Inspection and Sanitization Guidance for TIFF File Formats

Version 1.1.1
16 November 2017

National Security Agency
Information Assurance Capabilities
9800 Savage Rd, Suite 6699
Ft. George G. Meade. MD 20755

Authored/Released by:
Unified Cross Domain Capabilities Office

cds_tech@nsa.gov

DOCUMENT REVISION HISTORY

Date	Version	Description
02/03/2015	1.0	Final
11/16/2017	1.1.1	Revised Formatting. Added Camera RAW information.
12/13/2017	1.1.1	Updated Contact information, IAC Logo, Cited Trademarks and Copyrights, Expanded Acronyms, and added Legal Disclaimer

Disclaimer

The information and opinions contained in this document are provided "as is" and without any warranties or guarantees. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer or otherwise, does not constitute or imply its endorsement, recommendation or favoring by the United States Government and this guidance shall not be used for advertising or product endorsement purposes

EXECUTIVE SUMMARY

The Inspection and Sanitization Guidance (ISG) for Tagged Image File Format (TIFF) provides guidelines and specifications for developing file inspection and sanitization software for TIFF files. Specifically, this ISG covers files as defined by TIFF Revision 6.0, authored by the Adobe Developers Association; the GeoTIFF Format Specifications, authored by Dr.
Niles Ritter and Mike Ruth; and the BigTIFF Specification proposal1 documented by Aware Systems. TIFF is not an acronym and does not stand for anything.[2]

TIFF serves as a container for images and image related data. It is a raster graphics format; the bytes within the pixel array correspond to pixels in an image. The file format supports image layers, pages, lossy and lossless compression. TIFF is used to describe data from scanners, frame grabbers, and image editing programs. It supports the inclusion of private and special purpose information. For example, the DateTime tag can store the date and time the image was taken or created.

The TIFF file format is also leveraged by camera manufacturers for raw camera data. Camera RAW refers to a file format that contains the values that a camera sensor captures; this is a grayscale image that records the intensity of the light that the sensor measured. This information is then translated into color images using a color filter that may be proprietary; this information is generally included within the file format, but is encrypted.

This guidance document examines the TIFF specifications for data attack, data hiding, and data disclosure risks that exist within the file structure. It provides a breakdown of each component of a TIFF file and provides recommendations that can help assure that the TIFF file is not only compliant with the specifications, but also free of any risks.

[1] BigTIFF does not have an official specification, however the proposed format has been accepted by software libraries like LibTIFF
[2] TIFF Revision 5.0 was the last version of the specifications to refer to TIFF as an acronym; there is no reference starting with TIFF Revision 6.0

TABLE OF CONTENTS

LIST OF TABLES

LIST OF TABLES

1. SCOPE

1.1 Purpose

The purpose of this Inspection, Sanitization, and Guidance (ISG) document is to provide guidance for the development of a sanitization and analysis software tool for different versions of TIFF, BigTIFF, and GeoTIFF. This document analyzes various elements and objects that are contained within the TIFF file structure and then discusses data hiding, data attack, and data disclosure risks. It will describe how these elements can be a cause for concern from hidden, sensitive data or from possible attempts to exploit a system. This document provides numerous recommendations and mitigations that could be used to ensure the TIFF file is safer and more accurately conforms to the specification.

The intended audience of this document includes system engineers, designers, software developers, and testers who work with file inspection and sanitization applications that process TIFF files.

1.2 Introduction

File types that act as containers and store a variety of data introduce a significant amount of risk including data hiding, data disclosure, and data attack risks. TIFF is an example of one of these file types because it can hold both images and text.

The TIFF image format utilizes a growable architecture that can store both image data and tags. A growable architecture is one where an arbitrary amount of information can be inserted into the file format, while retaining similar functionality across image rendering applications. Tags contain information about the image, e.g. compression method, copyrights, etc. The file structure can be modified to include multiple images, compressed using different methods; these TIFF files are commonly known as multipage TIFFs. The structure of a TIFF file can be amended to include custom information and custom tags; however, TIFF renderers are only required to support the baseline TIFF features. Complex file types, like TIFF, that are growable can be used to take advantage of vulnerabilities within applications; these files must be inspected for correctness to identify and/or mitigate these vulnerabilities.

1.3 Background

The goal of TIFF is to "provide a rich environment within which applications can exchange image data…to take advantage of the varying capabilities of scanners and other imaging devices" [1]. The Aldus Corporation first published the TIFF Specification in 1986. Prior to that, there were two drafts, which were incorporated in the first published version, Revision 3.0. The next two specification revisions introduced new minor features. Aldus eventually merged with Adobe, who has managed the file format specification since that time.

This document focuses on the most recent TIFF specification, Revision 6.0, which was published on June 3, 1992. TIFF files commonly use the ".tif" or ".tiff" file extensions.

BigTIFF is an extension to the TIFF file format that modifies offsets to use eight bytes as opposed to the TIFF specifications, which define four-byte offsets. BigTIFF files can exceed 4GB in size. GeoTIFF is an extension to TIFF that contains georeferencing metadata; it takes advantage of the growable TIFF architecture to provide structures that can be utilized by specific programs and still be properly rendered by baseline applications.

Camera Raw leverages the TIFF file format to encapsulate raw camera data. Each camera manufacturers use proprietary versions of JPEG or TIFF to encapsulate raw camera data and proprietary information to recover the image. The other predominant Camera Raw format is the Camera Image File Format (CIFF); CIFF is JPEG based and is considered out of this document's scope. Moreover, the RAW information found in this document leveraged open source efforts to reverse proprietary Camera Raw protocols.

1.4 Document Organization

This section summarizes the organization of this document.

Table 1-1 Document Organization

Section	Description
Section 1: Scope	This section describes the purpose, introduction, background, organization, actions, and limitations related to this document.
Section 2: Constructs and Taxonomy	This section describes the constructs and taxonomy that are used throughout this document.
Section 3: Overview	This section describes the structure of TIFF, GeoTIFF, and BigTIFF.
Section 4: TIFF Constructs	This section contains the TIFF constructs that have risks and the options for mitigation.
Section 5: Acronyms	This section lists the acronyms in this document.
Section 6: Referenced Documents	This section lists the sources that were used to prepare this document.
Section 7: Summary of Risks	This section maps each construct to the corresponding specifications and risks.

1.5 Actions

Each construct description lists recommended actions for handling the construct when processing a message. Generally, inspection and sanitization programs will perform one of these actions on a construct: Validate, Remove, Replace, External Filtering Required, Review, or Reject.

The recommendation section in each construct lists each action that is applicable along with an explanation that is specific to the construct. Not all actions are applicable or appropriate for every context. As such, implementers are not expected to implement all the actions for a given risk; instead, they are expected to determine which action—or perhaps actions—applies best to their context. Definition of the criteria used to determine which action is "best" and of the specific method used to execute the action is left to the implementer.

Recommendations such as Remove and Replace may alter the integrity of TIFF files. It is important to address these issues in order to retain functionality.

NOTE

The recommendations in this document are brief explanations rather than a How--To Guide. Readers should refer to the construct description or official documentation for additional details.

This table summarizes the recommendation actions:

Table 1-2 Recommendation Actions

Recommendation Action	Comments
Validate	Verify the data structure's integrity, which may include integrity checks on other components in the message. (This should almost always be a recommended action.)
Replace	Replace the data structure or one or more of its elements with values that alleviate the risk (e.g., replacing a username with a non-identifying, harmless value or substituting a common name for all authors).
Remove	Remove the data structure or one or more of its elements and any other affected parts.
External Filtering Required	Note the data type and pass the data to an external action for handling that data type (e.g., extract text and pass it to a dirty word search).
Review	Present the data structure or its constructs for a human to review. (This should almost always be recommended if the object being inspected can be revised by a human.)
Reject	Reject the message.

NOTE

No recommendations for logging all actions and found data are included here because all activity logging in an inspection application should occur "at an appropriate level" and presented in a form that a human can analyze further (e.g., the audit information may be stored in any format but must be parsable and provide enough information to address the issue when presented to a human.)

1.6 Document Limitations

This document covers TIFF Revision 6.0, GeoTIFF Format Specification Revision 1.0, and BigTIFF File Format Proposal, documented by AWare Systems; BigTIFF Design, documented by LibTIFF; and PageMaker® 6.0 TIFF Technical Notes. This document also addresses Camera Raw; the Digital Negative Specification and open source research on proprietary camera raw formats were consulted.

There are many derivative specifications that take advantage of the original TIFF architecture to form a variation tailored to a specific use case. Because these specifications prescribe modifications that are still compliant to the original TIFF structure, the risks addressed in this document encompass derivative TIFF specifications. These include TIFF-F RFC 2306, TIFF-FX RFC 3950, TIFF/EP (ISO 122342), TIFF/IT (ISO 12639), etc.

1.6.1 Covert Channel Analysis

It is impossible to identify all available covert channels, whether in a file format or a communication protocol. Because they contain free-form text, searching for hidden data becomes increasingly difficult. No tool can possibly analyze every channel, so this document highlights the highest risk areas to reduce or eliminate data spills and malicious content.

Additionally, this document does not discuss steganography within block text or media files, such as a hidden message that is embedded within an innocuous image or paragraph. Separate file format filters that specialize in steganography should be used to handle embedded content, such as text, images, videos, and audio.

However, it is important to note that for RAW camera file formats typical steganography mitigations, e.g. Least Bit Stripping, may impact the usability of the image and degrade the process of converting the raw data to an standard non-raw image.

2. CONSTRUCTS AND TAXONOMY

2.1 Constructs

This document describes many of the constructs used in TIFF file formats, but it does not describe every construct, thus this document is not to be treated as a complete reference. Developers of a TIFF filter should consult the official specifications alongside this documentation for the full context. For each construct that is mentioned, the following sections exist:

- Overview: An explanation of the construct with examples.
- Risks and Recommendations: An explanation of potential risks posed by the construct with corresponding mitigation strategies.
- Product: The specifications in which the construct is found.
- Location: A textual description of where to find the construct.

2.2 Taxonomy

The following table describes the terms that appear in this document:

Table 2-1 Document Taxonomy

Term	Definition
Construct	An object that represents some form of information or data in the hierarchy of a TIFF image.
Inspection and Sanitization	Activities for processing files and protocols to prevent inadvertent data leakage, data exfiltration, and malicious data or code transmission
ISG	A document (such as this) that details a file format or protocol and inspection and sanitization activities for constructs within it.
Recommendations	A series of actions for handling a construct when performing inspection and sanitization activities.

3. OVERVIEW

TIFF is a data format designed to provide a rich environment where applications can exchange the information about images necessary for imaging devices (e.g., scanners) to interact with them. Image renderers rely on designated file offsets within the file format to properly process a TIFF file. All offsets use 0x0 as the origin point. The TIFF image data is organized into strips or tiles for faster random access and efficient I/O buffering. It is not uncommon for an entire image to be placed into a single tile or strip. The file format was designed to be extensible; this allows for custom tags and structures to be included in TIFF files. TIFF Revision 6.0 provides the guidelines for the basic tags and structures that should be processed by image renderers; the specification states that applications can ignoring data it does not recognize. TIFF can support layers, pages, and lossy and lossless compressed images.

3.1 Baseline TIFF File Structure

TIFF is composed of two primary structures: the image header and the image file directory (IFD). The image header points to the first IFD; each IFD points to the subsequent IFD or 0x0 if there are no additional IFDs. The IFD is composed of a variable number of IFD entries (also called tags). It is important to note that all offset locations are listed with respect to 0x0 (the beginning of the file). Some file formats may use offsets with respect to the current location; however, this is not the case for TIFF. The following subsections will enumerate the structures in a TIFF file; each structure is a logical grouping of bytes within the TIFF file format.

3.1.1 TIFF Image Header

The baseline TIFF image header makes up the first eight bytes of every TIFF file and contains pertinent information for processing the file. As shown in Figure 3-1, the first two bytes designate the byte order of the file or endianness. The following two bytes serve as the magic number it will always evaluate to 42 (0x2a). The next four bytes are referred to as the Image File Directory offset, which contains the location of the image directory relative to the address 0x0 (i.e., the beginning of the file). The Image File Directory offset will vary from file to file because it does not have to start directly after the TIFF Image Header. However, it does have to be aligned to a word boundary.

Figure 3-1 TIFF Image Header Structure

3.1.2 Image File Directory

The IFD is a simple structure that groups TIFF tags. It is located at the offset defined in the image header. The presence of two or more IFDs can be treated as a multi-layered or multi-paged image. A multi-paged image is a means of storing multiple images in a file and having the ability to page through the images when rendered correctly. A baseline TIFF renderer is not required to process more than one IFD.

The IFD is a variable sized structure that is dependent on the number of tags. The first two bytes of the directory will indicate the number of tags. There are then n twelve-byte tags, where n is the value of the first two bytes of the IFD. The final four bytes contain the offset to the next IFD or 0x0 if there are no more IFDs. Information can exist between the end of a given IFD and subsequent IFDs; this information is considered unreferenced data (see Section 4.6) because there is no pointer to this information in the file.

Figure 3-2 TIFF Image File Directory Structure

3.1.3 TIFF Tags

TIFF tags (also called IFD entries) are twelve-byte fields found within the IFD. TIFF tags can contain both information pertinent for image renders and information about the image, which is not directly related to rendering the image. For example, the date and time an image was created can be stored in the DateTime tag. Note that the name of the tag is only found in the specifications; in a TIFF file the tag is identified by a two-byte or four byte value. These values are correlated with names in the specification.

TIFF tags are composed of four fields: the tag identifier, the field type, the number of values (for the tag), and the value offset (see Figure 3-3). The possible values for each field are outlined in the baseline TIFF specifications. The tag identifier is a value that corresponds to a known tag definition (e.g., if the tag identifier is 0x103, then the following information deals with compression). For examples of tags, refer to Table 4-1, which enumerates the required tags (by name and value) per image type. Field types represent the type and size of each piece of data for the tag; some tags require specific field types. The baseline TIFF field types (see also BigTIFF Field Types) enumerate the values for the field type found in the specifications. The number of values field defines the number of values for the tag of that type. The TIFF specifications allow for custom TIFF tags; the only stipulation is that they must be well-formed (see TIFF Tag Format).

Tag Identifier *2 Bytes*	Field Type *2 Bytes*	Number of Values *4 Bytes*	Value Offset *4 Bytes*

Figure 3-3 TIFF Tag Structure

The image data for a TIFF file will be a block of data referenced by specific tags. For example, the StripOffsets tag points to the location of the image data. StripOffsets, as defined in the TIFF specification, would be represented as follows: the tag identifier would be 0x111, the field type should evaluate to either short or long, the number of values will be the total number of (short or long) values, and the value offset will contain the location of the image data. Because the image data can be found at the reference provided by the tag, the image data does not have to be in the same spot for every file, i.e. the image data can be at any location that does not overlap with another TIFF structure. Figure 3-4 shows the conceptual flow of a TIFF file; the image file header points to the image file directory and one of the tags, in this example it is StripOffsets, points to the pixel data containing the image stored in the file.

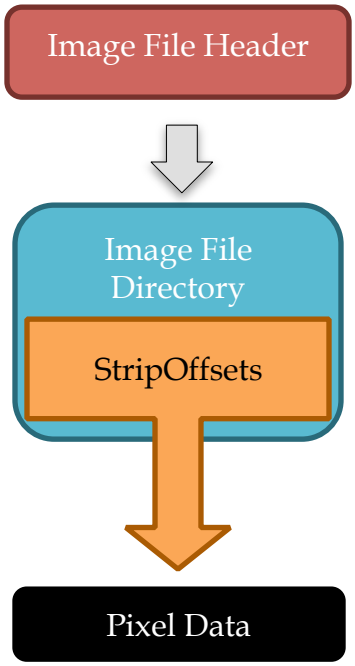

Figure 3-4 Sample TIFF Structure

3.1.3.1 Required TIFF Tags

The four different image classifications are bi-level, grey-scale, palette-color, and fullcolor. There is a minimum set of tags required for each image classification (see Table
4-1).

Basic TIFF renderers only require the tags enumerated in Table 4-1 in order to render an image of a certain type. Other tags may not deal with the rendering the image or may be an application specific tag.

3.1.3.2 Baseline TIFF Field Types

The values for each IFD entry will adhere to type definitions outlined in the baseline specifications. The possible types indicate the size of the value, as seen in the following table:

Table 3-1 Baseline TIFF Field Types

Type by Value	Size
1 = BYTE	8-bit unsigned integer.
2 = ASCII	8-bit byte that contains a 7-bit ASCII code; when there are multiple values the last byte must be NUL (binary zero).
3 = SHORT	16-bit (2-byte) unsigned integer.
4 = LONG	32-bit (4-byte) unsigned integer.
5 = RATIONAL	Two LONGs: the first represents the numerator of a fraction; the second, the denominator.
7 = UNDEFINED	An 8-bit byte that may contain anything, depending on the definition of the field.
8 = SSHORT	A 16-bit (2-byte) signed (twoscomplement) integer.
9 = SLONG	A 32-bit (4-byte) signed (twoscomplement) integer.
10 = SRATIONAL	Two SLONG's: the first represents the numerator of a fraction, the second the denominator.
11 = FLOAT	Single precision (4-byte) IEEE format.
12 = DOUBLE	Double precision (8-byte) IEEE format.

3.2 GeoTIFF Extension

The GeoTIFF extension defines a format for TIFF that includes georeferencing information (e.g., map projections and coordinate systems). The GeoTIFF specification defines two structures, the GeoKey Directory and GeoKeys, and six tags for processing GeoTIFF files. A TIFF application should be able to read and display a GeoTIFF file though it will not understand the Geo tags.

The GeoTIFF extension is the same as the TIFF structure except that it includes the two additional structures. Like TIFF, it contains an image file header and one or more image file directories. The presence of a particular tag, GeoKeyDirectoryTag, in one of the IFDs makes the file a GeoTIFF. The example below shows a conceptual outline of the GeoTIFF extension. Much like TIFF, a GeoTIFF file will have an image file header and an image file directory. All GeoTIFF files must contain the GeoKeyDirectoryTag, which will point to the location of the GeoKey directory. The image data for this sample file looks like Figure 3-4; there would be a tag in the image file directory pointing to the pixel data.

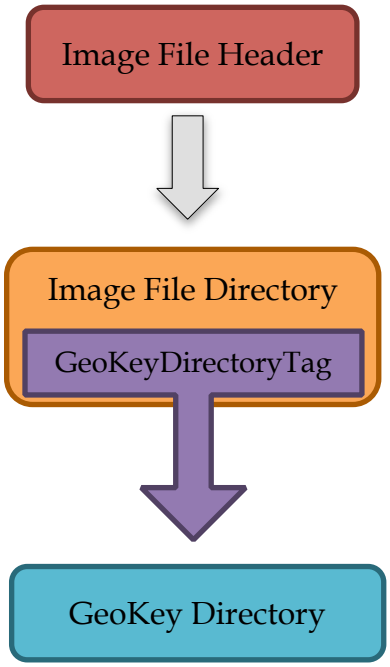

Figure 3-5 GeoTIFF Structure and Sample File

3.2.1 GeoTIFF Tags

The GeoTIFF tags have the same structure as Baseline TIFF tags. The new tags are listed in the table below. GeoKeyDirectoryTag, GeoDoubleParamsTag, and GeoAsciiParamsTag are discussed in the following sections; these tags, when present, will contain a variable number of values that require further analysis. ModelPixelScaleTag, ModelTransformationTag, and ModelTiePointTag all deal with mapping the raster image to a Model space; when the image is mapped to a model space, then it will correspond to reference points on Earth. Any specific geoinformation will be stored in the other three tags. In depth tag definitions are given in the GeoTIFF specifications [2].

Table 3-2 GeoTIFF Tags

Tag Name	Decimal Value
ModelPixelScaleTag	33550
ModelTransformationTag	34264
ModelTiepointTag	33922
GeoKeyDirectoryTag	34735
GeoDoubleParamsTag	34736
GeoAsciiParamsTag	34737

3.2.2 GeoKey Directory

The GeoKey Directory, shown in Figure 3-6, is structurally similar to the image file directory (Figure 3-2). It contains a field storing the number of GeoKeys and then lists each key. It is important to note that the entries are all of type SHORT, two-byte values; this is unlike the image file directory, which has fields of four or eight bytes. The key directory version field indicates the current version of the key implementation. The key revision and minor revision fields refer to the revision of keys and key codes that are used. The number of GeoKeys relays the number of GeoKey entries that will follow this field. The GeoKeyDirectoryTag points to the offset of the GeoKeyDirectory.

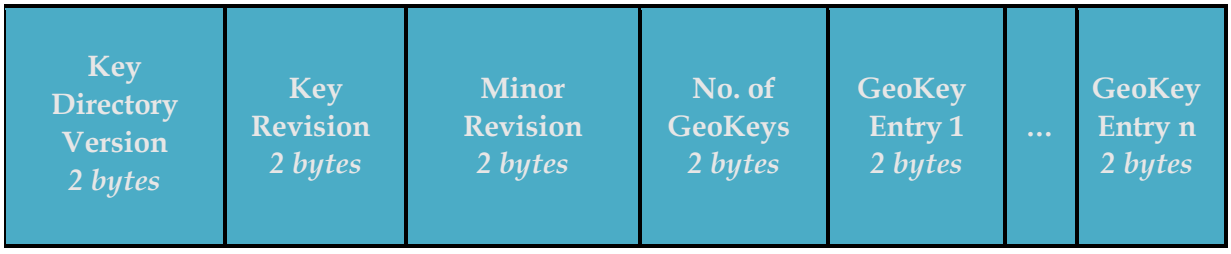

Figure 3-6 GeoTIFF Key Structure

3.2.3 GeoKey Entry

A GeoKey, or GeoTIFF key, is structurally similar to a tag, but is composed of unsigned SHORT, two-byte, values. The TIFFTagLocation refers to the tag that contains the values for this GeoKey. If it is 0, then the value type is SHORT and it is stored in the value offset. If the value type is DOUBLE or ASCII, then the TIFF tags should evaluate to GeoDoubleParamsTag or GeoAsciiParamsTag respectively, and the value offset will refer to the index location of the value. The values for the key will be found in the aforementioned tags (or at a location designated by the offset). Figure 3-7 is a visual representation of a single GeoTIFF key and the size of each entry.

Key ID *2 Bytes*	TIFF Tag Location *2 Bytes*	Number of Values *2 Bytes*	Value Offset *2 Bytes*

Figure 3-7 GeoTIFF Key Entry Structure

3.3 BigTIFF Extension

BigTIFF is an extension to the TIFF file format that modifies offsets to use eight bytes, which allows for more information to be stored in BigTIFF files. This allows for larger images, larger image file directories, and additional custom structures or information. BigTIFF reuses the same structures from TIFF; conceptually, they look the same, as

shown in Figure 3-8.

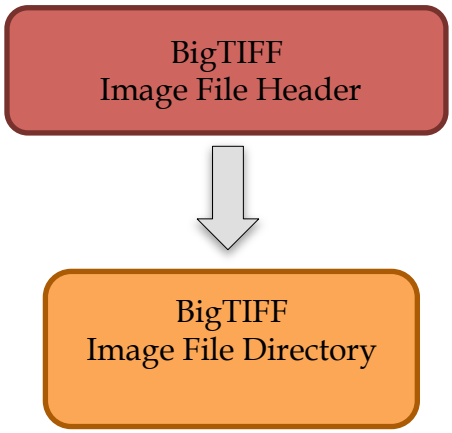

Figure 3-8 Conceptual BigTIFF Structure

3.3.1 BigTIFF Header

As shown in Figure 3-8, the BigTIFF header deviates from the traditional TIFF Header and is sixteen bytes long. The byte order field remains the same as the TIFF image file header for both size and function. The signature field should contain the value 43 (0x4b) and serves as the magic number for the file format. The offset size is a two-byte field that should always evaluate to 8 (0x08). If there is a TIFF format in the future that uses larger offsets, then this field will contain the size of those offsets; because BigTIFF uses eight-byte offsets the value is eight. The two-byte Reserved Bytes field should be set to 0. This allows for the file to be aligned on eight-byte offsets. The image file directory offset is eight bytes long and, like TIFF, points to the location of the IFD relative to 0x0.

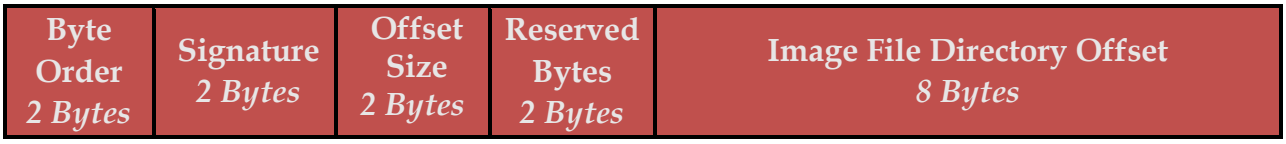

Figure 3-9 BigTIFF Header Structure

3.3.2 BigTIFF Image File Directory

The BigTIFF image file directory retains the same structure and functions as the baseline TIFF image file directory (see Image File Directory), but the sizes are different between the two. Figure 3-10, shown below, is a visual representation of the BigTIFF image file directory and contains the sizes of each part.

Figure 3-10 BigTIFF Image File Directory Structure

3.3.2.1 BigTIFF Tags

As shown in Figure 3-11, BigTIFF tags are similar to baseline TIFF tags. However, the number of values is an eight-byte field. The value offset is also an eight-byte field of type TIFF_LONG8 or a valid field type that can fit in the offset field.

Figure 3-11 BigTIFF Tag Structure

3.3.2.1.1 BigTIFF Field Types

BigTIFF defines three other field types (listed in Table 3-3), although acceptable field types also include baseline TIFF field types (listed in Table 3-1). These new field types allow for values that are eight bytes long, where the baseline TIFF field types have a maximum value of four bytes.

Table 3-3 BigTIFF Field Types

Type by Value	Size
16= TIFF_LONG8	8 byte unsigned integer.
17= TIFF_SLONG8	8 byte signed integer
18 = TIFF_IFD8	8 byte unsigned IFD offset

3.4 TIFF Electronic Photography (TIFF/EP)

Tagged Image File Format/Electronic Photography (TIFF/EP) is a variation of TIFF used for raw images formats; it does not hold wide-spread support. Note that this variation is structurally the same to the Baseline TIFF file structure, but requires certain tags to be found in the first IFD. Table 3-4 lists the mandatory tags.

Table 3-4 Mandatory TIFF/EP Tags[2]

TIFF/EP Tag#	Tag Name
014A	SubIFDs
015B	JPEGTables
828D	CFARepeatPatternDim
828F	BatteryLevel
83BB	IPTC/NAA
8773	InterColorProfile
8829	Interlace
882A	TimeZoneOffset
882B	SelfTimerMode
920D	Noise
9211	ImageNumber
9212	SecurityClassification
9213	ImageHistory
9216	TIFF/EPStandardID

3.5 Camera Raw

Camera RAW refers to a series of file formats that contain the values that camera sensors captured. The information required to process a Camera Raw image including to convert it another format like PNG or JPEG is usually proprietary to the camera manufacturer. Therefore, there is no standard "RAW" file format, but this term describes a wide swath of formats that have a common goal.

A camera may use a charge coupled device (CCD) or complementary metal-oxide semiconductor (CMOS) in order to capture in image, but in both cases the values that are recorded are light intensity values captured by the sensor. This information is then translated into color images using a color filter. RAW images generally require some

[2] Source: ISO/DIS 12234-2 - Photography — Electronic still picture imaging — Removable memory — Part 2: Image data format — TIFF/EP (PDF), 1998

convertor, either from the camera manufacturer or a third party, in order to process the image.

RAW files generally contain metadata about the image and the information necessary to perform "demosaicing[3]." This information enables a renderer to infer the color of pixel based on the values in the neighbor pixels. Moreover, RAW conversion involves: white balancing, colorimetric interpretation, gamma correction, lens distortion correction, and noise reduction.

There is no standard RAW format; camera manufacturers have their own proprietary file formats that may encrypt information pertinent to recovering the color information in a RAW file. Although these are proprietary, many are based on either TIFF or Camera Image File Format (CIFF), so there is a basic understanding of the structure of RAW image. Adobe created the Digital Negative (DNG) file format as a means to have an open-source, non-proprietary standard for RAW files, however it was met with little to no adoption. Major camera manufacturers continue to use formats that are not openly documented and require a convertor or the camera manufacturer's software.

There is a distinct risk in allowing RAW file formats because the proprietary information can be encrypted or found in non-standard tags; thus, permissive systems may be susceptible to more risk if private tags, sub IFDs or encrypted data are not inspected.

Table 3-4, below, summarizes analysis of a series of raw file formats and their similarity to known file formats. The exception count tracks how many private tags are not identifiable. Note that the TIFF 6.0 derived file formats may "augment" the original specification and include fields

[3] https://courses.cs.washington.edu/courses/cse467/08au/pdfs/lectures/09-Demosaicing.pdf

Table 3-5 Raw Formats Used by Cameras[4]

Maker	Model	Extension	Format	TIFF Tags Used	Number of IFDs	Exceptions
Canon	1Ds Mk II	CR2	TIFF 6.0	33	4	4
Canon	1Ds	TIF	TIFF 6.0	32	2	0
Canon	20D	CR2	TIFF 6.0	32	4	3
Canon	D30	CRW	CIFF	0	0	-
Canon	D60	CRW	CIFF	0	0	-
Fuji	S2	RAF	Fuji	19	2	0
Fuji	S3	RAF	Fuji	23	2	2
Kodak	SLR/n	DCR	TIFF/EP	86	3	9
Leaf	Aptus	MOS	TIFF 6.0	19	1	2
Leaf	Valeo	MOS	TIFF 6.0	18	1	1
Leica	Digilux 2	RAW	TIFF V85	34	1	25
Leica	AGR9	DNG	TIFF 6.0	44	2	1
Nikon	D100	NEF	TIFF/EP	41	2	1
Nikon	D1X	NEF	TIFF/EP	47	2	0
Nikon	D2H	NEF	TIFF/EP	41	2	1
Nikon	D2X	NEF	TIFF/EP	50	3	0
Nikon	D70	NEF	TIFF/EP	49	3	0
Nikon	D70S	NEF	TIFF/EP	50	3	0
Sigma	SD10	XCF	Foveon			-
Sony	F828	SRF	TIFF 6.0	24	2	0
Olympus	E300	ORF	TIFF ??	27	2	1
Adobe	D1X	DNG	TIFF 6.0	80	3	0

The MakerNote tag, 0x927c, is where manufacturer specific information is designated to be placed. This tag is found in the EXIF IFD.

[4] http://www.rags-int-inc.com/PhotoTechStuff/RawStandards/RawSummary.html

3.4.1 Deviation of NEF and CR2 from TIFF Baseline

Nikon Electronic File (NEF) and Cannon Raw Version 2 (CR2) all reuse the TIFF Baseline structures. However, both augment the file format such that a Baseline TIFF renderer should not be able to recreate the original image. NEF uses the Makernote tag to accomplish this, but preserves the TIFF structure.

CR2 has modified the file format in a way that the RAW picture is not accessible to a Baseline TIFF application [8]. For example, the header contains 4 additional fields: CR2 magic word (2 bytes), CR2 major version (1 byte), CR2 minor version (1 byte), and a RAW IFD offset (4 bytes). The RAW IFD contains the RAW image data generated by the camera; it would be treated as extraneous data by non-Canon applications.

4. TIFF CONSTRUCTS

This section discusses specific features and risks of the TIFF file format. Each construct provides a description, areas of concern, examples, and recommendations for potentially mitigating the risks.

4.1 File Size

OVERVIEW

TIFF files use four-byte (32 bit) offsets. Consequently, the maximum file size is 2^32 bytes or four gigabytes. The BigTIFF file format defines eight-byte (64 bit) offsets. The maximum file size for BigTIFF is 2^64 bytes or sixteen exabytes.

If the size of the file exceeds these limits, then it may be the source of a data attack or hiding risk. Information past the maximum length may go unprocessed by many image-rendering applications, which could provide an opportunity for data hiding. Files that are too large may be an attempt at a denial of service or buffer overflow attack.

Figure 4-1 shows malicious JavaScript embedded in a TIFF file at invalid offsets. The locations, depicted by the hex addresses in the light blue column, are five bytes long; the TIFF file uses four byte offsets. This information would go unreferenced and unprocessed by some applications. If the data is unprocessed and is placed in memory, then a malicious payload could be placed in this region and executed.

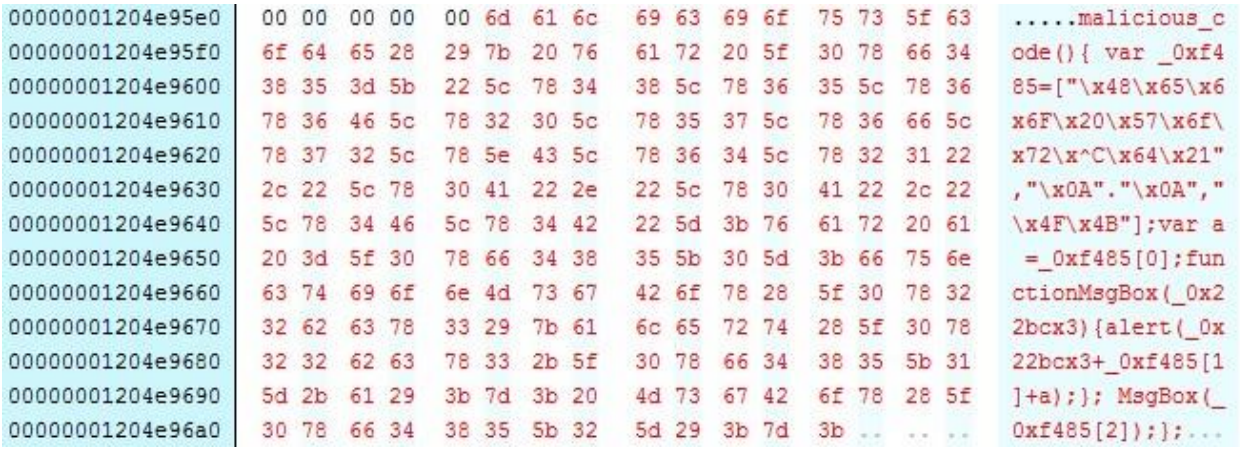

Figure 4-1 TIFF File with Javascript®[5]

RISKS AND RECOMMENDATIONS

Data Hiding & Data Attack – Data past the maximum file size will not affect images in files but has the potential to contain sensitive data. Additionally, data past the maximum file size can be a vehicle for malicious code. Files that exceed the allowed size may be a source of an attack risk (i.e., malicious payload) because of the in ability of reader to handle the out of specification file.

1. Validate – Validate that the file does not exceed the file size per the specification.
2. Validate – Validate that the file meets the minimum requirements for a TIFF file; the size should be equivalent or greater than the sum of the image file header, image file directory, and the product of the image height, image width, and sample size of the image.
3. Remove – Remove all data past the maximum file size from the file. This should not impact file functionality.
4. Reject – Reject files that exceed the maximum file size.

PRODUCT

TIFF Revision 6.0; BigTIFF Design documented by LibTIFF

LOCATION

The file size is a number denoting the summation of all the bytes within the file.

[5] Javascript is a registered trademark of Oracle Corp.

4.2 Image File Header

OVERVIEW

The TIFF image header is the first eight bytes in a TIFF file and is composed of three fields. The first two bytes designate the byte order used in the rest of the file; acceptable values are limited to 'II' in American Standard Code for Information Interchange (ASCII) or 0x4949 in hexadecimal and 'MM' in ASCII or 0x4d4d in hexadecimal. (Note: processing the file as a stream may require bits to be flipped based on the architecture of the system and the endianness of the TIFF file; otherwise the values read will be reversed.) The next two bytes identify the data as a TIFF file. The value of this field should always evaluate to the decimal value 42, but is dependent on the byte order designated in the first two bytes. This means it should either be 0x002A or 0x2A00. The final four bytes indicate the offset for the location of the first IFD. Although this field may point to any location within the file, it must begin on a word boundary (32 bits).

```
00000000 49 49 2a 00 08 00 00 00 54 68 69 73 20 69 73 20 II*.....This is
00000010 77 68 65 72 65 20 74 68 65 20 49 46 44 20 77 69 where the IFD wi
00000020 6c 6c 20 73 74 61 72 74 20 62 65 66 6f 72 65 20 ll start before
00000030 74 68 65 20 66 69 6c 65 20 68 61 73 20 62 65 65 the file has bee
00000040 6e 20 6d 6f 64 69 66 69 65 64 2e 2e 2e 2e 2e 2e n modified......
```

Figure 4-2 Sample Image Header

Figure 4-2 above depicts a typical image header and highlights the original IFD offset. The example in Figure 4-3 shows a version of the image header where IFD offset has been modified to insert extraneous data between the offset value and the start of the data specified by the offset (between the file header and the IFD).

```
00000000 49 49 2a 00 13 00 00 00 65 78 74 72 61 20 64 61 II*.....extra da
00000010 74 61 21 54 68 69 73 20 69 73 20 77 68 65 72 65 ta!This is where
00000020 20 74 68 65 20 49 46 44 20 77 69 6c 6c 20 73 74  the IFD will st
00000030 61 72 74 20 61 66 74 65 72 20 74 68 65 20 66 69 art after the fi
00000040 6c 65 20 68 61 73 20 62 65 65 6e 20 6d 6f 64 69 le has been modi
00000050 66 69 65 64 2e 00 00 00 00 00 00 00 00 00 00 00 fied............
```

Figure 4-3 Modified Image Header

The BigTIFF extension redefines the Image File Header as a sixteen-byte structure composed of the following fields:

- The first two bytes designate the Byte Order like a normal TIFF image.
- Bytes 2-3 must evaluate to 43; 0x002B or 0x2B00.
- Bytes 4-5 describe the size of the offsets; it must evaluate to 8.
- Bytes 6-7 should always evaluate to 0.
- The next eight bytes indicate the offset for the location of the first image file directory (IFD). This offset must evaluate to a location for a valid image file directory that begins on a word boundary.

RISKS AND RECOMMENDATIONS

Data Hiding – Files that do not conform to the specifications indicate the file may have been modified or created with an invalid application. Invalid offsets may indicate that the file was specially crafted and not generated by the intended application.

1. Validate – Validate the first two bytes of the file are either 0x4949 or 0x4d4d.
2. Validate – Validate that bytes 2-3 of the file evaluate to 42 or 43.
3. Validate – Validate the four-byte or eight-byte file offset (depending on the signature) begins on a word boundary.
4. Validate – Validate the file offset points to a valid IFD (see Section 4.3).
5. Replace – Replace the file offset with the value 0x8 so the IFD follows the image file header. This requires moving the IFD to this location.
6. Reject – Reject the file if the file header does not conform to specification requirements.
7. Reject – Reject files that have invalid IFD offsets. Invalid offsets point to locations already used for other data structures, e.g., the IFD offset pointing to the image file header.

PRODUCT

TIFF Revision 6.0; BigTIFF Design documented by LibTIFF

LOCATION

The Image File Header will make up the first eight bytes, for TIFF, or sixteen bytes, for BigTIFF, of every file.

4.3 Image File Directory

The constructs in this section will address the overall format of the Image File Directory and address specific concerns that pose data hiding, data attack, and/or data disclosure risks.

4.3.1 Structure

OVERVIEW

The TIFF IFD specifies the number of entries, also known as tags, followed by a series of twelve byte directory entries. Its location is dependent on the IFD offset indicated in the image file header.

- The first two bytes designate the number of entries, n.
- The next n*12 bytes contain n twelve-byte entries known as TIFF Tags

- The final two bytes contain the offset to the next directory or 0. The

BigTIFF extension redefines the IFD as such:

- The first eight bytes designate the number of entries, n.
- The next n*20 bytes contain n twenty-byte entries known as TIFF Tags.
- The final eight bytes contain the offset to the next directory or 0.

Image rendering applications do not have to process any information past the first IFD; as such, subsequent directories would be ignored by the renderer. Information past the first IFD can contain sensitive information that may not be rendered or displayed in the intended application; it can be effectively hidden from end users and poses a data hiding risk.

RISKS AND RECOMMENDATIONS

Data Attack – Files with an invalid structure may indicate that the file was specially crafted to exploit a system. Because TIFF applications are instructed by the specifications to follow all offsets, these fields are often loaded into memory and may cause malicious behavior.

1. Validate – Validate the number of entries is a valid number, e.g. less than or equal to a predefined number of entries.
2. Validate – Validate the offset to the next IFD evaluates to 0x0 or points to a valid location. An example of a valid location is a location that has yet to be referenced.
3. Validate – Validate that the IFD chain ends with an offset that points to 0x0.
4. Reject – Reject files that contain malformed IFDs.

Data Disclosure & Data Hiding– Files with multiple IFDs will have content that cannot be viewed in some image rendering applications. This information may be sensitive in nature.

1. Replace – Replace files with multiple IFD with a single flattened version.
2. Replace – Replace files with multiple IFD with a series of individual files containing only a single IFD from the original.
3. Review– Review all IFDs independently from other directories in a separate valid TIFF file
4. External Filtering Required – External filtering is required for all IFDs past the first one.
5. Reject – Reject files that contain multiple IFDs.

PRODUCT

TIFF Revision 6.0; BigTIFF Design documented by AWare Systems; TIFF Technical Notes Supplement 2

LOCATION

The IFD can be found anywhere within the file structure except overlapping existing structures.

4.3.2 Minimum Tags

Each image classification has a minimum set of tags. The absence of these tags indicates an invalid or malformed file. The following table enumerates the base set of necessary tags per image type; this can serve as a whitelist per image type.

Table 4-1 Minimum Set of Tags per Image Type

Image Classification	Tag Name	Tag Value (Hexadecimal)
Bi-Level	ImageWidth	0x100
	ImageLength	0x101
	Compression	0x103
	PhotometricInterpretation	0x106
	StripOffsets	0x111
	RowsPerStrip	0x116
	StripByteCounts	0x117
	XResolution	0x282
	YResolution	0x283
	ResolutionUnit	0x296
Grayscale	ImageWidth	0x100
	ImageLength	0x101
	BitsPerSample	0x102
	Compression	0x103
	PhotometricInterpretation	0x106
	StripOffsets	0x111
	RowsPerStrip	0x116
	StripByteCounts	0x117
	XResolution	0x11A
	YResolution	0x11B
	ResolutionUnit	0x128

Palette Color Images	ImageWidth	0x100
	ImageLength	0x101
	BitsPerSample	0x102
	Compression	0x103
	PhotometricInterpretation	0x106
	StripOffsets	0x111
	RowsPerStrip	0x116
	StripByteCounts	0x117
	XResolution	0x11A
	YResolution	0x11B
	ResolutionUnit	0x128
	ColorMap	0x140
RGB	ImageWidth	0x100
	ImageLength	0x101
	BitsPerSample	0x102
	Compression	0x103
	PhotometricInterpretation	0x106
	StripOffsets	0x111
	SamplesPerPixel	0x115
	RowsPerStrip	0x116
	StripByteCounts	0x117
	XResolution	0x11A
	YResolution	0x11B
	ResolutionUnit	0x128

RISKS AND RECOMMENDATIONS

Data Attack – Files with unknown tags may indicate that the file was specially crafted to exploit a system. Tags that are not recognized may contain malicious code instead of data pertinent to the file. Because TIFF applications are instructed by the specifications to ignore tags they do not know, these fields are often not checked.

1. Validate – Validate that the tags adhere to a whitelist of acceptable values. Tags are documented in the TIFF specifications and other application specific documentation.
2. Replace – Replace the IFD with a version that only contains the minimum required tags.
3. Reject – Reject files with IFDs that contain more than the minimum set of required tags.

Data Disclosure & Data Hiding– Files with multiple IFDs will have content that cannot be viewed in some image rendering applications. This information may be sensitive in nature.

1. Validate – Validate the required set of tags are present in the IFD.
2. Validate – Validate the tags adhere to a whitelist of acceptable values.
3. Replace – Replace the IFD with a version that only contains the minimum required tags.

PRODUCT

TIFF Revision 6.0; BigTIFF Design documented by AWare Systems; TIFF Technical Notes Supplement 2

LOCATION

The IFD or directories can be found anywhere within the file structure except overlapping existing structures.

4.3.3 Tag Order

The IFD must contain tag values in ascending order. The presence of tags that are out of order may indicate a malformed or crafted file. Unordered tags could trigger a stack buffer overflow because tags can be dependent on other tags in order to be processed. If the tags are out of order, then some applications may expend more resources in order to backtrack and find the necessary information.

RISKS AND RECOMMENDATIONS

Data Attack – Files with unordered tags may indicate that the file was specially crafted to exploit a system. By having unordered tags that create cycles, one could trigger a buffer overflow; some tags are dependent on other tags to be processed.

1. Validate – Validate tags are in the IFD are in ascending tag value.
2. Replace – Replace the IFD with a version that contains tags in ascending order.

PRODUCT

TIFF Revision 6.0; BigTIFF Design documented by AWare Systems; TIFF Technical Notes Supplement 2

LOCATION

The IFD or directories can be found anywhere within the file structure except overlapping existing structures.

4.4 TIFF Tags/Image File Directory Entries

The constructs in this section will address the overall format of the TIFF tag and address specific tags from the specifications that pose data hiding, data attack, and/or data disclosure risks.

Note: The ClipPath tag is not part of the Baseline TIFF tag list, but may pose a data risk for applications that process it. It is covered in Section 4.4.4.

4.4.1 TIFF Tag Format

OVERVIEW

The TIFF IFD entry (i.e., tag) specifies information relevant to the image. This includes both processing information and metadata.

- The first two bytes identify the tag.
- The next two bytes indicate the field type (See Table 3-1).
- Bytes 4-8 contain the number of values for the tag. For example, a tag like StripOffsets can specify the offsets where image strips are located and it would need a value in Bytes 4-8 specifying how many offsets there are.
- Bytes 8-12 contain the tag data or the offset where the data for the tag is found.

The BigTIFF extension redefines the IFD Entry as:

- The first two bytes identify the tag.
- The next two bytes indicate the field type. See Table 3-3 BigTIFF Field Types.
- Bytes 5-12 contain the number of values for the tag.
- Bytes 13-20 contain the tag data or the offset where the data for the tag is found.

All TIFF renderers should be able to understand and process the tags in Table 4-2, but they do not need to be in every TIFF file. Applications are not required to utilize the information provided by tags that are not enumerated in Table 4-2. Tags not defined in any specification serve as a potential data hiding vector. The table does not include hexadecimal or decimal values, but these can be found in the Appendix of the TIFF specification Revision 6.

Table 4-2 Baseline Support for TIFF Files

Tag Name	
NewSubfileType	StripByteCounts
SubfileType (Deprecated)	MinSampleValue
ImageWidth	MaxSampleValue
ImageLength	XResolution
BitsPerSample	YResolution
Compression	PlanarConfiguration
PhotometricInterpretation	FreeOffsets
Threshholding	FreeByteCounts
CellWidth	GrayResponseUnit
CellLength	GrayResponseCurve
FillOrder	ResolutionUnit
ImageDescription	Software
Make	DateTime
Model	Artist
StripOffsets	HostComputer
Orientation	ColorMap
SamplesPerPixel	ExtraSamples
RowsPerStrip	Copyright

Some applications may ignore parts of the TIFF tags, especially modifications to the type of values or upper bytes of known values that do not use the whole four bytes. This serves as a potential data hiding vector because these values do not have to be shown to end users.

RISKS AND RECOMMENDATIONS

Data Attack, Data Hiding, and Data Disclosure – Files with unknown tags or known tags with incorrect values for type/number of values can contain information that may not be presented to an end user or cause malicious behavior in applications. Tags that use values not found in the specifications may cause buffer overflows. Because the TIFF specifications recommend that unknown tags be ignored, unknown tags can serve as vectors for data hiding or data disclosure, e.g. custom metadata.

1. Validate – Validate the TIFF Tag ID, bytes 1-2, match a known tag defined in the Baseline Specifications or GeoTIFF Specifications.
2. Validate – Validate that the TIFF tag is on a whitelist of known good tags.
3. Validate –Validate that the field type, bytes 3-4, is a known, acceptable value for the tag.
4. Validate – Validate the number of values, bytes 5-8 or bytes 5-12, do not exceed a prescribed value in the specifications.
5. Validate – Validate the tag data or offset to the data, bytes 9-12 or bytes 13-20, contains well-formed data for the tag according to the specifications, e.g. a data type of RATIONAL contains two LONG values.
6. External Filtering Required – External filtering is required for tags with a field type, bytes 3-4, of ASCII because they can contain large volumes of text that may be unrelated to the image.
7. External Filtering Required – External filtering is required for tags that match a whitelist of another specification (e.g., EXIF tags should be sent to an appropriate filter).
8. Remove- Remove all tags and metadata that are not required by Baseline TIFF renders to be processed. This may compromise image quality in certain images.
9. Reject – Reject files that contain unknown tags.

PRODUCT

TIFF Revision 6.0; BigTIFF Design documented by AWare Systems; GeoTIFF Format Specification Revision 1.0

LOCATION

TIFF tags are found in IFDs.

4.4.2 NewSubFileType/SubFileType

OVERVIEW

The NewSubFileType tag, 0xFE, replaces the SubFileType tag, 0xFF. The presence of both of these tags is redundant, as only the first one encountered will be processed. This serves as a potential data hiding vector because the second tag would be ignored.

The values for the NewSubFileType are limited to the first three bits; the value should never exceed 0x7. Each bit acts as a flag and signifies to an image rendering application a different classification for the image. If all the flags are unset then it is the only image in the IFD and is immediately render-able.

If Bit 0 is set to 1, then the image for the IFD is a reduced-resolution version of another IFD in the same file. This can be used for thumbnails or pyramid layering.

If Bit 1 is set to 1, then the image is one page in a multi-page image.

If Bit 2 is set to 1, then the image in the IFD defines a transparency mask for another image. The PhotometricInterpretation tag for the IFD must be set to 4 for the transparency mask to work.

Many applications adhere to only a single flag being set at a time; therefore, acceptable values are limited to 0x0, 0x1, 0x2, and 0x4. Many applications will not render the image in an IFD if the values are set to incorrect values and will instead opt to skip over the IFD and go to the next (if it exists). This serves as a data hiding vector because the image data would still exist despite not being rendered.

A transparency mask will modify what parts of another layer are transparent when being rendered. This can be used to control what the user will see when viewing the image even though the entire image still exists in the file. This can serve as a disclosure risk if parts of an image are meant to be hidden because this information still exists within the file. It can also be a data hiding risk because information can be obfuscated using the transparency mask.

The images in Figure 4-5 show satellite imagery with a layer mask applied (left) and the mask removed (right). The transparent pixels would render as black pixels in some applications or may not be rendered at all.

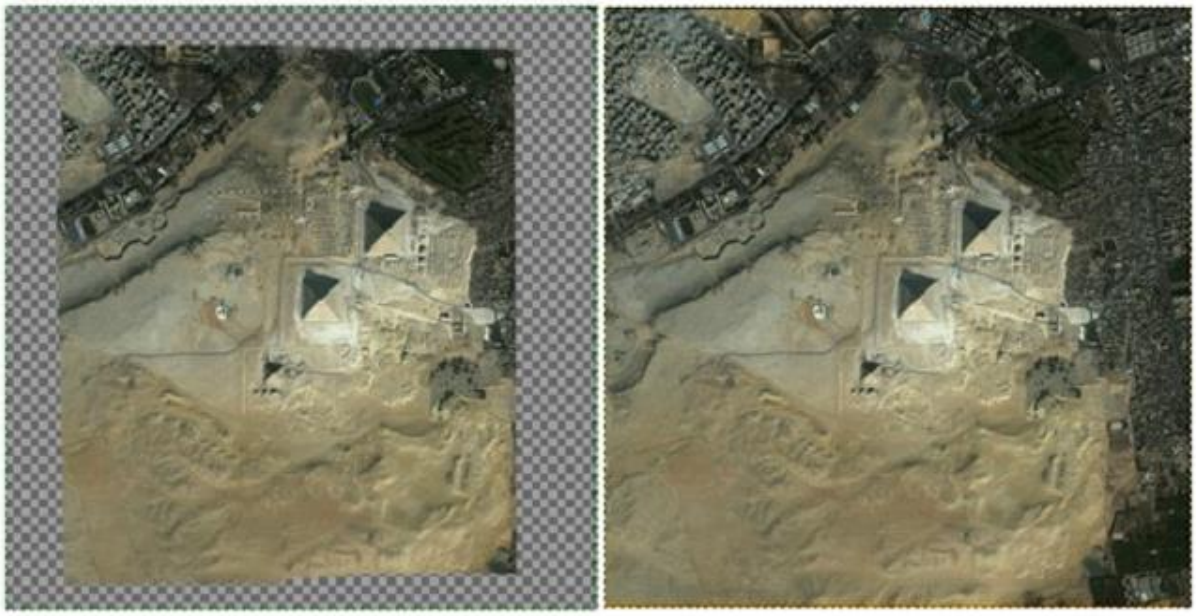

Figure 4-4 Satellite Imagery with and without a Transparency Mask

RISKS AND RECOMMENDATIONS

Data Hiding & Data Disclosure – Use of a transparency mask allows for the obfuscation of information within a file. The original image exists in its entirety; if the mask were removed, then the original data could be recovered. Also, the areas that are obfuscated may contain sensitive information underneath the transparency mask.

1. Validate – Validate NewSubFileType and SubFileType tags do not exist in the same IFD.
2. Validate – Validate NewSubFileType is set to an acceptable value (0x0, 0x1, 0x2, and 0x4).
3. Remove – Remove an IFD that has a NewSubFileType tag set to 0x4 indicating the contained image is a transparency mask. This would remove the mask and not the image.
4. Remove – Remove an IFD with an invalid NewSubFileType value.
5. Reject – Reject files with the presence of both the NewSubFileType and SubFileType tag.
6. Reject – Reject files containing transparency masks.
7. Reject – Reject files containing an invalid NewSubFileType value, e.g. anything greater than or equal to 0x5.

PRODUCT

TIFF Revision 6.0; BigTIFF Design documented by AWare Systems; GeoTIFF Format Specification Revision 1.0

LOCATION

TIFF tags or IFD Entries are found in IFDs.

4.4.3 Compression

OVERVIEW

When the value of the compression tag is 6 or 7, then the compression type for image stored in the IFD is JPEG. A value of 6 indicates an older style of storing JPEG information (defined in the TIFF Specifications Revision 6) and 7 indicates a newer style (defined in the TIFF Specification Supplement 2). The table below enumerates the different methods of compression and the corresponding value that should be found in the compression tag.

Table 4-3 Documented Compression Types and Values

Compression Type	Decimal Value
No Compression	1
CCITT modified Huffman RLE	2
PackBits	32773
CCITT Group 3 fax encoding	3
CCITTT Group 4 fax encoding	4
LZW	5
Old Style JPEG	6
JPEG	7

The respective specifications describe how the files are compressed and the additional tags necessary to process the file as JPEG.

RISKS AND RECOMMENDATIONS

Data Attack – As JPEG data may not be understood by software designed to process

TIFF files, it cannot be sanitized until it has been decompressed. A malformed JPEG compressed TIFF file may trigger a buffer overflow[6] for applications that cannot properly process them.

1. Replace – Replace JPEG compressed data with another valid compression that can be processed.
2. External Filtering Required – External filtering is required for images with Compression values set to six or seven. The extraction process for a JPEG compressed image in a TIFF file may reduce the quality of the image.
3. Reject – Reject JPEG compressed TIFF files.

PRODUCT

TIFF Revision 6.0; TIFF Specification Supplement 2

LOCATION

The compression tag will be found in every IFD within a TIFF file.

[6] Reference :http://www.exploit-db.com/exploits/30011/

4.4.4 ClipPath

OVERVIEW

Clipping paths provide the ability to only display portions of an image without the need to add an additional layer. ClipPath is a tag that will contain a series of instructions, documented in TIFF Technical Notes Supplement 2. These instructions provide specific renderers the information necessary to display only a certain portion of an image.

ClipPath does not utilize another layer to generate the clipped image; the clipping instructions will be found in the same IFD. The images in Figure 4-5[7] depict the difference between a clipping path and a transparency mask. Image C and F are the final rendered images.

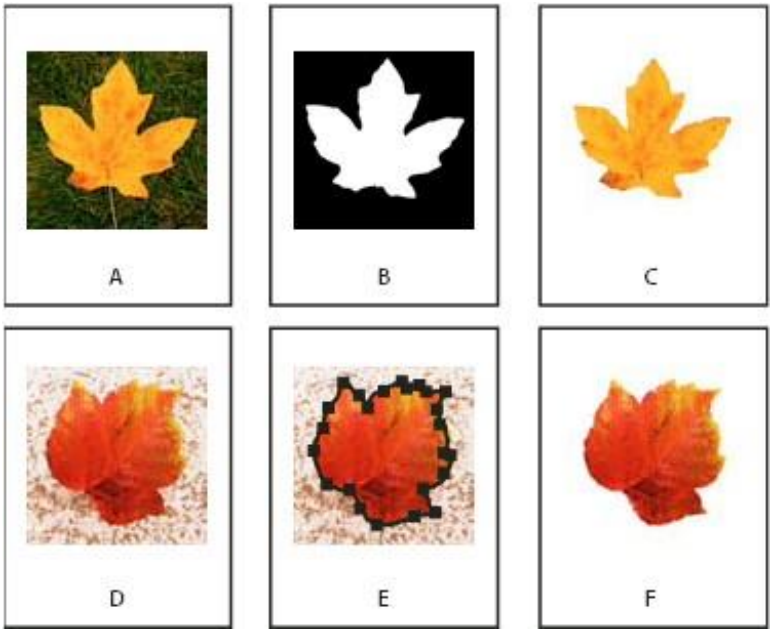

Figure 4-5 Transparency Mask (A, B, C) vs. a Clipping Path (D, E, F)

RISKS AND RECOMMENDATIONS

Data Hiding & Data Disclosure – Because the full contents of the file will not be displayed to an end user if there is a clipping path, the ClipPath tag poses a data hiding risk. Sensitive information can be placed in obfuscated regions.

1. Replace – Replace the ClipPath instructions with values that do not manipulate the image.
2. Remove – Remove the tag, in its entirety, from the file. This may alter how the image renders.
3. Reject – Reject files with the ClipPath tag.

[7] Source: http://help.adobe.com/en_US/indesign/cs/using/images/gh_08.png

PRODUCT

TIFF Revision 6.0; TIFF Specification Supplement 2

LOCATION

The ClipPath tag, if in the file format, will be found in the IFD

4.4.5 MakerNote

OVERVIEW

The MakerNote tag is designated to as a place holder for manufacturer specific information. It is typically found within the EXIF IFD, however it would be valid to have the tag in any IFD. The information in this tag is generally encrypted and can only be understood by the target software; this information may be pertinent to converting an image from camera raw.

RISKS AND RECOMMENDATIONS

Data Hiding – As the data contained in the MakerNote tag may not be understood by software designed to process it, then cannot be reliably sanitized.

1. Validate - If the format is DNG then inspection may be possible without licensing the Camera manufacturer's RAW format. Otherwise it is possible to license the RAW format information from the camera manufacturer.

2. Reject – Reject files with the MakerNote tag.

PRODUCT

TIFF Revision 6.0; TIFF Specification Supplement 2; DNG Specification

LOCATION

The MakerNote tag, if it exists, will usually be found in the EXIF IFD

4.4.6 ColorMap

OVERVIEW

The TIFF ColorMap tag provides the ability for images to use an RGB-lookup table for applications to render them. Modifying the color table to obfuscate information is possible by crafting a special color table that maps colors incorrectly. Figure 4-6 and Figure 4-7 demonstrate images where a white background is mapped to black in order to hide sensitive text. In the color table for the modified image, there are duplicate entries for black. This poses a data hiding risk because image data can be obfuscated. The examples below illustrate this data hiding vector.

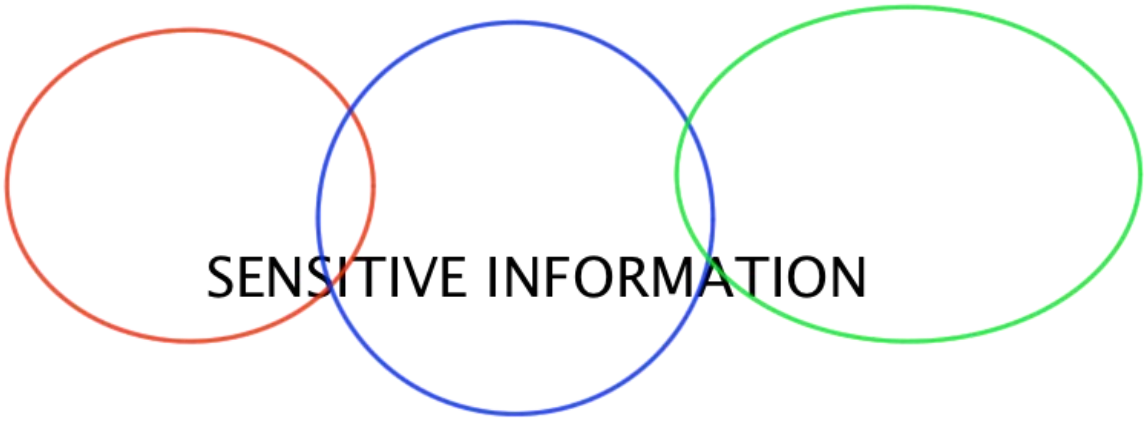

Figure 4-6 Image with Sensitive Information

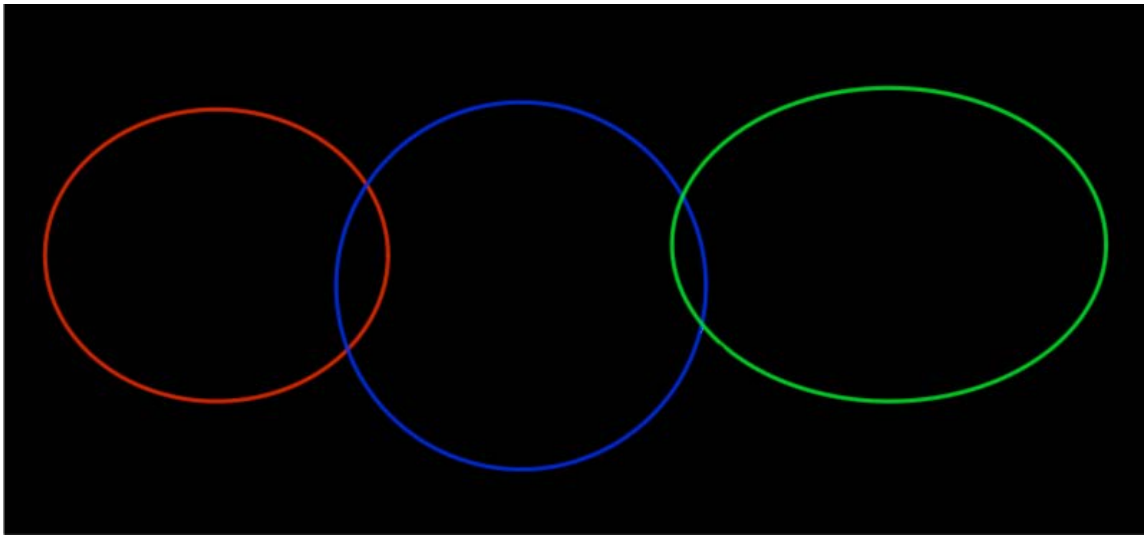

Figure 4-7 Image with Obfuscated Text

Because each color is indexed it is possible to modify how the image is rendered without directly modifying the image.

RISKS AND RECOMMENDATIONS

Data Hiding– Because image data can be obfuscated without modifying pixel data there is a data hiding risk.

1. Validate – Validate the values for ColorMap do not include duplicate colors.
2. Replace – Replace the image data with a format that does not require a color table. Note: This may alter the image content.
3. Replace – Replace duplicate colors in the ColorMap with a color not found in the table. Note: This may alter the image content.
4. Reject – Reject TIFF files containing the ColorMap tag.
5.

PRODUCT

TIFF Revision 6.0; TIFF Specification Supplement 2

LOCATION

The ColorMap tag, if in the file format, will be found in the IFD.

4.4.7 SubIFD

As of TIFF Technical Notes Supplement 1 [5], there exists a tag for sub image file directories (SubIFD). This creates a tree structure for TIFF as opposed to the linear one described. Each SubIFD must adhere to the same rules as a normal IFD. SubIFDs post the same level of risk as having multiple IFDs in a single TIFF file.

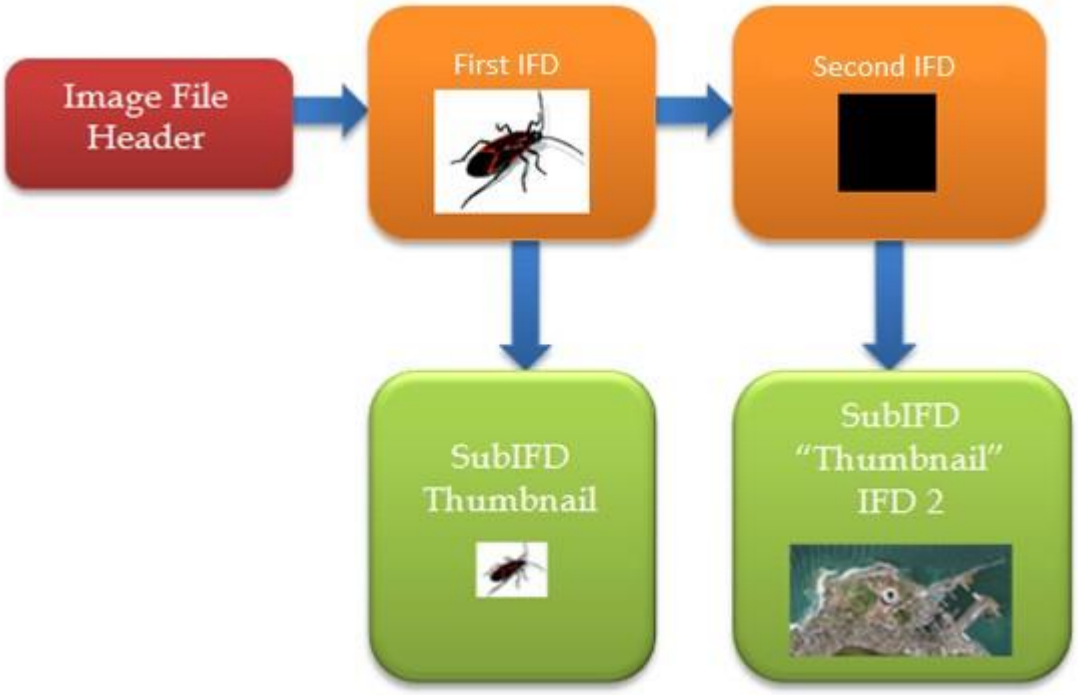

Figure 4-8 Data Hiding Vector using SubIFD

Figure 4-8 illustrates a potential data hiding vector using the SubIFD tag. By altering the thumbnail for the second image, you can store another image that is a larger size than its parent.

RISKS AND RECOMMENDATIONS

Data Attack – Files with SubIFD tags are susceptible to invalid offsets, unordered tags, and unknown tags. By having offsets that create cycles, one could trigger a stack buffer overflow. Additionally, unordered tags could create the same problem if the file was large enough; some tags are dependent on other tags to be processed. Tags that are not

recognized may contain malicious code instead of data pertinent to the file. Because TIFF applications are instructed by the specifications to ignore tags they do not know, these fields are often not checked.

1. Validate – Validate that the IFD chain and any SubIFD chain ends with an offset that points to 0x0.
2. External Filtering Required – External filtering is required for all SubIFDs. It can be passed in to the same filter that handles the IFD structure.

Data Disclosure & Data Hiding– Files with multiple IFDs will have content that cannot be viewed in some image rendering applications. This information may be sensitive in nature.

1. Review– Review all SubIFDs independently from other directories in a separate valid TIFF file
2. External Filtering Required – External filtering is required for all SubIFDs.
3. Remove – Remove all SubIFDs tags and associated information. This should have no bearing on the original image; however, it may reduce functionality in some applications (e.g., no thumbnails).

PRODUCT

TIFF Revision 6.0; TIFF Specification Supplement 2

LOCATION

The SubIFD tag, if in the file format, will be found in the IFD.

4.5 GeoTIFF Extension

The following subsections address potential data risks found in the GeoTIFF extension for TIFF files. The first three sections address structural issues, whereas the final section enumerates specific tags and key codes.

4.5.1 GeoTIFF Tags

OVERVIEW

GeoTIFF tags adhere to the TIFF tag format and should be validated in the same manner as the TIFF tag. Table 3-2 enumerates GeoTIFF specific tags defined in the specification.

GeoTIFF files require the GeoKeyDirectoryTag; if there are other GeoTIFF specific tags without the GeoKeyDirectoryTag, then they are invalid. These other tags may contain sensitive information that will not be processed by a GeoTIFF rendering application; therefore, it is a data disclosure risk.

Because the GeoTIFF tags adhere to the TIFF tag structure, they can use the data types that the TIFF file format defines. For example, the GeoAsciiParamsTag is of the ASCII data type. Files that do not use the prescribed data type in the specifications may not render, which make these tags a data hiding vector.

RISKS AND RECOMMENDATIONS

Data Hiding – Because tags do not have to be processed by TIFF rendering applications, they serve as a vector for carrying information that an intended viewer may not display.

1. Validate – Validate the presence of the GeoKeyDirectoryTag.
2. Validate – Validate the tags adhere to known sizes and data types, e.g.

   GeoDoubleParamsTag, which is designated to have data type DOUBLE, does not have the data type ASCII.
3. Remove – Remove all GeoTIFF tags if the GeoKeyDirectoryTag is not found.
4. External Filtering Required – External filtering is required for tags with the ASCII data type.

PRODUCT

GeoTIFF Format Specification Revision 1.0

LOCATION

GeoTIFF tags will be found in the IFD.

4.5.2 GeoTIFF Keys

OVERVIEW

Possible GeoTIFF keys are defined in the GeoTIFF Format Specification Revision 1.0. These keys may contain sensitive information; this section will address data risks in keys.

If the data type for the keys is ASCII or double, then the values should be stored in GeoASCIIParamsTag or GeoDoubleParamsTag, respectively. Applications may not be able to render these values if they do not conform to the specifications. This can be a data hiding vector.

Figure 4-9 shows a sample set of GeoKeys, starting below the field labeled GeoTIFF version. It shows the Geodetic Datum[8] used and the reference ellipsoid. This information, paired with the Model Tie Point tag (the values were redacted), can be used to process the image in the file to determine the location. The GeoKey labeled GT Citation also reveals how the imagery was obtained despite not being the intended

[8] http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/opt/csbgeo05.html

purpose; as an ASCII field, it can hold information related to the image but not pertinent to rendering the image.

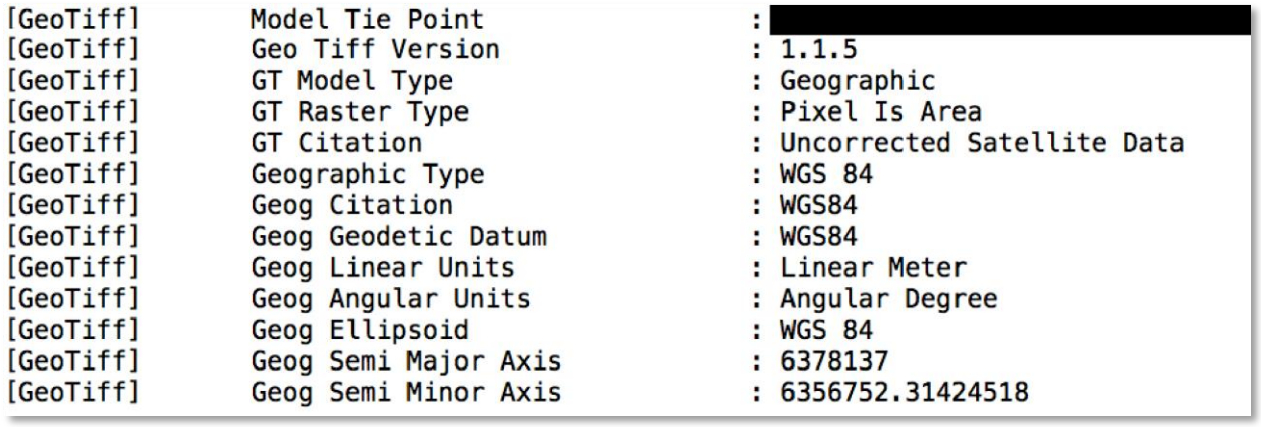

Figure 4-9 Data Disclosure in GeoKeys

RISKS AND RECOMMENDATIONS

Data Hiding & Data Disclosure – Because malformed keys do not have to be processed by TIFF rendering applications, they serve as a vector for carrying information that an intended viewer may not display. Additionally, GeoKeys can contain values and information related to where and how images were taken, which may be sensitive in nature.

1. Validate – Validate keys with defined values adhere to a whitelist of acceptable values.
2. Validate – Validate that keys adhere to known sizes and data types.
3. Validate – Validate all keys of the same data type store their information in the same tag.
4. Remove – Remove keys that do not store their values in an appropriate location, e.g., a key with value double storing information outside of GeoDoubleParamsTag.
5. External Filtering Required – External filtering is required for keys with the ASCII data type.

PRODUCT

GeoTIFF Format Specification Revision 1.0

LOCATION

GeoTIFF keys will be found in the IFD.

4.5.3 GeoTIFF Key Codes

OVERVIEW

Specific values for GeoKeys may relay sensitive information embedded in the geographic information (e.g., latitude and longitude information, distances, etc.). The values are enumerated in the GeoTIFF Specifications and fall under four broad categories.

- General Codes
- Geographic Coordinate System Codes
- Projected Coordinate System Codes
- Vertical Coordinate System Codes

RISKS AND RECOMMENDATIONS

Data Disclosure – The Key Codes may contain sensitive information related to the locations.

1. Validate – Validate that codes adhere to a whitelist of known, acceptable GeoTIFF Key Codes and values
2. Remove – Remove keys that contain unknown, invalid, or blacklisted codes.
3. Reject – Reject files containing unknown, invalid, or blacklisted codes.

PRODUCT

GeoTIFF Format Specification Revision 1.0

LOCATION

GeoTIFF key codes will be found in the IFD.

4.5.4 Geographic Information

GeoTIFF tags and keys identify datum, coordinate systems, and projections in TIFF images. Because this information discloses geographic information, it poses a data disclosure risk. In order to mitigate this risk, the geographic information included within the file format should be processed and checked against a whitelist of acceptable locations. The following tags are vectors that can disclose sensitive information when paired with an image. Each of these tags, if they exist, should be evaluated to see if the file discloses sensitive information.

4.5.4.1 ModelTiepoint Tag

OVERVIEW

The ModelTiepoint tag (0x8482) maps the raster space to the model space. The raster space is the raster pixel data of the image and the model space refers to the corresponding geographic, geocentric, or projected coordinate system. A geographic coordinate system is a three–dimensional reference system that locates points on the Earth's surface[9].

The ModelTiepoint tag enumerates a variable number of tiepoints that each contains two sets of coordinates. Tiepoints are points in a digital image or aerial photograph that represent the same location in an adjacent image or aerial photograph[10]. For GeoTIFF, these tiepoints map the raster space to the model space. The first set of coordinates (A,B,C) refers to a point in the raster space at (A,B) that has the pixel value C, and the second set of coordinates, (E, F, G), represents a vector in model space.

Tiepoints can be used to determine whether or not the content of the image is disclosing sensitive data. Because tiepoints can be exact affine transformations[11], the second set of coordinates provided can be processed to determine the location of the image. Processing these coordinates will require knowledge of the coordinate system used, which should be defined in the GeoKeyDirectory. After processing the information with the correct coordinate system (see Section 4.5.4.2), it is possible to make a determination on data disclosure.

RISKS AND RECOMMENDATIONS

Data Disclosure – The information enumerated in this tag may contain disclose sensitive geographic coordinates that provide context for the image in the file.

1. Validate – Validate that the tiepoints are within an acceptable range of values for known good coordinates, e.g. the tiepoints are all located within a pre-defined bounding box.
2. Reject – Reject files that contain coordinates that are not whitelisted.

PRODUCT

GeoTIFF Format Specification Revision 1.0

LOCATION

This tag will be found in TIFF Directory.

---

[9] Source: http://edndoc.esri.com/arcsde/9.1/general_topics/what_coord_sys.htm

[10] Source: http://support.esri.com/en/knowledgebase/GISDictionary/term/tie%20point

[11] Affine transformations: An affine transformation is any transformation that preserves collinearity (i.e., all points lying on a line initially still lie on a line after transformation) and ratios of distances (e.g., the midpoint of a line segment remains the midpoint after transformation) [6]

4.5.4.2 GeoKeyDirectory Tag

The GeoKeyDirectory tag (0x87AF) contains geokeys that are required to process GeoTIFF files. The values for these keys, known as geocodes, store location specific information about an image.

The GeoKeyDirectory Tag is composed of a variable number of geokeys that provide complete geographic information for the GeoTIFF file format. This information indicates to applications how to process the data provided and provides geographic context for the model space to which the raster pixels have been mapped.

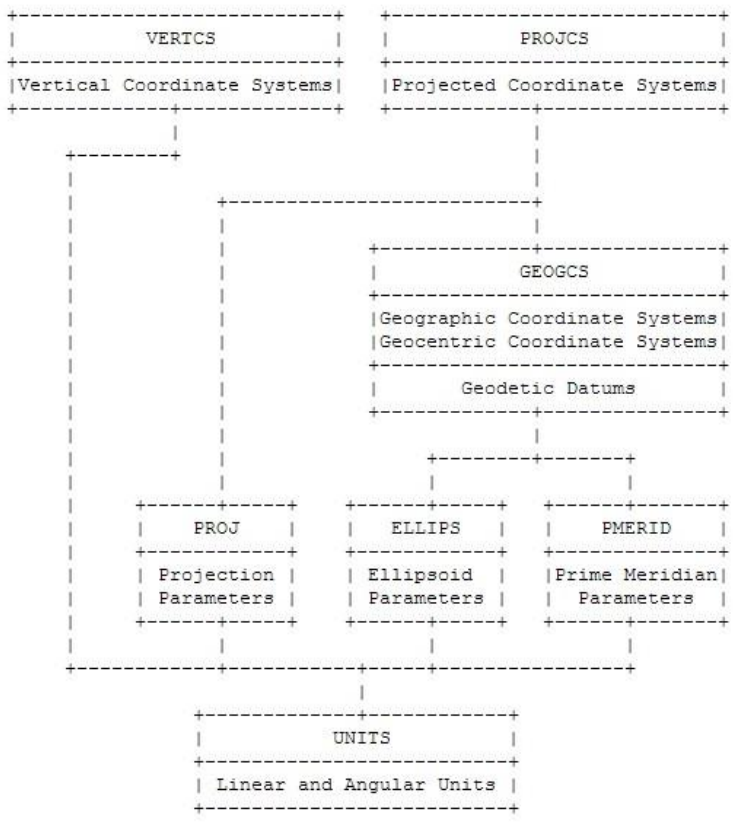

Figure 4-10 Data Flow for Coordinate Transformations[12]

Figure 4-10 shows the dataflow of various parameter datasets; the flow chart enumerates required values when selecting a specific coordinate transformation starting with a vertical or projected coordinate system. This diagram shows the information that needs to be processed in order to detect data disclosure.

The following subsections will enumerate keys that may disclose sensitive information.

[12] Source: http://www.remotesensing.org/geotiff/spec/geotiff2.6.html#2.6.2

4.5.4.2.1 GeoTIFF Configuration Keys

OVERVIEW

The GeoTIFF configuration keys establish the coordinate system of the image. The GTModelTypeGeoKey defines the type of model coordinate system used; this is model space to which the raster data will be mapped. The system selected references the model of the earth onto which image will be imposed; there are four types of coordinate systems: geographic, geocentric, projected and vertical. This information is necessary for processing tiepoints.

RISKS AND RECOMMENDATIONS

Although, there is no data risk in this field, it can be used to determine whether or not certain geographic information should pass, e.g. only geographic coordinate systems are allowed and vertical is not.

PRODUCT

GeoTIFF Format Specification Revision 1.0

LOCATION

This information will be found in the GeoKey Directory.

4.5.4.2.2 Geographic Coordinate System Parameter Keys

OVERVIEW

The GeographicTypeGeoKey specifies the code for the geographic coordinate system; this will map the latitude and longitude to a specific ellipsoid over the earth. This field may define a specific region, which can disclose sensitive information.

Figure 4-11 is a subset of possible geocodes for the key; some geocodes are named using locations.

```
GCS_Batavia =          4211
GCS_Barbados =         4212
GCS_Beduaram =         4213
GCS_Beijing_1954 =     4214
GCS_Belge_1950 =       4215
GCS_Bermuda_1957 =     4216
GCS_Bern_1898 =        4217
```

Figure 4-11 Subset of Geocodes for GeographicTypeGeoKey

RISKS AND RECOMMENDATIONS

Data Disclosure – The geocode used for this field may disclose sensitive location information based off the associated value name.

1. Validate – Validate the value at this field conforms to an acceptable, whitelisted value.
2. Reject – Reject files that contain values that are not found on a whitelist of known, acceptable values.

PRODUCT

GeoTIFF Format Specification Revision 1.0

LOCATION

This information will be found in the GeoKey Directory.

4.5.4.2.3 Projected CS Parameter Keys

OVERVIEW

ProjectedCSTypeGeoKey specifies the code for the projected coordinate system. This will map the data to a 2D surface that uses meters and feet for distance. The projection type used may define a specific region, which can disclose sensitive information.

Figure 4-12 shows a subset of possible geocodes for this key; many geocodes are named using locations.

```
PCS_NAD27_Kansas_North =     26777
PCS_NAD27_Kansas_South =     26778
PCS_NAD27_Kentucky_North =   26779
PCS_NAD27_Kentucky_South =   26780
PCS_NAD27_Louisiana_North =  26781
PCS_NAD27_Louisiana_South =  26782
PCS_NAD27_Maine_East =       26783
```

Figure 4-12 Subset of Geocodes for ProjectedCSTypeGeoKey

RISKS AND RECOMMENDATIONS

Data Disclosure – The geocode used for this field may disclose sensitive location information based off the associated value name.

1. Validate – Validate the value at this field conforms to an acceptable, whitelisted value.
2. Reject – Reject files that contain values that are not found on a whitelist of known, acceptable values.

PRODUCT

GeoTIFF Format Specification Revision 1.0

LOCATION

This information will be found in the GeoKey Directory.

4.5.4.3 Projection Definition Keys

OVERVIEW

ProjectionGeoKey specifies the coordinate transformation method and the projection zone parameters. This information paired with a Geographic System will form a Projected Coordinate System that can be used to process the coordinates enumerated in the ModelTiepoint tag.

Figure 4-13 shows a subset of possible geocodes for this key; many geocodes are named using locations.

```
Proj_Hawaii_CS27_4 = 15104
Proj_Hawaii_CS27_5 = 15105
Proj_Hawaii_CS83_1 = 15131
Proj_Hawaii_CS83_2 = 15132
Proj_Hawaii_CS83_3 = 15133
Proj_Hawaii_CS83_4 = 15134
Proj_Hawaii_CS83_5 = 15135
```

Figure 4-13 Geocodes for ProjectionGeoKey

RISKS AND RECOMMENDATIONS

Data Disclosure – The geocode used for this field may disclose sensitive location information based off the associated value name.

1. Validate – Validate the value at this field conforms to an acceptable, whitelisted value.
2. Reject – Reject files that contain values that are not found on a whitelist of known, acceptable values.

PRODUCT

GeoTIFF Format Specification Revision 1.0

LOCATION

This information will be found in the GeoKey Directory.

4.5.4.4 Geo_Metadata Tag

OVERVIEW

The private TIFF tag Geo_Metadata (0xC6DD) stores XML-based Geographic Markup Language (GML) at the offset specified by the tag. This information will have to be inspected separately to ensure it is free of any data risks, because it may contain attack, hiding, and disclosure vectors unique to markup languages and not found in TIFF. GeoTIFF does not require this tag, but it may be included in certain environments[13].

The Geo_Metadata tag is a private tag registered by the National GeospatialIntelligence Agency (NGA). This tag is designed to store GML metadata. The information contained within this markup language is constrained by schemas, but can include similar information found elsewhere in the file format. Figure 4-14 below shows the full specification for the tag.

IFD		Image File Directory
Code		50909 (hex 0x87AF)
Name		Geo_Metadata
Type		ASCII
Count		4-byte (max. size = 4GB)
Default		None

Figure 4-14 Geo_Metadata Tag Information

[13] Source: http://www.gwg.nga.mil/ntb/baseline/docs/nga.ip.0001_1.0-geotiff/NGA.IP.0001_2.0.pdf

Figure 4-15 shows a fragment of GML. It contains the latitude and longitude coordinates of a location.

```
<gmd:geographicElement>
    <gmd:EX_GeographicBoundingBox>
        <gmd:westBoundLongitude>
            <gco:Decimal>-90.4</gco:Decimal>
        </gmd:westBoundLongitude>
        <gmd:eastBoundLongitude>
            <gco:Decimal>-90.2</gco:Decimal>
        </gmd:eastBoundLongitude>
        <gmd:southBoundLatitude>
            <gco:Decimal>38.6</gco:Decimal>
        </gmd:southBoundLatitude>
        <gmd:northBoundLatitude>
            <gco:Decimal>38.8</gco:Decimal>
        </gmd:northBoundLatitude>
    </gmd:EX_GeographicBoundingBox>
</gmd:geographicElement>
```

Figure 4-15 GML Snippet

RISKS AND RECOMMENDATIONS

Data Attack – Because GML data may not be understood by software designed to process TIFF files, it should be processed and sanitized by software designed to process markup languages. A malformed or malicious GML file may pose a data attack risk for applications that attempt to process it.

1. Remove – Remove the Geo_Metadata tag and the GML from the file format.
2. External Filtering Required – External filtering is required for GML within the file format.
3. Reject – Reject files that contain the Geo_Metadata tag.

Data Disclosure – GML data by definition contains geographic information. This can be sensitive in nature and can disclose city names, longitude and latitude values, etc.

4. Validate – Validate the fields in the GML Metadata adhere to a whitelist of known, acceptable values.
5. External Filtering Required – External filtering is required for the fields in the GML metadata.
6. Reject – Reject files that have fields with values that do not conform to acceptable values.

PRODUCT

GeoTIFF Format Specification Revision 1.0

LOCATION

This tag will be found in the IFD.

4.6 Unreferenced Data

OVERVIEW

Some TIFF readers will ignore information without a corresponding offset. This includes, but is not limited to, data before the first IFD, data between IFDs, and data at the end of a TIFF file.

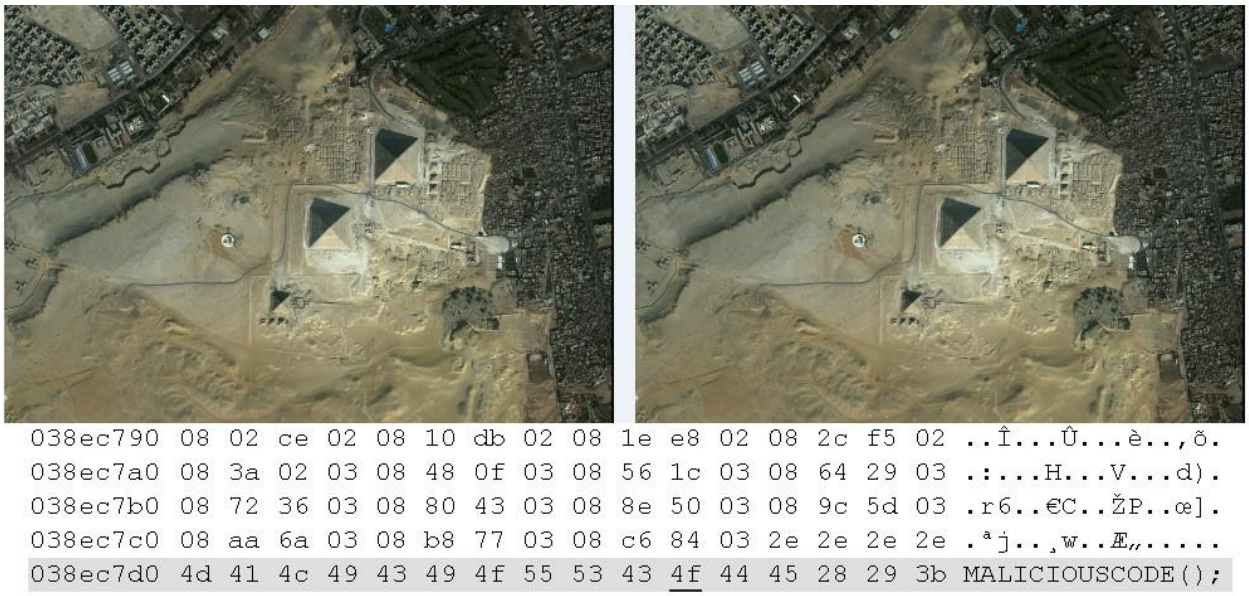

Figure 4-16 Information at the end of a TIFF file

Figure 4-16 above illustrates the ability to append information to the end of a file without affecting how the image is rendered. The left side shows the original image and the right side shows the modified image; the snippet of hex from the file shows how the image on the right was modified by adding additional text to the end of the file.

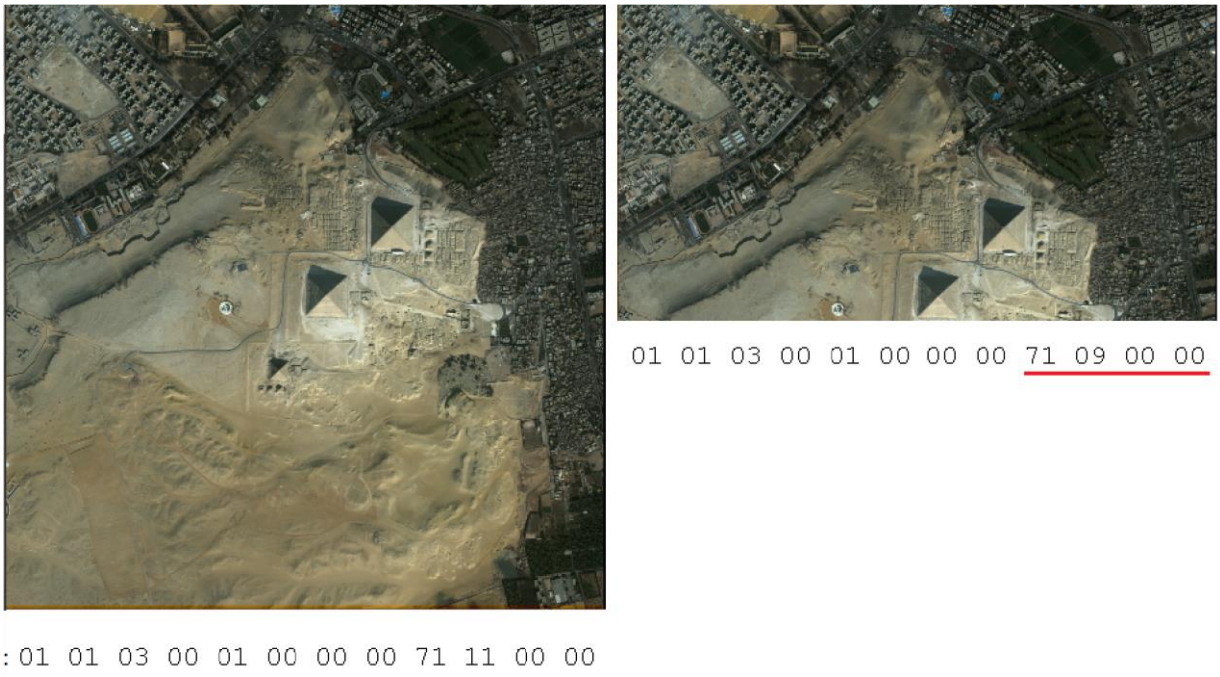

Figure 4-17 Modified ImageLength Tag

Figure 4-17 (above) demonstrates the use of the ImageLength tag to crop an image. The rest of the image data still exists within the file format and can be viewed by reverting the underlined value to the original size. The image data not rendered is considered unreferenced data because it may never be read by an image rendering application.

RISKS AND RECOMMENDATIONS

Data Hiding & Data Attack– Because information that does not fall into the tree structure of TIFF files is ignored, there is a data hiding risk and data attack risk. Information not referenced by an offset will not be processed by an intended application; it can exist in the file format and not affect how a file is processed. The information stored here can be used as a payload after an exploit is triggered.

1. Remove – Remove information that does not have an offset after the Image File Header. This approach would require modifying the offsets throughout the file to reference the new locations. The end result is structures that run in sequence with no space in between them. Note: In order to determine if there is data between offsets, the offsets and structure sizes need to be examined.
2. Remove – Remove image information that exceeds the given height and/or width. This may require the image data to be uncompressed.
3. Replace – Replace unreferenced information with a safe value, e.g. write out zeroes instead of the original values.

PRODUCT

TIFF Revision 6.0

LOCATION

Unreferenced data will be data within the file format that does not have an offset pointing to it.

5. ACRONYMS

Table 5-1 Acronyms

Acronym	Denotation
ASCII	American Standard Code for Information Interchange
DTG	Data Transfer Guidance
GML	Geography Markup Language
I/O	Input/Output
IFD	Image File Directory
ISG	Inspection and Sanitization Guidance
JPEG	Joint Photographic Experts Group
NGA	National Geospatial-Intelligence Agency
RGB	Red, Green, Blue
XML	Extensible Markup Language

6. REFERENCED DOCUMENTS

[1] TIFF Revision 6.0, Available at:
http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf

[2] GeoTIFF Format Specification Revision 1.0, Available at:
http://www.remotesensing.org/geotiff/spec/geotiffhome.html

[3] BigTIFF File Format Proposal documented by AWare Systems,
http://www.awaresystems.be/imaging/tiff/bigtiff.html

[4] BigTIFF Design documented by LibTIFF,
http://www.awaresystems.be/imaging/tiff/bigtiff.html

[5] Adobe PageMaker® 6.0 TIFF Technical Notes,
http://partners.adobe.com/public/developer/en/tiff/TIFFPM6.pdf

[6] Affine Transformation Wolfram MathWorld,
http://mathworld.wolfram.com/AffineTransformation.html

[7] Digital Negative (DNG) Specification
http://wwwimages.adobe.com/content/dam/Adobe/en/products/photoshop/pdfs/dng_spec_1.4.0.0.pdf

[8] Understanding What is stored in a Canon RAW .CR2 file, How and Why
http://lclevy.free.fr/cr2/#interpol

7. SUMMARY OF RISKS

Table 7-1 Summary of Risks

ISG Section	Specification	Hiding	Attack	Disclosure
4.1 File Size	TIFF Revision 6.0 BigTIFF Design documented by LibTIFF	x	x	
4.2 Image File Header	TIFF Revision 6.0 BigTIFF Design documented by LibTIFF	x		
4.3.1 Structure	TIFF Revision 6.0 BigTIFF Design documented by AWare Systems TIFF Technical Notes Supplement 2	x	x	x
4.3.2 Minimum Tags	TIFF Revision 6.0 BigTIFF Design documented by AWare Systems TIFF Technical Notes Supplement 2	x	x	x
4.3.3 Tag Order	TIFF Revision 6.0 BigTIFF Design documented by AWare Systems TIFF Technical Notes Supplement 2	x		

4.4.1 TIFF Tag Format	TIFF Revision 6.0 BigTIFF Design documented by AWare Systems GeoTIFF Format Specification Revision 1.0	x	x	x
4.4.2 NewSubFileType/SubFileType	TIFF Revision 6.0 BigTIFF Design documented by AWare Systems GeoTIFF Format Specification Revision 1.0	x		x
4.4.3 Compression	TIFF Revision 6.0 TIFF Specification Supplement 2		x	
4.4.4 ClipPath	TIFF Revision 6.0 TIFF Specification Supplement 2	x		x
4.4.5 MakerNote	TIFF Revision 6.0 TIFF Specification Supplement 2 DNG Specification	x		
4.4.6 ColorMap	TIFF Revision 6.0 TIFF Specification Supplement 2	x		

4.4.7 SubIFD	TIFF Revision 6.0 TIFF Specification Supplement 2	x	x	x
4.5.1 GeoTIFF Tags	GeoTIFF Format Specification Revision 1.0	x		
4.5.2 GeoTIFF Keys	GeoTIFF Format Specification Revision 1.0	x		x
4.5.3 GeoTIFF Key Codes	GeoTIFF Format Specification Revision 1.0			x
4.5.4.1 ModelTiepoint Tag	GeoTIFF Format Specification Revision 1.0			x
4.5.4.2.1 GeoTIFF Configuration Keys	GeoTIFF Format Specification Revision 1.0			
4.5.4.2.2. Geographic Coordinate System Parameter Keys	GeoTIFF Format Specification Revision 1.0			x
4.5.4.2.3 Projected CS Parameter Keys	GeoTIFF Format Specification Revision 1.0			x
4.5.4.4 Geo_Metadata Tag	GeoTIFF Format Specification Revision 1.0	x		x
4.6 Unreferenced Data	TIFF Revision 6.0	x		x