**TIFF CHANGE INFORMATION**

**Current Version**: v4.0.7  
**Previous Version**: [v4.0.6](http://docs.google.com/v4.0.6.html)  
**Master FTP Site**:  [download.osgeo.org](ftp://download.osgeo.org/libtiff), directory pub/libtiff  
**Master HTTP Site #1**: <http://www.simplesystems.org/libtiff/>  
**Master HTTP Site #2**: <http://libtiff.maptools.org/>

This document describes the changes made to the software between the *previous* and *current* versions (see above). If you don't find something listed here, then it was not done in this timeframe, or it was not considered important enough to be mentioned. The following information is located here:

* [Major Changes](#gjdgxs)
* [Changes in the software configuration](#30j0zll)
* [Changes in libtiff](#1fob9te)
* [Changes in the tools](#3znysh7)
* [Changes in the contrib area](#2et92p0)

**MAJOR CHANGES:**

* The libtiff tools bmp2tiff, gif2tiff, ras2tiff, sgi2tiff, sgisv, and ycbcr are completely removed from the distribution. These tools were written in the late 1980s and early 1990s for test and demonstration purposes. In some cases the tools were never updated to support updates to the file format, or the file formats are now rarely used. In all cases these tools increased the libtiff security and maintenance exposure beyond the value offered by the tool.

**CHANGES IN THE SOFTWARE CONFIGURATION:**

* None

**CHANGES IN LIBTIFF:**

* libtiff/tif\_dirread.c: in TIFFFetchNormalTag(), do not dereference NULL pointer when values of tags with TIFF\_SETGET\_C16\_ASCII / TIFF\_SETGET\_C32\_ASCII access are 0-byte arrays. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2593 (regression introduced by previous fix done on 2016-11-11 for CVE-2016-9297). Reported by Henri Salo. Assigned as CVE-2016-9448
* libtiff/tif\_aux.c: fix crash in TIFFVGetFieldDefaulted() when requesting Predictor tag and that the zip/lzw codec is not configured. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2591
* libtiff/tif\_dirread.c: in TIFFFetchNormalTag(), make sure that values of tags with TIFF\_SETGET\_C16\_ASCII / TIFF\_SETGET\_C32\_ASCII access are null terminated, to avoid potential read outside buffer in \_TIFFPrintField(). Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2590
* libtiff/tif\_dirread.c: reject images with OJPEG compression that have no TileOffsets/StripOffsets tag, when OJPEG compression is disabled. Prevent null pointer dereference in TIFFReadRawStrip1() and other functions that expect td\_stripbytecount to be non NULL. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2585
* libtiff/tif\_strip.c: make TIFFNumberOfStrips() return the td->td\_nstrips value when it is non-zero, instead of recomputing it. This is needed in TIFF\_STRIPCHOP mode where td\_nstrips is modified. Fixes a read outsize of array in tiffsplit (or other utilities using TIFFNumberOfStrips()). Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2587 (CVE-2016-9273)
* libtiff/tif\_predict.h, libtiff/tif\_predict.c: Replace assertions by runtime checks to avoid assertions in debug mode, or buffer overflows in release mode. Can happen when dealing with unusual tile size like YCbCr with subsampling. Reported as MSVR 35105 by Axel Souchet & Vishal Chauhan from the MSRC Vulnerabilities & Mitigations
* libtiff/tif\_dir.c: discard values of SMinSampleValue and SMaxSampleValue when they have been read and the value of SamplesPerPixel is changed afterwards (like when reading a OJPEG compressed image with a missing SamplesPerPixel tag, and whose photometric is RGB or YCbCr, forcing SamplesPerPixel being 3). Otherwise when rewriting the directory (for example with tiffset, we will expect 3 values whereas the array had been allocated with just one), thus causing a out of bound read access. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2500 (CVE-2014-8127, duplicate: CVE-2016-3658)
* libtiff/tif\_dirwrite.c: avoid null pointer dereference on td\_stripoffset when writing directory, if FIELD\_STRIPOFFSETS was artificially set for a hack case in OJPEG case. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2500 (CVE-2014-8127, duplicate: CVE-2016-3658)
* libtiff/tif\_getimage.c (TIFFRGBAImageOK): Reject attempts to read floating point images.
* libtiff/tif\_predict.c (PredictorSetup): Enforce bits-per-sample requirements of floating point predictor (3). Fixes CVE-2016-3622 "Divide By Zero in the tiff2rgba tool."
* libtiff/tif\_pixarlog.c: fix out-of-bounds write vulnerabilities in heap allocated buffers. Reported as MSVR 35094. Discovered by Axel Souchet and Vishal Chauhan from the MSRC Vulnerabilities & Mitigations team.
* libtiff/tif\_write.c: fix issue in error code path of TIFFFlushData1() that didn't reset the tif\_rawcc and tif\_rawcp members. I'm not completely sure if that could happen in practice outside of the odd behaviour of t2p\_seekproc() of tiff2pdf). The report points that a better fix could be to check the return value of TIFFFlushData1() in places where it isn't done currently, but it seems this patch is enough. Reported as MSVR 35095. Discovered by Axel Souchet & Vishal Chauhan & Suha Can from the MSRC Vulnerabilities & Mitigations team.
* libtiff/tif\_pixarlog.c: Fix write buffer overflow in PixarLogEncode if more input samples are provided than expected by PixarLogSetupEncode. Idea based on libtiff-CVE-2016-3990.patch from libtiff-4.0.3-25.el7\_2.src.rpm by Nikola Forro, but with different and simpler check. (bugzilla #2544)
* libtiff/tif\_read.c: Fix out-of-bounds read on memory-mapped files in TIFFReadRawStrip1() and TIFFReadRawTile1() when stripoffset is beyond tmsize\_t max value (reported by Mathias Svensson)
* libtiff/tif\_read.c: make TIFFReadEncodedStrip() and TIFFReadEncodedTile() directly use user provided buffer when no compression (and other conditions) to save a memcpy()
* libtiff/tif\_write.c: make TIFFWriteEncodedStrip() and TIFFWriteEncodedTile() directly use user provided buffer when no compression to save a memcpy().
* libtiff/tif\_luv.c: validate that for COMPRESSION\_SGILOG and PHOTOMETRIC\_LOGL, there is only one sample per pixel. Avoid potential invalid memory write on corrupted/unexpected images when using the TIFFRGBAImageBegin() interface (reported by Clay Wood)
* libtiff/tif\_pixarlog.c: fix potential buffer write overrun in PixarLogDecode() on corrupted/unexpected images (reported by Mathias Svensson) (CVE-2016-5875)
* libtiff/libtiff.def: Added \_TIFFMultiply32 and \_TIFFMultiply64 to libtiff.def
* libtiff/tif\_config.vc.h (HAVE\_SNPRINTF): Add a '1' to the HAVE\_SNPRINTF definition.
* libtiff/tif\_config.vc.h (HAVE\_SNPRINTF): Applied patch by Edward Lam to define HAVE\_SNPRINTF for Visual Studio 2015.
* libtiff/tif\_dirread.c: when compiled with DEFER\_STRILE\_LOAD, fix regression, introduced on 2014-12-23, when reading a one-strip file without a StripByteCounts tag. GDAL #6490
* libtiff/\*: upstream typo fixes (mostly contributed by Kurt Schwehr) coming from GDAL internal libtiff
* libtiff/tif\_fax3.h: make Param member of TIFFFaxTabEnt structure a uint16 to reduce size of the binary.
* libtiff/tif\_read.c, tif\_dirread.c: fix indentation issues raised by GCC 6 -Wmisleading-indentation
* libtiff/tif\_pixarlog.c: avoid zlib error messages to pass a NULL string to %s formatter, which is undefined behaviour in sprintf().
* libtiff/tif\_next.c: fix potential out-of-bound write in NeXTDecode() triggered by http://lcamtuf.coredump.cx/afl/vulns/libtiff5.tif (bugzilla #2508)
* libtiff/tif\_luv.c: fix potential out-of-bound writes in decode functions in non debug builds by replacing assert()s by regular if checks (bugzilla #2522). Fix potential out-of-bound reads in case of short input data.
* libtiff/tif\_getimage.c: fix out-of-bound reads in TIFFRGBAImage interface in case of unsupported values of SamplesPerPixel/ExtraSamples for LogLUV / CIELab. Add explicit call to TIFFRGBAImageOK() in TIFFRGBAImageBegin(). Fix CVE-2015-8665 reported by limingxing and CVE-2015-8683 reported by zzf of Alibaba.
* libtiff/tif\_dirread.c: workaround false positive warning of Clang Static Analyzer about null pointer dereference in TIFFCheckDirOffset().
* libtiff/tif\_fax3.c: remove dead assignment in Fax3PutEOLgdal(). Found by Clang Static Analyzer
* libtiff/tif\_dirwrite.c: fix truncation to 32 bit of file offsets in TIFFLinkDirectory() and TIFFWriteDirectorySec() when aligning directory offsets on a even offset (affects BigTIFF). This was a regression of the changeset of 2015-10-19.
* libtiff/tif\_write.c: TIFFWriteEncodedStrip() and TIFFWriteEncodedTile() should return -1 in case of failure of tif\_encodestrip() as documented
* libtiff/tif\_dumpmode.c: DumpModeEncode() should return 0 in case of failure so that the above mentionned functions detect the error.
* libtiff/\*.c: fix MSVC warnings related to cast shortening and assignment within conditional expression
* libtiff/\*.c: fix clang -Wshorten-64-to-32 warnings
* libtiff/tif\_dirread.c: prevent reading ColorMap or TransferFunction if BitsPerPixel > 24, so as to avoid huge memory allocation and file read attempts
* libtiff/tif\_dirread.c: remove duplicated assignment (reported by Clang static analyzer)
* libtiff/tif\_dir.c, libtiff/tif\_dirinfo.c, libtiff/tif\_compress.c, libtiff/tif\_jpeg\_12.c: suppress warnings about 'no previous declaration/prototype'
* libtiff/tiffiop.h, libtiff/tif\_dirwrite.c: suffix constants by U to fix 'warning: negative integer implicitly converted to unsigned type' warning (part of -Wconversion)
* libtiff/tif\_dir.c, libtiff/tif\_dirread.c, libtiff/tif\_getimage.c, libtiff/tif\_print.c: fix -Wshadow warnings (only in libtiff/)

**CHANGES IN THE TOOLS:**

* tools/Makefile.am: The libtiff tools bmp2tiff, gif2tiff, ras2tiff, sgi2tiff, sgisv, and ycbcr are completely removed from the distribution. The libtiff tools rgb2ycbcr and thumbnail are only built in the build tree for testing. Old files are put in new 'archive' subdirectory of the source repository, but not in distribution archives. These changes are made in order to lessen the maintenance burden.
* tools/tiff2pdf.c: avoid undefined behaviour related to overlapping of source and destination buffer in memcpy() call in t2p\_sample\_rgbaa\_to\_rgb() Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2577
* tools/tiff2pdf.c: fix potential integer overflows on 32 bit builds in t2p\_read\_tiff\_size() Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2576
* tools/fax2tiff.c: fix segfault when specifying -r without argument. Patch by Yuriy M. Kaminskiy. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2572
* tools/tiffinfo.c: fix out-of-bound read on some tiled images. (http://bugzilla.maptools.org/show\_bug.cgi?id=2517)
* tools/tiffcrop.c: fix multiple uint32 overflows in writeBufferToSeparateStrips(), writeBufferToContigTiles() and writeBufferToSeparateTiles() that could cause heap buffer overflows. Reported by Henri Salo from Nixu Corporation. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2592
* tools/tiffcrop.c: fix out-of-bound read of up to 3 bytes in readContigTilesIntoBuffer(). Reported as MSVR 35092 by Axel Souchet & Vishal Chauhan from the MSRC Vulnerabilities & Mitigations team.
* tools/tiff2pdf.c: fix write buffer overflow of 2 bytes on JPEG compressed images. Reported by Tyler Bohan of Cisco Talos as TALOS-CAN-0187 / CVE-2016-5652. Also prevents writing 2 extra uninitialized bytes to the file stream.
* tools/tiffcp.c: fix out-of-bounds write on tiled images with odd tile width vs image width. Reported as MSVR 35103 by Axel Souchet and Vishal Chauhan from the MSRC Vulnerabilities & Mitigations team.
* tools/tiff2pdf.c: fix read -largely- outsize of buffer in t2p\_readwrite\_pdf\_image\_tile(), causing crash, when reading a JPEG compressed image with TIFFTAG\_JPEGTABLES length being one. Reported as MSVR 35101 by Axel Souchet and Vishal Chauhan from the MSRC Vulnerabilities & Mitigations team.
* tools/tiffcp.c: fix read of undefined variable in case of missing required tags. Found on test case of MSVR 35100.
* tools/tiffcrop.c: fix read of undefined buffer in readContigStripsIntoBuffer() due to uint16 overflow. Probably not a security issue but I can be wrong. Reported as MSVR 35100 by Axel Souchet from the MSRC Vulnerabilities & Mitigations team.
* tools/tiffcrop.c: fix various out-of-bounds write vulnerabilities in heap or stack allocated buffers. Reported as MSVR 35093, MSVR 35096 and MSVR 35097. Discovered by Axel Souchet and Vishal Chauhan from the MSRC Vulnerabilities & Mitigations team.
* tools/tiff2pdf.c: fix out-of-bounds write vulnerabilities in heap allocate buffer in t2p\_process\_jpeg\_strip(). Reported as MSVR 35098. Discovered by Axel Souchet and Vishal Chauhan from the MSRC Vulnerabilities & Mitigations team.
* tools/tiff2bw.c: fix weight computation that could result of color value overflow (no security implication). Fix bugzilla #2550. Patch by Frank Freudenberg.
* tools/rgb2ycbcr.c: validate values of -v and -h parameters to avoid potential divide by zero. Fixes CVE-2016-3623 (bugzilla #2569)
* tools/tiffcrop.c: Fix out-of-bounds write in loadImage(). From patch libtiff-CVE-2016-3991.patch from libtiff-4.0.3-25.el7\_2.src.rpm by Nikola Forro (bugzilla #2543)
* tools/tiff2rgba.c: Fix integer overflow in size of allocated buffer, when -b mode is enabled, that could result in out-of-bounds write. Based initially on patch tiff-CVE-2016-3945.patch from libtiff-4.0.3-25.el7\_2.src.rpm by Nikola Forro, with correction for invalid tests that rejected valid files. (bugzilla #2545)
* tools/tiffcrop.c: Avoid access outside of stack allocated array on a tiled separate TIFF with more than 8 samples per pixel. Reported by Kaixiang Zhang of the Cloud Security Team, Qihoo 360 (CVE-2016-5321 / CVE-2016-5323 , bugzilla #2558 / #2559)
* tools/tiffdump.c: fix a few misaligned 64-bit reads warned by -fsanitize
* tools/tiffdump.c (ReadDirectory): Remove uint32 cast to \_TIFFmalloc() argument which resulted in Coverity report. Added more mutiplication overflow checks.

**CHANGES IN THE CONTRIB AREA:**

* None

Last updated $Date: 2016-11-19 17:47:40 $.