Exiv2 - Bug #1247

out of bounds read access in Exiv2::Image::setIccProfile

21 Oct 2016 01:17 - Hanno Böck

Status:	Closed	**Start date:**	21 Oct 2016
Priority:	Normal	**Due date:**	
Assignee:	Robin Mills	**% Done:**	100%
Category:	image format	**Estimated time:**	7.00 hours
Target version:	0.26		

Description

The attached file will cause an out of bounds read access of one byte, visible with address sanitizer (add "-fsanitize=address" to CFLAGS/CXXFLAGS/LDFLAGS). This bug was found with american fuzzy lop. Tested with the latest svn code.

Here's the asan error message:
6112ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60200000ec33 at pc 0x7f169de57b9b bp 0x7ffd9b30fa40 sp 0x7ffd9b30fa38
READ of size 1 at 0x60200000ec33 thread T0
#0 0x7f169de57b9a in Exiv2::getULong(unsigned char const*, Exiv2::ByteOrder) /f/exiv2/trunk/src/types.cpp:246:28
#1 0x7f169dbd2272 in Exiv2::Image::setIccProfile(Exiv2::DataBuf&, bool) /f/exiv2/trunk/src/image.cpp:617:45
#2 0x7f169dc218bd in Exiv2::JpegBase::readMetadata() /f/exiv2/trunk/src/jpgimage.cpp:487:21
#3 0x54538c in Action::Print::printSummary() /f/exiv2/trunk/src/actions.cpp:289:9
#4 0x544a58 in Action::Print::run(std::string const&) /f/exiv2/trunk/src/actions.cpp:244:44
#5 0x4fe07f in main /f/exiv2/trunk/src/exiv2.cpp:170:19
#6 0x7f169c42a78f in __libc_start_main (/lib64/libc.so.6+0x2078f)
#7 0x421eb8 in _start (/r/exiv2/exiv2+0x421eb8)

0x60200000ec33 is located 2 bytes to the right of 1-byte region [0x60200000ec30,0x60200000ec31)
allocated by thread T0 here:
#0 0x4fab70 in operator new[](unsigned long) (/r/exiv2/exiv2+0x4fab70)
#1 0x7f169dc212e0 in Exiv2::DataBuf::DataBuf(long) /f/exiv2/trunk/include/exiv2/types.hpp:204:46
#2 0x7f169dc212e0 in Exiv2::JpegBase::readMetadata() /f/exiv2/trunk/src/jpgimage.cpp:483
#3 0x54538c in Action::Print::printSummary() /f/exiv2/trunk/src/actions.cpp:289:9
#4 0x544a58 in Action::Print::run(std::string const&) /f/exiv2/trunk/src/actions.cpp:244:44
#5 0x7f169c42a78f in __libc_start_main (/lib64/libc.so.6+0x2078f)

SUMMARY: AddressSanitizer: heap-buffer-overflow /f/exiv2/trunk/src/types.cpp:246:28 in Exiv2::getULong(unsigned char const*, Exiv2::ByteOrder)
Shadow bytes around the buggy address:
0x0c047fff9d30: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c047fff9d40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c047fff9d50: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c047fff9d60: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c047fff9d70: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c047fff9d80: fa fa fa fa fa[01]fa fa 00 00 fa fa 00 00
0x0c047fff9d90: fa fa 00 fa fa fa 00 fa fa fa 00 00 fa fa 00 00
0x0c047fff9da0: fa fa 00 fa fa fa 00 fa fa fa fd fa fa fa fd fa
0x0c047fff9db0: fa fa fd fa fa fa fd fa fa fa fd fa fa fa fd fa
0x0c047fff9dc0: fa fa 06 fa fa fa 00 04 fa fa 00 04 fa fa 00 04
0x0c047fff9dd0: fa fa 00 04 fa fa 00 04 fa fa 00 04 fa fa 00 04
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
Heap right redzone: fb
Freed heap region: fd
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack partial redzone: f4
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
6112ABORTING

Related issues:

Related to Exiv2 - Bug #1248: floating point exception / crash on malformed i...	**Closed**	**21 Oct 2016**

Associated revisions

Revision 4651 - 21 Oct 2016 17:44 - Robin Mills

#1247 Thank You Hanno for reporting this and providing a patch.

Revision 4653 - 21 Oct 2016 19:19 - Robin Mills

#1247 Fix Linux/GCC compilation warning. Added Hanno's file to the test suite.

Revision 4655 - 21 Oct 2016 19:24 - Robin Mills

#1247 Restrict clang pragma to *APPLE*

Revision 4666 - 31 Oct 2016 18:21 - Robin Mills

#1247. Correction to r4655 to handle clang on plaforms other than MacOS-X.

Revision 4667 - 31 Oct 2016 18:42 - Robin Mills

#1247 Another correction to r4655 concerning clang/apple.

History

#1 - 21 Oct 2016 06:06 - Robin Mills

- Category set to not-a-bug

- Status changed from New to Closed

- Assignee set to Robin Mills

- Target version set to 0.26

- % Done changed from 0 to 100

- Estimated time set to 1.00 h

That is not a legal JPEG. It is a deliberately corrupted file and we throw an exception. Not a bug.

```
501 rmills@rmillsmbp:~ $ exiv2 -pS http://dev.exiv2.org/attachments/download/1087/exiv2-oob-heap-Exiv2__Image__setIccProfile.jpg
STRUCTURE OF JPEG FILE: http://dev.exiv2.org/attachments/download/1087/exiv2-oob-heap-Exiv2__Image__setIccProfile.jpg
 address | marker       |  length | data
       0 | 0xffd8 SOI
       2 | 0xffe2 APP2  |      16 | ICC_PROFILE.... chunk 0/0Exiv2 exception in print action for file http://dev.exiv2.org/attachments/download/1087/exiv2-oob-heap-Exiv2__Image__setIccProfile.jpg:
Failed to read image data
502 rmills@rmillsmbp:~ $ exiv2 -pa http://dev.exiv2.org/attachments/download/1087/exiv2-oob-heap-Exiv2__Image__setIccProfile.jpg
Warning: JPEG format error, rc = 5
http://dev.exiv2.org/attachments/download/1087/exiv2-oob-heap-Exiv2__Image__setIccProfile.jpg: (No XMP data found in the file)
503 rmills@rmillsmbp:~ $
```

#2 - 21 Oct 2016 11:40 - Hanno Böck

- File exiv2-fix-oob-setIccProfile.diff added

Even with a corrupted JPEG you shouldn't read beyond the bounds of the allocated memory.

I've looked into the code, I think it's relatively trivial to fix, just have to check before the getULong that you can really read a long from that buffer. See

attached patch.

#3 - 21 Oct 2016 17:45 - Robin Mills

Thanks. I've submitted you patch. r4651

I doesn't crash on the Mac. It throws an exception. However it does crash on Linux. Probably because the Mac uses clang and linux uses gcc. What about msvc/cl? I'll look into that this evening.

To make the code really robust and deal with ever possible file is non trivial. We have a project scheduled for v0.27 (#992) to stress and strengthen that part of the code. It has been written in the (naive) assumption that all files conform to the standard.

#4 - 21 Oct 2016 17:45 - Robin Mills

- Category changed from not-a-bug to image format

#5 - 21 Oct 2016 17:55 - Robin Mills

It seems sane on Windows:

```
C:\Users\rmills\gnu\exiv2\trunk\msvc2005\bin\x64\releasedll>exiv2 -pa http://dev.exiv2.org/attachments/downloa
d/1087/exiv2-oob-h
eap-Exiv2__Image__setIccProfile.jpg
Warning: JPEG format error, rc = 5

C:\Users\rmills\gnu\exiv2\trunk\msvc2005\bin\x64\releasedll>exiv2 -pR http://dev.exiv2.org/attachments/downloa
d/1087/exiv2-oob-h
eap-Exiv2__Image__setIccProfile.jpg
STRUCTURE OF JPEG FILE: http://dev.exiv2.org/attachments/download/1087/exiv2-oob-heap-Exiv2__Image__setIccProf
ile.jpg
 address | marker       |  length | data
       0 | 0xffd8 SOI
       2 | 0xffe2 APP2  |      16 | ICC_PROFILE.... chunk 0/0Exiv2 exception in print action for file http://d
ev.exiv2.org/attac
hments/download/1087/exiv2-oob-heap-Exiv2__Image__setIccProfile.jpg:
Failed to read image data

C:\Users\rmills\gnu\exiv2\trunk\msvc2005\bin\x64\releasedll>
```

#6 - 21 Oct 2016 19:28 - Robin Mills

- Estimated time changed from 1.00 h to 2.00 h

r4653. Fixing GCC compiler warning. Add Hanno's test file to the test suite.

#7 - 22 Oct 2016 10:43 - Robin Mills

- Estimated time changed from 2.00 h to 6.00 h

I've spent more time looking into this. I discovered (to my horror) that the command: ***$ exiv2 -pR test/data/exiv2-bug1247.jpg*** crashes on Windows when I use Hanno's patch. I could modify exiv2.cpp as follows:

```
void signalHandler(int signal)
{
    const char* reason = "!Access Violation!";
    std::cerr << "signalHandler " << signal <<" "<< reason << std::endl;
    throw std::exception(reason);
}

// *****************************************************************************
// Main
int main(int argc, char* const argv[])
try {
#ifdef EXV_ENABLE_NLS
    setlocale(LC_ALL, "");
    bindtextdomain(EXV_PACKAGE, EXV_LOCALEDIR);
    textdomain(EXV_PACKAGE);
#endif

    typedef void (*signalHandlerPointer)(int);

    signalHandlerPointer previousHandler = signal(SIGSEGV , signalHandler);

    // Handle command line arguments
```

```
   ... no changes to the rest of the code ...
    // Return a positive one byte code for better consistency across platforms
    return static_cast<unsigned int>(rc) % 256;
} catch ( std::exception& e ) {
    std::cerr << "std::exception " << e.what() << std::endl;
    return 99;
} // main
```

When the code hits an access violation, a signal is emitted and that's what's being handled by signalHandler() where it is converted to an exception.

I'm not going to add this code to exiv2(.exe) and all the other sample applications at this time for 2 reasons:

1 It is very late in the Exiv2 v0.26 project.
2 This is a process mechanism. Adding signal handlers inside functions such as image::printStructure() or image::readMetadata() is tricky. We have to restore existing signal handlers when we return from those functions.

Perhaps a simpler approach is to write ***Image::lint()*** The function ***Image::lint()*** would call ***Image::readMetadata()*** and ***Image::printStructure()*** without modification. ***Image::lint()*** can be called by the ImageFactory to review an image. The function image::lint() can install/restore signal handlers.

I'm also wondering about a couple matters:

1) Why does the Mac/Clang report exceptions when cl/msvc signals? Perhaps clang converts signals to exceptions by default.

2) When I run tests in the Terminal/console, Windows displays asserts and signals as a dialog box. When I'm running the tests on the buildserver (jenkins), asserts and signals are reported in the transcript. How does it do that? Is there something we can set in a Windows process to cause this behaviour?

#8 - 22 Oct 2016 17:21 - Robin Mills

- Estimated time changed from 6.00 h to 7.00 h

There is a major difference in signal/exception handling between GCC and Clang.

Clang:

```
1013 rmills@rmillsmbp-kubuntu:~/gnu/exiv2/trunk $ exiv2 -pa test/data/exiv2-bug1247.jpg
Exiv2 exception in print action for file test/data/exiv2-bug1247.jpg:
Not a valid ICC Profile
1014 rmills@rmillsmbp-kubuntu:~/gnu/exiv2/trunk $ exiv2 -pR test/data/exiv2-bug1247.jpg
STRUCTURE OF JPEG FILE: test/data/exiv2-bug1247.jpg
 address | marker       |  length | data
       0 | 0xffd8 SOI
       2 | 0xffe2 APP2  |      16 | ICC_PROFILE.... chunk 0/0
      18 | 0xfffffffffff (null)Exiv2 exception in print action for file test/data/exiv2-bug1247.jpg:
This does not look like a JPEG image      <---- This is an Exiv2 exception handler message
1015 rmills@rmillsmbp-kubuntu:~/gnu/exiv2/trunk $
1015 rmills@rmillsmbp-kubuntu:~/gnu/exiv2/trunk $ exiv2 -vVg svn -g compiler
exiv2 0.25 001900 (64 bit build)
compiler=Clang
svn=4655
1016 rmills@rmillsmbp-kubuntu:~/gnu/exiv2/trunk $ grep "does not look" src/*.cpp
src/error.cpp:        {  3, N_("This does not look like a %1 image") }, // %1=Image type
src/error.cpp:        { 15, N_("This does not look like a JPEG image") },
src/error.cpp:        { 33, N_("This does not look like a CRW image") },
1017 rmills@rmillsmbp-kubuntu:~/gnu/exiv2/trunk $
```

GCC:

```
1017 rmills@rmillsmbp-kubuntu:~/gnu/exiv2/trunk $ exiv2 -pa test/data/exiv2-bug1247.jpg
Exiv2 exception in print action for file test/data/exiv2-bug1247.jpg:

Not a valid ICC Profile

1018 rmills@rmillsmbp-kubuntu:~/gnu/exiv2/trunk $ exiv2 -pR test/data/exiv2-bug1247.jpg

STRUCTURE OF JPEG FILE: test/data/exiv2-bug1247.jpg

 address | marker       |  length | data

       0 | 0xffd8 SOI

       2 | 0xffe2 APP2  |      16 | ICC_PROFILE.... chunk 0/0
```

```
Segmentation fault (core dumped)     <--- segmentation fault

1019 rmills@rmillsmbp-kubuntu:~/gnu/exiv2/trunk $ exiv2 -vVg svn -g compiler
exiv2 0.25 001900 (64 bit build)
compiler=G++
svn=4655
1020 rmills@rmillsmbp-kubuntu:~/gnu/exiv2/trunk $
```

I also discovered (on Stackoverlow) that you can hook the CRT to install your own report handler ***_CrtSetReportHook()***.
http://stackoverflow.com/questions/13943665/how-can-i-disable-the-debug-assertion-dialog-on-windows

This is interesting - however it requires modifying the code of the host application. That doesn't explain how Jenkins achieves this. Perhaps Jenkins hooks the CRT before using CreateProcess() to start a sub-process which inherits the hook. That doesn't seem likely to me. I think there's something in the process itself. Anyway, we don't need to solve this today. A search of the Jenkins source will reveal the necessary magic.

Files

exiv2-oob-heap-Exiv2__Image__setIccProfile.jpg	17 Bytes	21 Oct 2016	Hanno Böck
exiv2-fix-oob-setIccProfile.diff	541 Bytes	21 Oct 2016	Hanno Böck