Inspection and Sanitization Guidance for Exchangeable Image File Format (Exif)

Version 1.0

8 August 2016

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
8 August 2016	1.0	Initial Version
12/13/2017	1.0	Updated Contact information, IAC Logo, Cited Trademarks and Copyrights, Expanded Acronyms, and added Legal Disclaimer

DISCLAIMER OF WARRANTIES AND ENDORSEMENT

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 Exchangeable Image File Format (Exif)* provides guidance for file inspection and sanitization software for Exif version 2.3, developed by the Camera and Imaging Products Association (CIPA) and the Japan Electronics and Information Technology Industries Associations (JEITA).

Exif is structured, tagged metadata contained within some media file formats. This data is used by digital camera manufacturers and applications that process digital images to provide additional information about media files. The metadata includes manufacturer specific information such as the make, model and lens information of the device that generated the file; image information (e.g., date/time of capture) and geolocation information (e.g., latitude/longitude) can also be recorded. Exif data is found in two image standards: Joint Photographic Experts Group (JPEG) File Interchange Format (JFIF) (as defined in International Standards Organization/International Electrotechnical Commission (ISO/IEC) 10918-1) and TIFF Revision 6.0[1]. The Exif format is also defined for audio files in the format of Resource Interchange File Format (RIFF) Waveform Audio File Format (WAVE).

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

[1] 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 FIGURES

LIST OF TABLES

1 SCOPE

1.1 Purpose

The purpose of this document is to provide guidance for the development of a sanitization or analysis software tool for files containing Exchangeable image file format (Exif) components. This document analyzes the various elements contained within Exif components and then discusses data attack, data disclosure, and data hiding risks. It describes how these elements can be a cause for concern for hidden, sensitive data or for attempts to exploit a system. This report provides numerous recommendations and mitigations that could be used to ensure the use of Exif components in image and audio files 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 on file inspection and sanitization applications that involve processing media files with Exif components.

1.2 Introduction

Exif is structured, tagged metadata that describes media content encapsulated in a supported file format. The data is used by digital camera manufactures and imaging applications to provide structured data about media files. Imaging applications can add, view and/or manipulate the Exif data stored in a media file. The metadata includes manufacturer specific information such as make, model and lens information. Image information like the date or time of capture and geolocation information such as latitude/longitude can also be recorded.

1.3 Background

Exif is a standard jointly published by the Camera & Imaging Products Association (CIPA) as document DC-008-2012, and also by the Japan Electronics and Information Industries Association (JEITIA) as document CP-3451C. Both published standards recognize the current Exif version as being 2.3, published in 2012. The previous Exif standard, version 2.21, was published in 2009 by the JEITIA as document CP-3451 and CIPA as document DCG-008-2009.

The initial Exif specification, revision 1.0, was published in October, 1995 and included the basic structure and tag definitions. Revision 1.1 incorporated additional tags. Revision 2.0 was a major update and included tags for color space, global positioning system (GPS), audio files, and compressed thumbnail data. Revision 2.1 included tags for interoperability. Revision 2.2 included additional tags for print quality and GPS.

Files that contain Exif metadata should follow the interoperability guidance established by the Design rule for Camera File system (DCF) in the CIPA DC-009-2010 document [1]. The guidance deals with file systems, directory structure, and file naming conventions. DCF interoperability is covered in the Exif specification.

Table 1-1 provides an overview of the tag changes in the Exif specification between versions 2.2, 2.21, and 2.3. The table indicates which version of the Exif specification contained the update for the given tag and all changes are carried forward. Exif version 2.2 included additional tags for positioning and GPS; the GPS tag definitions were updated in version 2.3. Exif version 2.21 included tag updates for color space and flash tags. Exif version 2.3 added and updated tags related to sensitivity, the camera and lens, light-source color, orientation and GPS.

Table 1-1 Exif Specification Tag Updates

Tag Name	Tag ID	Type	Exif Specification		
			2.2	2.21	2.3
Tags for Image Data					
Gamma	42240	Rational		X	X
ColorSpace	40961	Short		X	
Other Tags					
Camera Owner Name	42032	ASCII			X
Body Serial Number	42033	ASCII			X
Lens Specification	42034	Rational			X
Lens Make	42035	ASCII			X
Lens Model	42036	ASCII			X
Lens Serial Number	42037	ASCII			X
Print Quality Tags			X		
GPS Tags					
Positioning and GPS tags added			X		
Horizontal Position Error	31	Rational			X

1.4 Document Organization

This section summarizes the organization of this document. Table 1-2 provides a brief description of each section of this document.

Table 1-2 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 Exif components
Section 4: Exif Constructs	This section contains the Exif 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 Exif components contained in image 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.

Table 1-3 summarizes the recommendation actions:

Table 1-3 Recommendation Actions

Recommendation Action	Comments
Validate	Verify the data structure's integrity, which may include integrity checks on other components in the metadata. (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 provides guidance on Exif data that conform to the JEITIA and CIPA format specifications revision 2.3. Exif revisions prior to 2.2 utilize similar constructs; however, the Exif tag definitions may differ. Tags may not exist in a previous revision or the format of a tag may be different. The previous versions of these tags are not addressed by this guidance document.

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.

2 CONSTRUCTS AND TAXONOMY

2.1 Constructs

This document describes many of the constructs used in Exif format, but it does not describe every construct, thus this document is not to be treated as a complete reference. Developers of an Exif 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

Table 2-1 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 Exif.
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

The Exif version 2.3 specification contains components that describe images and audio that is part of a container file format. The Exif format is found in the Joint Photographic Experts Group (JPEG) File Interchange Format (JFIF) file format (ISO/IEC 10918-1) and TIFF file format (Rev 6.0)[2]. The Exif standard also provides for embedding Exif components into audio files; Exif can be encoded in Resource Interchange File Format (RIFF) Waveform Audio File Format (WAVE) files. Exif data should exist in valid locations that adhere to the respective specification.

The basic Exif structure, which is common to both JPEG and TIFF files, is based on a TIFF construct called an image file directory (IFD) [3] . The Exif audio metadata is contained in a tag structure that differs from the TIFF based construct and does not include value offsets. The Exif specification provides guidance on the format and structure for the inclusion of metadata using existing file structures within the parent file format.

The Exif IFD allows camera manufactures and image renders to read and write informational data in a structured manner. Exif uses metadata to provide additional detail for image and audio files. Additionally, GPS related data and interoperability data is included in Exif metadata.

Exif data has been used to enable many exploits including cross-site scripting, and arbitrary code execution. The vast majority of the exploits have taken advantage of either poor input validation or errors in commonly used Exif parsing libraries.[3] Examples of some of the exploits can be found in the Common Vulnerability Exposures (CVE) database[4].

Another avenue of recent attack has been through hiding malicious code inside of the Exif data elements. The hidden code is often placed in Exif common ASCII tags, such as camera maker and model. Protections must be provided to Exif viewers to ensure that malicious code, carried within tags, and often obfuscated is detected and filtered to avoid attacks of this sort[5].

3.1 Exif Data Structure

The Exif metadata format, as described in the specification, contains details on how a file containing Exif data should be constructed for image files (TIFF and JPG images). However, these same file formats, TIFF and JPG, include Exif data according to their respective specifications. The following sections describe how Exif data should be formatted into image files and how files use Exif in practice.

[2] Currently the Portable Network Graphic (PNG) specification does not support the direct embedding of Exif data like JPEG and TIFF do though it does have a robust metadata storage capability.

[3] See http://libexif.sourceforge.net/ for more information

[4] http://www.cvedetails.com/vulnerability-list/vendor_id-2861/Libexif.html

[5] https://www.whitefirdesign.com/blog/2014/07/07/hackers-hiding-malicious-code-in-exif-data-of-images

3.1.1 Exif in TIFF

The TIFF file image format is composed of two primary structures, the image header and IFDs. The TIFF image header points to the first IFD; each IFD points to any subsequent image file directories. The Primary TIFF IFD contains pointers to the Exif IFD and an optional GPS IFD. The Exif specification is compliant with the TIFF specification; Exif data placement in both specifications is the same.

The Exif IFD contains metadata tags and an optional Interoperability IFD. The Interoperability IFD contains a tag with a value that indicates with which Design Rule for Camera File system rules to conform [1]. See Figure 3-1 for an illustration of Exif components in a TIFF file.

Figure 3-1 illustrates the TIFF header and Primary image IFD, with pointers to Exif and optional GPS IFD. The Exif data is contained in a separate Exif IFD as the GPS data is also contained in a separate GPS IFD.

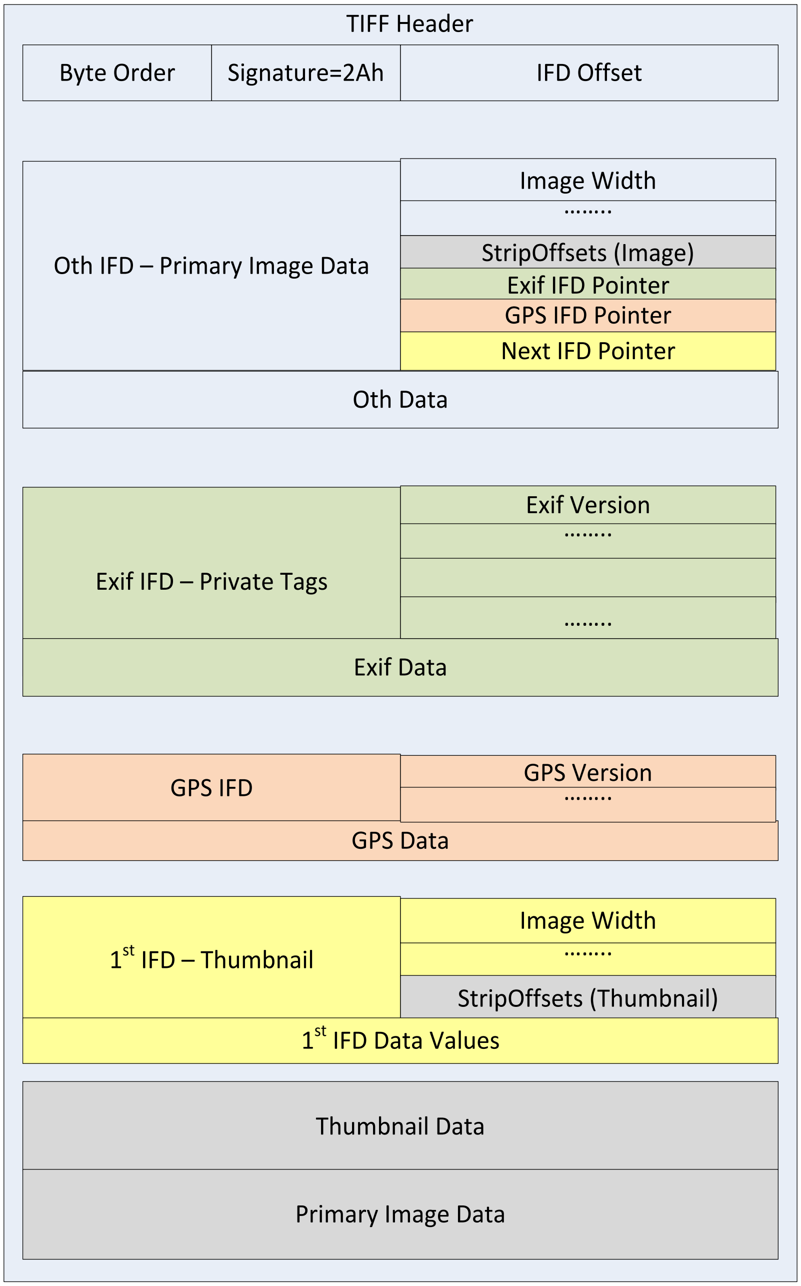


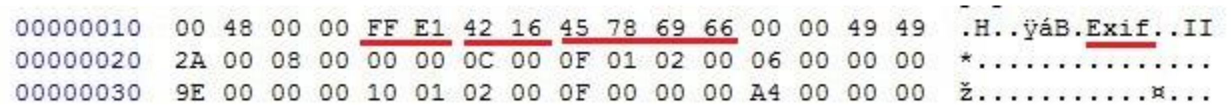
Figure 3-1 Exif in TIFF Structure

3.1.2 Exif in JFIF

JFIF uses an application marker segment to store Exif data; however the Exif and JFIF specifications prescribe different methods of storing this data.

JFIF files are required to contain an APP0 marker segment as the first segment in the file, but the Exif specification defines APP1 as the first marker segment for JFIF files [6]. Note that most JFIF applications do not follow the Exif specification; rather they follow the JFIF specification and begin with segment marker APP0. The APP0 segment in JFIF files with Exif data was observed to have the minimum length of 16 when a file was generated with Exif Data. However, files without the APP0 marker were opened by most applications.

A specific Exif identifier code distinguishes the segment as Exif and is followed by the size of the segment; this includes all segment information except the two byte marker. The Exif marker is followed by a full TIFF file that contains the Exif data, but should not contain image data. Figure contains sample hexadecimal data for application marker one (0xFF E1), the segment size (0x4216), the identifier code “Exif” (0x45 78 69 66) and two termination bytes (0x00 00). Beginning at “II” (0x4949) is the TIFF file; note that the entire TIFF file is not contained in Figure 3-2.



```
00000010   00 48 00 00 FF E1 42 16 45 78 69 66 00 00 49 49   .H..ÿáB.Exif..II
00000020   2A 00 08 00 00 00 0C 00 0F 01 02 00 06 00 00 00   *...............
00000030   9E 00 00 00 10 01 02 00 0F 00 00 00 A4 00 00 00   ž...........¤...
```

Figure 3-2 Exif APP1

The Exif components are contained in the APP1 marker, and only the APP2 or Com segments are used according to the Exif specification. Any undefined application segments, i.e. APPn (other than APP1 or APP2), are not expressly prohibited, but they are not used by Exif. The Exif specification indicates that other APP markers should be ignored. The standard allows for APPn markers that are undefined and likely used by vendors for application specific extensions. See Figure 3-3 for an illustration of embedding Exif in a JFIF file.

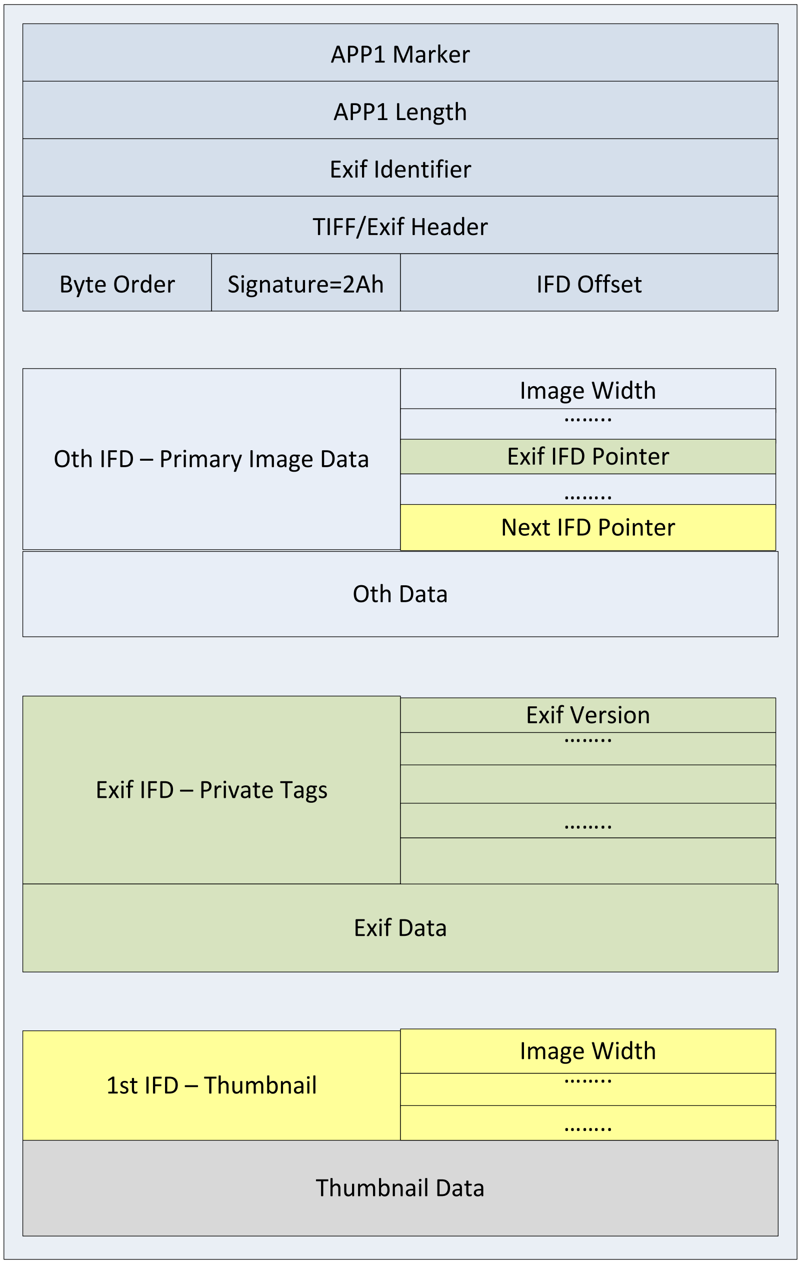
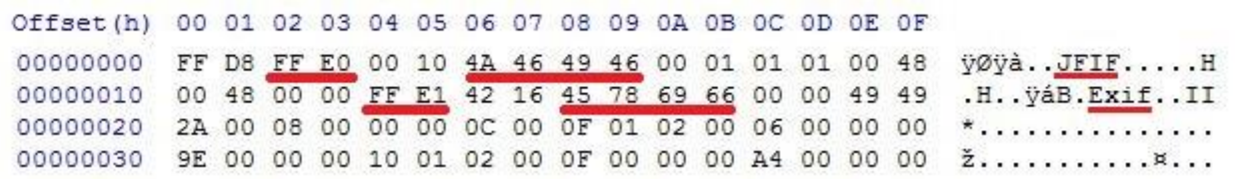


Figure 3-3 Exif in APP1

Figure 3-3 illustrates Exif metadata placed into an APP1 Segment using the file structure indicated in the Exif specification.

JFIF files are required to contain an APP0 marker segment as the first segment in the file. Whereas the Exif specification defines the APP1 marker to be the first marker segment for JFIF files. In all of the jpeg files we have reviewed the Exif marker segment is not the first segment, but follows the standard JFIF segment marker APP0.

Figure 3-4 illustrates the JFIF compliant header and application segment markers found in JFIF files. The first segment maker, following the start of image marker, is the APP0 (0xFF E0). The Exif marker segment (APP1) follows the JFIF marker.



```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 48   ÿØÿà..JFIF.....H
00000010   00 48 00 00 FF E1 42 16 45 78 69 66 00 00 49 49   .H..ÿáB.Exif..II
00000020   2A 00 08 00 00 00 0C 00 0F 01 02 00 06 00 00 00   *...............
00000030   9E 00 00 00 10 01 02 00 0F 00 00 00 A4 00 00 00   ž...........¤...
```

Figure 3-4 JFIF with Exif

3.2 Exif Components

Exif components include metadata concerned with the image, manufacturer, geolocation of the image, and interoperability. Pointers to the Exif IFD and GPS IFD are recorded in the Primary Image IFD. Figure 3-5 illustrates the unique IFDs associated with the Exif data format.

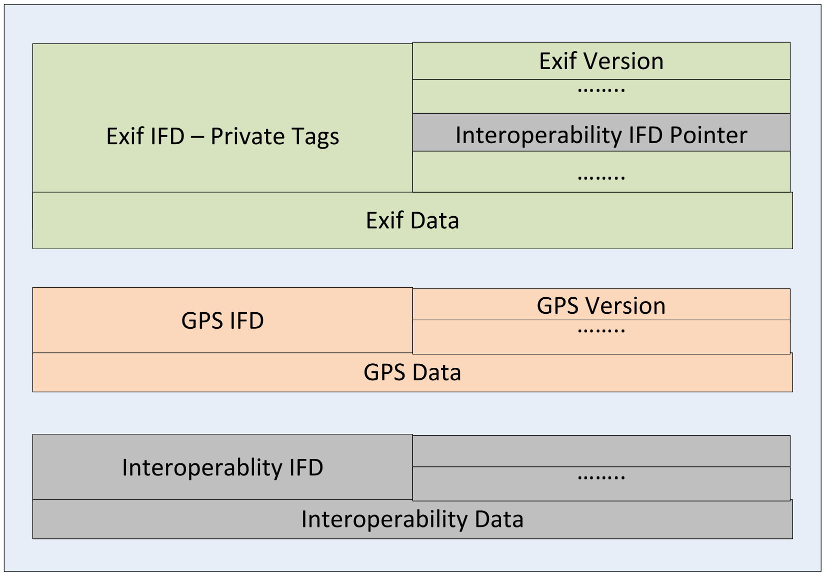


Figure 3-5 Exif IFD Components

3.2.1 Image File Directory (IFD)

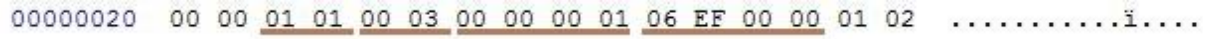
The TIFF Image File Directory (IFD) was reused by Exif to contain metadata about images. The Exif specification defines three IFD types: the Exif IFD, the GPS IFD, and the Interoperability IFD

The IFD is a simple structure that groups collections of twelve-byte interoperability arrays or Exif tags. 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 indicate that there are no more IFDs with 0x0, or it contains the offset to the next IFD, as shown in Figure 3-6.

No. of IFD Entries	IFD Entry 1	IFD Entry 2	….	IFD Entry *n*	Next IFD Offset
2 Bytes	*12 Bytes*	*12 Bytes*		*12 Bytes*	*4 Bytes*

Figure 3-6 Image File Directory Structure

Figure 3-7 illustrates the components of a TIFF tag. The tag id (0x01 10) indicates that this is the ImageLength tag. The type field value (0x00 03) indicates that the type is a short value. The length field value (0x00 00 00 01) indicates that the value is contained in the value field and is not an offset to the tag data. The final value is the tag value (0x06 EF 00 00).



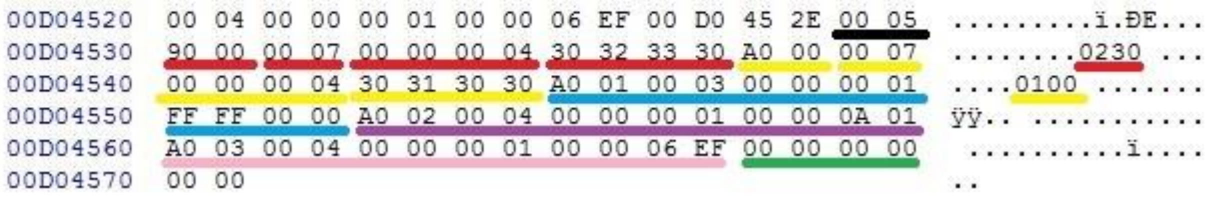
```
00000020   00 00 01 01 00 03 00 00 00 01 06 EF 00 00 01 02   ..........ï.....
```

Figure 3-7 Example TIFF Tag

3.2.1.1Exif IFD

The Exif IFD contains the collection of Exif related attribute tags in the image file. The Exif IFD is pointed to by the private Exif tag value (decimal value 34665), which contained in the first TIFF IFD. The last tag in the IFD is a pointer to the next IFD, which is recorded as zeros when there is no next IFD, which is the expected case for the Exif IFD. The Exif IFD should not contain image data.

Error! Reference source not found.Error! Reference source not found., below, lustrates a sample Exif IFD. The IFD contains five tags. Each tag structure is illustrated in a different color. The first tag, underlined in red, is the Exif version tag beginning with the hex value 0x90 00. The Exif version tag in this example indicates the current Exif version "'2.3" (0x30 32 33 30). The second tag, underlined in yellow, has id 40960(0xA0 00) and contains the supported FlashPix version. Three additional tags follow, underlined in blue, purple and pink; the IFD is terminated by zero (0x00 00 00 00). Note that the last value, underlined in green, can contain an offset to another Exif IFD.



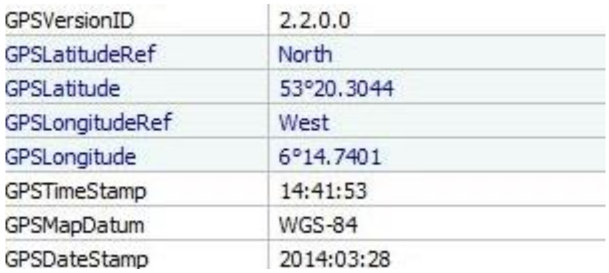
```
00D04520   00 04 00 00 00 01 00 00 06 EF 00 D0 45 2E 00 05   .........ï.ÐE...
00D04530   90 00 00 07 00 00 00 04 30 32 33 30 A0 00 00 07   ........0230 ...
00D04540   00 00 00 04 30 31 30 30 A0 01 00 03 00 00 00 01   ....0100 .......
00D04550   FF FF 00 00 A0 02 00 04 00 00 00 01 00 00 0A 01   ÿÿ.. ...........
00D04560   A0 03 00 04 00 00 00 01 00 00 06 EF 00 00 00 00    ..........ï....
00D04570   00 00                                             ..
```

Figure 3-8 Exif IFD Sample

3.2.1.2 GPS IFD

The GPS IFD contains the set of tags that are used for recording geolocation data. It is pointed to by the GPS tag (0x88 25) from the first TIFF IFD. When the GPS IFD is present, the only mandatory tag is the GPS version tag (tag 0).

An example of GPS Exif data is shown below in Figure 3-9. The current Exif specification contains approximately thirty GPS related tags. In addition to directional tags, GPS attributes also include data on time, GPS satellites, speed of receiver, and GPS processing methods and corrections.



GPSVersionID	2.2.0.0
GPSLatitudeRef	North
GPSLatitude	53°20.3044
GPSLongitudeRef	West
GPSLongitude	6°14.7401
GPSTimeStamp	14:41:53
GPSMapDatum	WGS-84
GPSDateStamp	2014:03:28

Figure 3-9 GPS IFD Example Tag Values

3.2.1.3 Interoperability IFD

The Interoperability IFD refers to the JEITA specification for the Design rule for Camera File system.[6] The interoperability of the file refers to the directory structure, file naming conventions, character set, file format and metadata. The Interoperability IFD currently contains only one tag, the Interoperability tag (tag 1). It is pointed to by the tag value (40965) written in the Exif IFD. The Interoperability Index is an American Standard Code for Information Interchange (ASCII) string that is used to express the conformance, or interoperability, of the given file against a rule specification. The interoperability rules are given in Table 3-1.

Table 3-1 Interoperability Rules

Interoperability Index	Interoperability Rule
"R98"	File conforms to the R98 file specification (Exif R 98) or the DCF basic file standard by Design Rule for Camera File System

[6] JEITA CP-3461B, http://www.jeita.or.jp/japanese/standard/book/CP-3461B_E

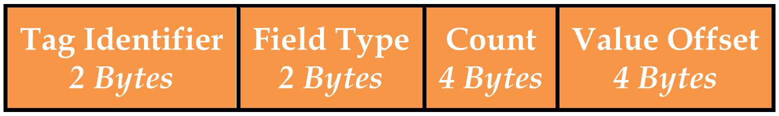
“THM”	File conforms to the DCF thumbnail file specification by Design Rule for Camera File System
“R03”	File conforms to the Option File specification by Design Rule for Camera File System

3.2.2 Exif Tags

Exif tags are twelve-byte fields found within Exif, GPS, and Interoperability IFDs. Exif tags contain only metadata. For example, the date and time an image was created can be stored in the DateTime tag in the Exif IFD.

Exif tags are composed of four fields: the tag identifier, the field type, the number of values, and the value offset, as illustrated in Figure 3-10. The possible values for each tag structure is outlined in the Exif specification. The tag identifier is a value that corresponds to a known tag definition (e.g., if the tag identifier is 0xA001 (40961), then the following information deals with color space information). Field types represent the type and size of data for the tag; some tags require specific field types. The Count contains the number of actual values, not total bytes, for the tag structure. For example, if a count value is two and a type is Long, then the total number of bytes for the value would be eight bytes.

The value offset contains either the actual value, when the value is four bytes or less, or an offset to the data containing the value.



Tag Identifier *2 Bytes*	Field Type *2 Bytes*	Count *4 Bytes*	Value Offset *4 Bytes*

Figure 3-10 Exif Tag Structure

3.2.2.1Exif Tag 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 Table 3-2.

Table 3-2 Baseline Exif Field Types

Type by Value	Size
1 = BYTE	8-bit unsigned integer.
2 = ASCII	8-bit byte that contains a 7-bit ASCII code; the last byte must be NULL (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.
9 = SLONG	A 32-bit (4-byte) signed (twos-complement) integer.
10 = SRATIONAL	Two SLONGs: the first represents the numerator of a fraction, the second the denominator.

3.2.2.2 Exif Tag Field Value

The Value field can contain data or an offset pointing to data. If the tag Count and Type (see Table 3-2) indicate that the number of bytes is less than or equal to four bytes, then the Value field contains the actual value. If the Value is greater than four bytes, then the Value field contains an offset position, from the start of the TIFF header to the position of the actual value.

Figure 3-11 illustrates the relationship between tag values and the Exif data block. When a tag value is greater than four bytes in length, the tag value then points to the actual value contained in the Exif data block. The diagram highlights how the value for the tag ExposureTime is contained in the Exif data block and is pointed to by the tag-value offset in the Exif IFD.

Figure 3-11 also illustrates that the UserComment tag includes an eight byte code designation before the data value in the Exif Data block. The code designation is used to indicate the character code set for the data that follows. This code allows for additional character codes to be used, such as a two byte character code. See Table 3-3 for the list of defined UserComment character codes.

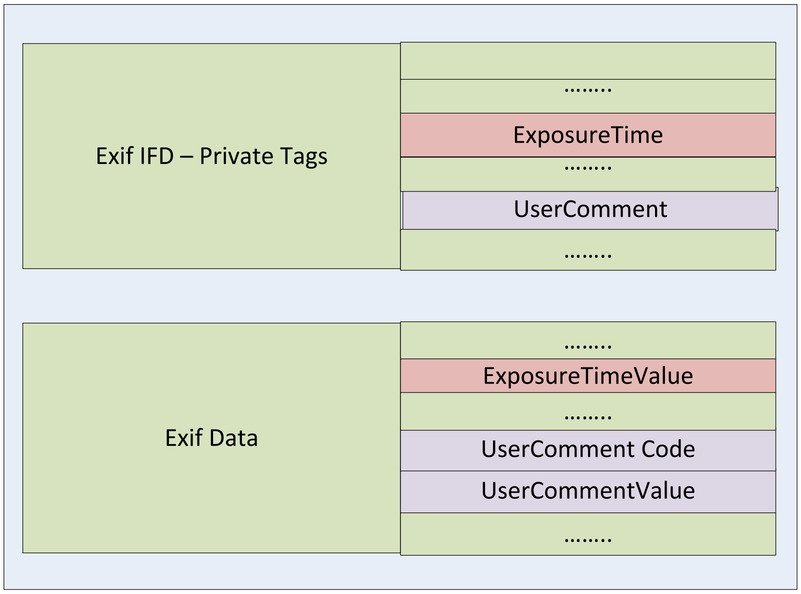


Figure 3-11 Exif Tag Values

Table 3-3 Character Codes

Character Codes	Code Identifier 8 bytes	Reference
ASCII	"ASCII"000	ITU-T T.50 IA5
Japanese Industrial Standard (JIS)	"JIS"00000	JIS X208-1990
Unicode	"UNICODE"0	Unicode Standard
Undefined	00000000	Undefined

3.3 Exif Audio

Exif data for audio files is contained in RIFF WAVE audio format files. The audio files are designed to be "data-compatible" with Windows WAVE audio files.

The general attribute data is written to an INFO list; the Exif attributes are written in Exif chunks. The Exif attributes are written in such a way as to be compatible with the INFO list data.

Exif attribute data is contained in a pre-registered generic LIST chunk. The LIST chunk contains a set of sub-chunks. This attribute chunk LIST is compatible with all chunk types, so it is compatible with the WAVE formatted audio file. The actual Exif data is written to the sub-chunks of the LIST chunk, which has a LIST type value of “Exif”.

The LIST chunk is written as a subchunk in the RIFF chunk. It should be noted that the LIST chunk itself may also contain subchunks. See Figure 3-12 for an illustration of the LIST structure in a WAVE file.

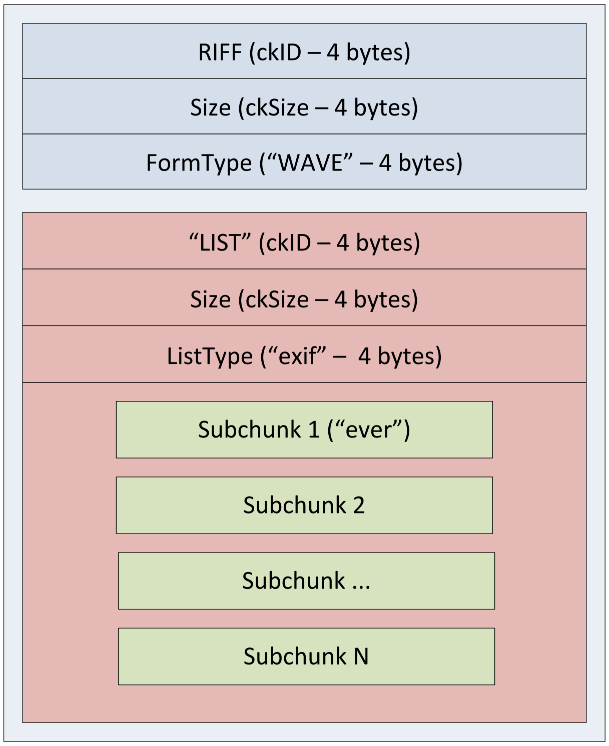


Figure 3-12 Exif Audio LIST Chunk

3.3.1 Exif Audio Chunks

A dedicated chunk is used to record the Exif attribute information in the audio files. The table below describes the Exif chunks and the information they contain.

Table 3-4 Exif Audio Chunks

Chunk ID	Description	Comments
“ever”	Contains version number of the Exif standard	ASCII string, but may not be null terminated. This is the only mandatory chunk.
“erel”	Contains information pointing to Exif image file using Exif file naming rules.	ASCII string null terminated. If audio file is related to an Exif image

		file, then this chunk is considered mandatory.
“etim”	Contains the time when the audio file was created.	ASCII string null terminated. Format is: hour, minute, second, subsecond
“ecor”	Contains manufacturer of recording equipment.	ASCII string null terminated.
“emdl”	Contains model of the recording equipment.	ASCII string null terminated.
“emnt”	Contains manufacturer design information.	Contents defined by manufacturer.
“eucm”	Contains user comments	Character code (ASCII, JIS, Unicode, Undefined) precedes the data.

3.3.2 Exif Audio Chunks Structure

The Exif audio chunk structure consists of a four-byte chunk ID, followed by a four-byte chunk size and then the chunk data. The chunk ID is an ASCII value and is defined in Table 3-4. The chunk size defines the total number of bytes required for the chunk data. The chunk data is either an ASCII value or a numeric value of length chunk size.

Error! Reference source not found., below, illustrates the contents of the Exif Audio ubchunks for a file with four Exif subchunks. Subchunk 1 is the mandatory “ever” subchunk with a version of 2.21. Subchunk 2 is the “erel” subchunk indicating that the related Exif image file is “DSC00005.JPG”. Subchunk 3 is the “etim” subchunk containing the time. Subchunk 4 is the “ecor” subchunk containing the equipment manufacturer. The chunk data immediately follows the chunk size. No value offsets are used in Exif audio metadata.

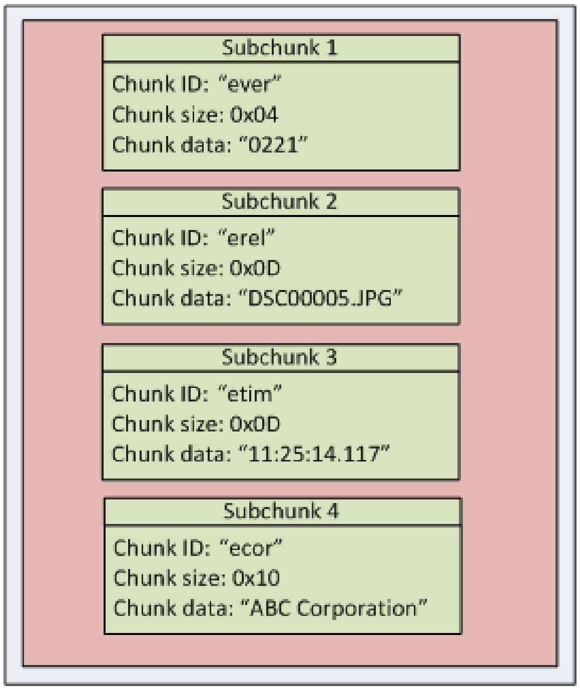
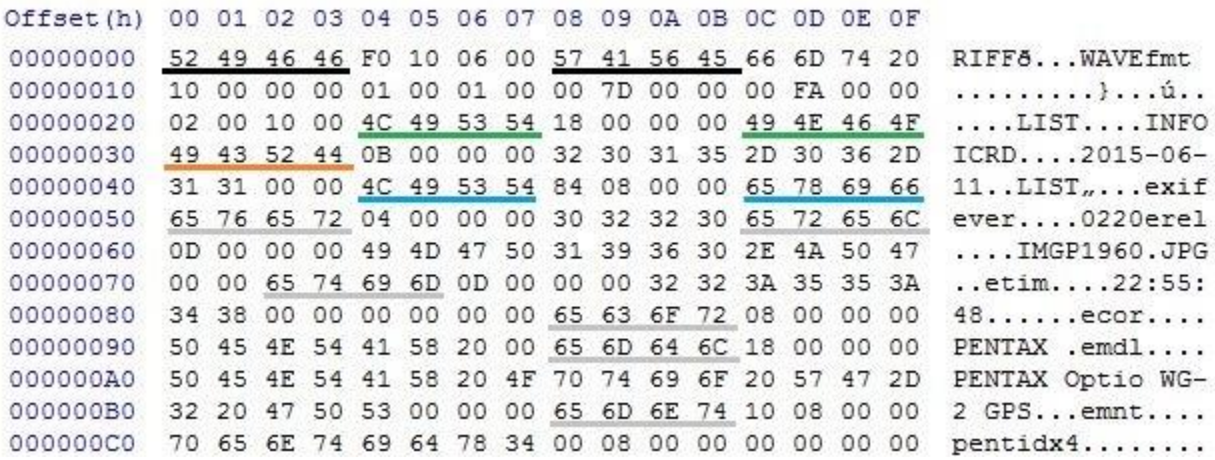


Figure 3-13 Exif Audio Subchunks

3.3.3 Exif Audio Structure

Figure 3-14 contains a sample from an Exif audio file. The initial black underlining indicates that the file is a RIFF WAVE file. The green underlining indicates the first LIST section that contains an ICRD (creation date) chunk. The blue underlining begins the Exif LIST section. The Exif LIST section contains six grey underlined sub chunks identifiers. Each Exif sub-chunk identifier is followed by a chunk size and then the chunk data. The first Exif chunk is the "ever" mandatory chunk. This chunk is four bytes in size and contains the value "0220".



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  52 49 46 46 F0 10 06 00 57 41 56 45 66 6D 74 20  RIFFð...WAVEfmt 
00000010  10 00 00 00 01 00 01 00 00 7D 00 00 00 FA 00 00  .........}...ú..
00000020  02 00 10 00 4C 49 53 54 18 00 00 00 49 4E 46 4F  ....LIST....INFO
00000030  49 43 52 44 0B 00 00 00 32 30 31 35 2D 30 36 2D  ICRD....2015-06-
00000040  31 31 00 00 4C 49 53 54 84 08 00 00 65 78 69 66  11..LIST„...exif
00000050  65 76 65 72 04 00 00 00 30 32 32 30 65 72 65 6C  ever....0220erel
00000060  0D 00 00 00 49 4D 47 50 31 39 36 30 2E 4A 50 47  ....IMGP1960.JPG
00000070  00 00 65 74 69 6D 0D 00 00 00 32 32 3A 35 35 3A  ..etim....22:55:
00000080  34 38 00 00 00 00 00 00 65 63 6F 72 08 00 00 00  48......ecor....
00000090  50 45 4E 54 41 58 20 00 65 6D 64 6C 18 00 00 00  PENTAX .emdl....
000000A0  50 45 4E 54 41 58 20 4F 70 74 69 6F 20 57 47 2D  PENTAX Optio WG-
000000B0  32 20 47 50 53 00 00 00 65 6D 6E 74 10 08 00 00  2 GPS...emnt....
000000C0  70 65 6E 74 69 64 78 34 00 08 00 00 00 00 00 00  pentidx4........
```

Figure 3-14 Exif Audio Example

4 EXIF CONSTRUCTS

In this section, specific features and risks of each identified Exif component are considered. Each construct provides a description, areas of concern, some examples, and recommendations for potentially mitigating the risks.

4.1 Image File Directory (IFD)

OVERVIEW

The Exif specification defines three IFD components: Exif IFD, GPS IFD, and Interoperability IFD. The Exif IFD is to be placed in the APP1 section per the Exif specification. The IFDs all share a common structure. This IFD construct provides a common method for handling the three IFDs.

An IFD specifies the number of tags and is followed by a series of twelve-byte directory entries. Its location is dependent on the particular IFD pointer indicated in a specific tag. The basic structure of an IFD is:

- The first two bytes designate the number of tags, n.
- The next n*12 bytes contain n twelve-byte structures known as IFD tags.
- The final four bytes contain the offset to the next directory or 0.

RISKS AND RECOMMENDATIONS

Data Attack – Files with invalid offsets, unordered tags, or unknown tags may indicate that the file was specially crafted to exploit a machine. By having offsets that create recursive loops among the offsets one could trigger a stack buffer overflow[7]. Tags that are not recognized may contain malicious code instead of data pertinent to the file. Because Exif applications are instructed by the specifications to ignore tags they do not know, these fields are often not checked.

1. 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.
2. Validate – Validate that the IFD chain ends with an offset that points to 0x0.
3. Validate – Validate that the tags adhere to a whitelist of acceptable values. Tags are documented in the Exif specifications and other application specific documentation.
4. Reject – Reject files that contain malformed IFDs.

[7] CVE-201-2814: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-2814

Data Hiding and Data Disclosure– 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 IFD chain ends with an offset that points to 0x0.
3. Validate – Validate the tags adhere to a whitelist of acceptable values. Acceptable values are tags defined by the specification or an authoritative source[8].
4. Replace – Replace files with multiple IFDs with a single flattened version.
5. Reject – Reject files that contain multiple IFDs.

PRODUCT

Exif version 2.2, 2.21, 2.3

LOCATION

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

4.2 Exif Tag Format

OVERVIEW

The Exif IFD entry, commonly called a 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 3-2).
- Bytes 4-8 contain the number of values for the tag. For example, a tag like ImageDescription can specify the offsets where image description values are located and the length of the description.
- Bytes 8-12 contain the tag data or the offset where the data for the tag is found.

All Exif renderers should be able to understand and process the required tags in a file containing Exif metadata. Applications are not required to utilize the information provided by tags not enumerated. Tags not defined in any specification serve as a potential data hiding vector. The definition for the Exif tags can be found in the Exif specification version 2.3.

Some applications may ignore parts of the Exif tags, especially modifications to the type of values or upper bytes of known values that do not fill the whole four bytes. This serves as a potential data hiding vector because these values do not have to be shown to end users. Tags can contain malicious scripts or code. Inserting the malicious code only

[8] Exif Tags: http://www.sno.phy.queensu.ca/~phil/exiftool/TagNames/EXIF.html

requires an attacker to insert the code in an Exif ASCII tag. For example, using the application exiftool the following command[9] inserts Hypertext Preprocessor (PHP) code into an Exif tag:

```
      Exiftool –documentname=’<?php echo “foobarbaz”; ?>’
```

RISKS AND RECOMMENDATIONS

Data Hiding and Data Attack – 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. PHP code has been known to be placed in Exif tags and cause denial of service or execution of arbitrary code[10]. Careful whitelisting of the tag values is necessary to prevent Exif from carrying PHP attacks. Several well published attacks have used PHP scripts embedded in Exif tags [7].

1. Validate – Validate that the Exif Tag ID, bytes 1-2, match a known tag defined in the baseline specification for Exif version 2.3.
2. Validate – Validate that the Exif 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 that the number of values, bytes 5-8 or bytes 5-12, do not exceed a prescribed value in the specifications.
5. Validate – Validate that the tag data or offset to the data, bytes 9-12 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. Remove- Remove all tags and metadata that are not required by Baseline Exif renders to be processed. This may compromise image quality in certain images.
8. Reject – Reject files that contain unknown tags.
9. Reject – Reject files that contain malicious code in tags.

PRODUCT

Exif version 2.2, 2.21, 2.3

LOCATION

Exif IFD

[9] See: http://websec.io/2012/09/05/A-Silent-Threat-PHP-in-EXIF.html

[10] CVE-2014-3670: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3670

4.3 Exif Tags

The following constructs address specific Exif tags that can contain data risks. Note that the structure of the tags should be compliant with Section 4.1.2. Note that these constructs **will not** address structural validation.

4.3.1 MakerNote Tag

OVERVIEW

The MakerNote tag, with tag id 0x92 7c, contains proprietary information that is not always disclosed to the general public[11]. This tag is application specific, and it cannot be reliably parsed or extracted since the format is unknown. The structure and contents of the tag are undefined. The maximum number of MakerNote tags allowed to exist in Exif is also undefined. This could lead to a large amount of undefined data residing in this location.

RISKS AND RECOMMENDATIONS

Data Hiding, Data Disclosure and Data Attack - MakerNote tags that are not validated can pose a risk to application parsers and may contain sensitive data. Because the MakerNote data format is proprietary, the tag data may contain obfuscated information about the image (placed intentionally or unintentionally), which cannot be reliably verified or inspected. The content of the tag value may be an active content script[12] which could allow an attacker to gain privileges on a remote machine when an application attempts to process the MakerNote tag.

1. Remove- Remove all MakerNote tags in the Exif IFD. Note this will cause loss of information when using proprietary applications that understand the MakerNote tag.

PRODUCT

Exif version 2.2, 2.21, 2.3

LOCATION

Exif IFD

[11] http://www.exiv2.org/makernote.html
[12] CVE-2014-2333: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-2333

4.3.2 UserComment Tag

OVERVIEW

The UserComment tag, tag id 0x92 86, allows for additional comments to be added, in a defined character code, to the image file. The current set of character codes include: ASCII, JIS, Unicode, and Undefined.

RISKS AND RECOMMENDATIONS

Data Hiding and Data Attack - Tag values that are not validated can pose a risk to application parsers and may contain sensitive data. The content of the tag value may be an active content script which could allow an attacker to gain privileges on a remote machine[13].

1.
2. External Filtering Required – Pass the contents of the UserComment tag to a text filter that is capable of dirty word filtering and other text filtering techniques to ensure that comment is approved textual data.

PRODUCT

Exif version 2.2, 2.21, 2.3

LOCATION

Exif IFD

4.3.3 GPS Tag Sensitivity

OVERVIEW

GPS data often contains information that may be sensitive in nature. Sensitive data includes position (latitude, longitude, and altitude) about the captured image.

RISKS AND RECOMMENDATIONS

Data Disclosure– GPS tags may contain data that is sensitive or data that should not be disclosed.

[13] CVE-2014-1980: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-1980

1. Validate – Validate that the GPS tag is on a whitelist of acceptable GPS tags and the values are within approved ranges.
2. Remove – Remove any tags that are not on the accepted whitelist or are within a approved ranges.
3. Reject – Reject files that contain GPS tags that are not on the whitelist.
4.

PRODUCT

Exif version 2.2, 2.21, 2.3

LOCATION

GPS tags written in the GPS IFD.

4.4 Audio

OVERVIEW

Exif audio files detail methods for the writing of audio files with specific components. The Exif audio components include:

- Audio file format
- Detailed structure of the audio data
- Chunks within the file
- File naming conventions

The audio data recorded in an Exif audio file is written in RIFF tagged file structure. RIFF files are written in blocks of data and are referred to as chunks. The most commonly recorded attributes are written in an INFO list. The Exif attributes are written to the Exif chunks. Chunk data typically contains three components: chunk ID, chunk size, and chunk data. The only valid Exif audio chunks are listed in Table 3-4.

RISKS AND RECOMMENDATIONS

Data Hiding– Data hiding may be accomplished by inserting data into the Exif audio chunks that are not validated as part of the standard. The Specification guidance recommends that applications skip information chunks that they cannot process or are unknown.

1. Validate – Validate that only the known Exif Audio Chunks are present in the audio file.
2. Validate – Validate that the mandatory sub-chunk data “ever” is present. See Table 3-4 for a description of the “ever” sub-chunk.

3. External Filtering Required – Validate the sub-chunk data by using an external ASCII filter and that he contents adhere to acceptable values.
4. Remove – Remove any Exif audio chunks that are not defined in the specification.
5. Remove – Remove the undefined “emnt” chunk, since any manufacturer defined data can reside this this chunk.
6. Reject – Reject files that contain Chunks not contained in the specification.

PRODUCT

Exif version 2.2, 2.21, 2.3

LOCATION

Exif audio sub-chunks written in the List chunk.

5 ACRONYMS

Table 5-1 Acronyms

Acronym	Denotation
ASCII	American Standard Code for Information Interchange
CIPA	Camera and Imaging Products Association
DCF	Design rule for Camera File system
EOI	End of Image
FPXR	FlashPix Identifier
GPS	Global Positioning System
IFD	Image File Directory
ISG	Inspection and Sanitization Guidance
JEITA	Japan Electronics and Information Technology Industries Associations
JFIF	JPEG File Interchange Format
JIS	Japanese Industrial Standard
JPEG	Joint Photographic Experts Group
PHP	PHP Hypertext Processor
RGB	Red, Green, Blue
RIFF	Resource Interchange File Format
SOI	Start of Image
TIFF	Tagged Image File Format
WAVE	Waveform Audio File Format

6 REFERENCED DOCUMENTS

[1] *Design rule for Camera File system : DCF version 2.0*, CIPA DC-009-2010 http://www.cipa.jp/std/documents/e/DC-009-2010_E.pdf

[2] Exif Version 2.3, Camera & Imaging Products Association and Japan Electronics and Information Technology Industries Association, DC-008-Translation-012, Revised December, 2012

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

[4] Joint Photographic Experts Group (1992) *Information Technology – Digital Compression and Coding of Continuous-tone Still Images – Requirements and Guidance* ISO/IEC 10918-1 and CCITT Rec T.81

[5] *Inspection and Sanitization Guidance for TIFF File Formats*, Version 1.0, 3 February, 2015

[6] *Inspection and Sanitization Guidance for JPEG File Interchange Format*, Version 1.0, November, 2012

[7] *Inspection and Sanitization Guidance for Waveform Audio File Format,* Version1.0, March, 2012

[8] *A Silent Threat – PHP in Exif,* Chris Cornutt, 05 September, 2012, http://websec.io/2012/09/05/A-Silent-Threat-PHP-in-EXIF.html

7 SUMMARY OF RISKS

Table 7-1 Summary of Risks

ISG Section	Specification	Hiding	Attack	Disclosure