Exiv2
C++ metadata library

Exiv2 v0.24.1

Team Meeting to discuss KDE Security Issue

Project Status

- v0.23 released April 2012
  Canon CR2. Pentax DNG. Bug Fixes

- v0.24 released December 2013
  Included the GSoC2012 video-read code
  Bug Fixes and many improvements

- v0.25 code complete December 2014
  Many bug fixes, functional changes and new features
  Many maker-note updates
  Option to build GSoC2013 cloud ready code (http etc)
  GSoC2013 video-write code deferred to v0.26

- v0.26 over the horizon
  GSoC2013 video-write
  GSoC2015 webp/webm
  Update to latest (and external) XMPsdk
  Accessory ‘overflow’ file

Meeting Objective

- To decide how to address KDE’s Security Issue
- *** No other meeting objective ***
- Duration: 1 hour maximum
- No finger pointing, blaming or complaining
- Other useful discussion items if time permits:
  Introduce new team members
  Discuss v0.25
  Discuss v0.26
  Discuss Exiv2 Branding

KDE Security Issue

makosoft@gmail.com

Someone filed a Debian bug report for the first one a few weeks back and I expect they'll find the others shortly if they haven't already: https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=781123 (This was "fixed" a while back by increasing the buffer size, which didn't actually fix the security issue.) Even though all of these are video metadata related and KDE only uses exiv2 for image files, they still affect KDE because if you rename a video file to .jpg, KDE thinks it's an image and uses exiv2 to get metadata from it.

The others are a heap buffer overflow in nikonTagsParser here:

```
io_->read(buf.pData_, 2);
tagID = Exiv2::getULong(buf.pData_, littleEndian);
io_->read(buf.pData_, 2);
dataSize = Exiv2::getULong(buf.pData_, littleEndian);
temp -= (4 + dataSize);
```

A stack buffer overflow (underflow?) here:

```
while(dataSize) {
        std::memset(buf.pData_, 0x0, buf.size_);
        io_->read(buf.pData_, 1);
        str[(4 - dataSize) * 2] = (char)(Exiv2::getULong(buf.pData_, littleEndian) + 48);
        --dataSize;
}
```

This pair is particularly nasty because the stack buffer overflow allows the address of the heap buffer to be partially overwritten, making it possible to write to other areas of memory, and because the stack is executable in certain libexiv2-based applications. With ASLR disabled, I can get full arbitrary code execution by using this to overwrite a vtable pointer, and I reckon it should be possible to exploit this on real systems with ASLR on (it's just really fiddly to do -

I believe there's another heap buffer overflow in the AsfVideo::codecList here:

```
io_->read(buf.pData_, 2);
descLength = Exiv2::getUShort(buf.pData_, littleEndian) * 2;
io_->read(buf.pData_, descLength);
```

Current Situation

- Robin has offered that Exiv2 will produce a dot release

- This meeting should decide on our actions:
  1 Who will 'own' this matter
  2 Who will conduct the discussion with KDE
  3 Options to address this matter
  4 Assignments to Team Members
  5 Schedule

- Caution:
  We will discuss assignments and schedules today
  We will review them in response to KDE feedback

Analysis of Security Issue

- Isolate video code behind non-default build option (as already done for webready in v0.25)
- Video code requires simplification:
  - refactoring into functions
  - constants defined using enums in headers
  - implement code review recommendations
  - test HUGE (> 3.2G) video files over http
- Harden video code to resist suspicious files
- Lever webready to test HUGE video files
  - to stress test video ***AND*** webready
  - to not increase the size of the tar-ball

Options for v0.24.1

- The video code should be either:
  1) declared “experimental” and not built by default
  2) removed and reintroduced in v0.26 (or later)

- Start with v0.24
  - do as little work as possible to isolate the video code
  - add selected trunk fixes (such as maker notes)
  This can be done quickly if the selection is small
  Time Guestimate: 2 weeks

- Forget v0.24.1. Go directly to Release v0.25
  - isolate the video-read code
  This cannot be done quickly. v0.25 has lots of new code (webready), many bug fixes and new features
  Time Guestimate: Unknown

Exiv2 Project Status

- exiv2.org Redmine Forum/Wiki/Issues
  - lots of user activity
  - we are very responsive
- The code is widely used
- exiv2.dyndns.org:8080 Jenkins Build Server works well
- Our release interval is getting longer
  - now 18 months, was 12 months, aim for 6 months.
- The team is growing: 3 new hires in 2015
  Thomas S Thomas B Alan
- Islam's GSoC 2015 webp/webm project
- Robin is overloaded

Exiv2 v0.26

Video Read and Write	Abhinav and Mahesh
webp/webm	Islam
Accessory ‘overflow’ file	Thomas B
XMPsdk as an external library	Andreas
Coverity Scan	Mahesh
CMake/MSVC Android and iOS	Robin
More features and bug fixes	Alan and Thomas S
Server & Release Management	Nehal and Shawn
Branding, Logo, Website Makeover	Ocean

The Future of Exiv2

- Exiv2 is an excellent open source project
  - Useful
  - Highly functional
  - Reliable

- Excellent C++ code and test harness:
  - Andreas Brad
  - Gilles Volker

- Excellent updates for cameras and accessories:
  - Niels Thomas B (v0.26)

- Excellent build, developer and user support:
  - Robin Alan

- Exiv2 Team Work can be improved by:
  - Me, you, users, contributors