ffc26bd4430 sp 0x7ffc26bd4420
```

```
READ of size 8 at 0x603000012970 thread T0
```

```
#0 0x55c1c290fa19 in JBIG2Stream::close() /validate/xpdf/xpdf-4.04/xpdf/JBIG2Stream.cc:1216
```

```
#1 0x55c1c290fa84 in JBIG2Stream::~JBIG2Stream() /validate/xpdf/xpdf-4.04/xpdf/JBIG2Stream.cc:1180
```

```
#2 0x55c1c290ffdc in JBIG2Stream::~JBIG2Stream() /validate/xpdf/xpdf-4.04/xpdf/JBIG2Stream.cc:1202
```

```
#3 0x55c1c293642a in Object::free() /validate/xpdf/xpdf-4.04/xpdf/Object.cc:138
```

```
#4 0x55c1c280217d in Gfx::opXObject(Object*, int) /validate/xpdf/xpdf-4.04/xpdf/Gfx.cc:4136
```

```
#5 0x55c1c2868ab3 in Gfx::execOp(Object*, Object*, int) /validate/xpdf/xpdf-4.04/xpdf/Gfx.cc:862
```

```
#6 0x55c1c2868f88 in Gfx::go(int) /validate/xpdf/xpdf-4.04/xpdf/Gfx.cc:747
```

```
#7 0x55c1c28695a6 in Gfx::display(Object*, int) /validate/xpdf/xpdf-4.04/xpdf/Gfx.cc:669
```

```
#8 0x55c1c2943025 in Page::displaySlice(OutputDev*, double, double, int, int, int, int, int, int, int, int (*)(void*), void*) /validate/xpdf/xpdf-4.04/xpdf/Page.cc:422
```

```
#9 0x55c1c2943882 in Page::display(OutputDev*, double, double, int, int, int, int, int (*)(void*), void*) /validate/xpdf/xpdf-4.04/xpdf/Page.cc:368
```

```
#10 0x55c1c294d4a3 in PDFDoc::displayPages(OutputDev*, int, int, double, double, int, int, int, int, int (*)(void*), void*) /validate/xpdf/xpdf-4.04/xpdf/PDFDoc.cc:460
```

```
#11 0x55c1c280656d in main /validate/xpdf/xpdf-4.04/xpdf/pdfimages.cc:156
```

```
#12 0x7efd8b7e7d8f in __libc_start_call_main ../sysdeps/nptl/libc_start_call_main.h:58
```

```
#13 0x7efd8b7e7e3f in __libc_start_main_impl ../csu/libc-start.c:392
```

```
#14 0x55c1c28070b4 in _start (/validate/xpdf/xpdf-4.04/build/xpdf/pdfimages+0x1280b4)
```

0x603000012970 is located 0 bytes inside of 32-byte region [0x603000012970,0x603000012990)

freed by thread T0 here:

```
#0 0x7efd8bdd022f in operator delete(void*, unsigned long) ../../../../src/libsanitizer/asan/asan_new_delete.cpp:172
#1 0x55c1c290b19f in JBIG2Bitmap::~JBIG2Bitmap() /validate/xpdf/xpdf-4.04/xpdf/JBIG2Stream.cc:740
#2 0x55c1c290b19f in JBIG2Stream::readPageInfoSeg(unsigned int) /validate/xpdf/xpdf-4.04/xpdf/JBIG2Stream.cc:3960
```

previously allocated by thread T0 here:

```
#0 0x7efd8bdcf1c7 in operator new(unsigned long) ../../../../src/libsanitizer/asan/asan_new_delete.cpp:99
#1 0x55c1c290b1fd in JBIG2Stream::readPageInfoSeg(unsigned int) /validate/xpdf/xpdf-4.04/xpdf/JBIG2Stream.cc:3969
```

SUMMARY: AddressSanitizer: heap-use-after-free /validate/xpdf/xpdf-4.04/xpdf/JBIG2Stream.cc:1216 in JBIG2Stream::~close()

Shadow bytes around the buggy address:

```
0x0c067ffa4d0: fd fd fa fa fd fd fd fa fa fa 00 00 01 fa fa fa
0x0c067ffa4e0: 00 00 00 00 fa fa 00 00 00 00 fa fa fd fd fd
0x0c067ffa4f0: fa fa fd fd fd fd fa fa fd fd fd fa fa 00 00
0x0c067ffa500: 01 fa fa fa 00 00 00 fa fa fa 00 00 00 fa fa fa
0x0c067ffa510: 00 00 00 fa fa fa 00 00 00 fa fa fa 00 00 00 00
=>0x0c067ffa520: fa fa 00 00 00 fa fa fa 00 00 00 fa fa fa[fd]fd
0x0c067ffa530: fd fd fa fa fd fd fd fd fa fa fa fa fa fa fa
0x0c067ffa540: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c067ffa550: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c067ffa560: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c067ffa570: 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

==1164636==ABORTING

...

Credit

Han Zheng, NCNIPC of China, Hexhive

POC

[poc_uaf_pdfimages.zip](#)

(15.28 KiB) Downloaded 388 times

derekn



Re: [BUG] use-after-free in pdfimages,xpdf-4.04

Fri Aug 12, 2022 7:56 pm

That was running into an incomplete sanity check on bitmap size in the JBIG2 decoder. I'll have it fixed in the next release.

Thanks for the bug report.

hzheng



Re: [BUG] use-after-free in pdfimages,xpdf-4.04

Sat Aug 13, 2022 7:56 am

Thanks for your quick validation & confirmation.

Post Reply



3 posts • Page **1** of **1**

[Return to "Xpdf open source"](#)

Jump to

[Board index](#)

[Delete cookies](#) All times are UTC

Powered by [phpBB®](#) Forum Software © phpBB Limited

[Privacy](#) | [Terms](#)