U/OO/234065-17

Inspection and Sanitization Guidance for JPEG File Interchange Format

Version 1.0

6 November 2012

National Security Agency
9800 Savage Rd, Suite 6721
Ft. George G. Meade. MD 20755

Authored/Released by:
Unified Cross Domain Capabilities Office
cds_tech@nsa.gov

DOCUMENT REVISION HISTORY

Date	Version	Description
11/6/2012	1.0	final
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

This *Inspection and Sanitization Guidance for JPEG File Interchange Format* document provides guidelines and specifications for developing inspection and sanitization software for continuous-tone images encoded using the Joint Photographic Experts Group (JPEG) encoding standard, ISO 10918-1 and CCITT Rec. T.81, and the JPEG File Interchange Format Version 1.02.

The JPEG Interchange Format (JIF) specification, as defined in Annex B of ISO 10918-1, is meant to be a minimal data format for the purpose of sharing compressed image data among applications. Other file formats have emerged from or incorporated the JPEG standard, including JPEG File Interchange Format (JFIF) (C-Cube Microsystems), Still Picture Interchange File Format (SPIFF) (ISO/IEC 10918-3, Annex F) and Exchangeable image file format (Exif).

Two of the most widely used JPEG file formats are JFIF and Exif. This document will focus on these formats for the purposes of inspection and sanitization. Identified risks of JFIF and Exif include data hiding in the data structure, data disclosure in comment marker segments or marker segments custom to a specific application, and buffer overflow from invalid marker segments. This document will provide guidance to mitigate risks determined through this inspection.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DOCUMENT REVISION HISTORY

Date	Version	Description
12/22/10	0.1	Initial Draft
2/4/2011	0.2	Added JFIF APP0 marker segment and JFIF extension APP0 marker segments. Reordered structure table.
2/14/2011	0.3	Renamed section 4 to section 6. Renamed section 3 to section 5. Inserted new sections 3 and 4. Rewrote section 2. Rewrote section 5. Rewrote section 6. Renamed title.
6/9/2011	0.4	Modified section headers and footers. Removed paragraph classification markings.
7/26/2011	0.5	Added JPEG Extension marker segments from ISO 10918-3.
7/5/2012	0.6	Added additional recommendations. Added JFIF specific marker recommendations. Changed all sources to JFIF.
7/17/2012	0.7	Added NXPowerLite and Exif recommendations.
8/10/2012	0.8	Added ICC and updated to address comments
9/6/2012	0.9	Updated 5.1 recommendations to reflect research into 0xFFFF and 0xFF00 bytes.
11/6/2012	1.0	Final Release

1. SCOPE

1.1 Purpose of this Document

This document is meant to provide guidance for the development of software tools to analyze and mitigate potential security risks in the JPEG File Interchange Format (JFIF) version 1.02. These guidelines come from the evaluation of the format specification, not from any vendor specific implementation or software application.

The intended audience of this document includes system engineers, designers, software developers, and testers who work on file inspection and sanitization applications that involve the JPEG Interchange and JPEG File Interchange Formats.

1.2 Introduction

The Joint Photographic Experts Group (JPEG) provides a standard (ISO 10918-1) for digital compression and coding of continuous-tone still images. For effective exchange of JPEG encoded image data, a file format must be specified. JPEG Interchange Format (JIF) was specified in Annex B of the standard and defines the basic format of a JPEG image and provides mechanisms for extension.

A number of file formats have emerged that extend the original JIF specification including JPEG File Interchange Format (JFIF) and Exchangeable Image File Format (Exif). Due to the popularity of these formats, JIF is rarely used. Practically speaking, a "JPEG file" is almost always formatted as a JFIF, Exif or a combination of these two formats.

All JFIF files are JIF compliant, since JIF allowed the extension of the format. All JIF files, however, are not JFIF compliant since not all encoding methods are supported. Since JFIF is an extension of JIF, all format segments described in this document are referred to as JFIF even if they were inherited from the JIF specification.

The JFIF specification provides room for developers to customize and extend the capabilities for specific application needs. However, this flexibility also allows a simple means to include possibly dangerous content within or appended to the image file which may be unprocessed by applications processing the image.

1.3 Background

JFIF was created by C-Cube Microsystems. It has become a de facto file standard for compressed images in the World Wide Web. ISO is currently considering the addition of JFIF as part 5 to ISO/IEC 10918.

1.3.1 Related formats

The Japan Electronics and Information Technology Industries Association (JEITA) wrote Exif for digital still cameras to add camera metadata such as shutter speed and geolocation information to images. Because Exif data is routinely added to JFIF files, this ISG will add recommendations for Exif markers.

ISO 10918-3, Annex F, defines the SPIFF file format. SPIFF includes improved lossless processing algorithms, more compression methods, as well as a directory structure. The SPIFF file structure is intended to be similar enough to JIF that JPEG decoders should be able to process SPIFF files that have been compressed using the JPEG algorithm and ignore the SPIFF marker segments and directory tree. SPIFF specific data elements will not be covered by this ISG.

JPEG 2000 (ISO/IEC 15444-1) is a more recent format by the Joint Photographers Exchange Group. JPEG 2000 does not use the DCT but instead employs wavelet transforms. Besides the name and authoring organization, JPEG and JPEG 2000 do not share any technical similarities.

JPEG compressed images are capable of being embedded into other file formats such as Portable Document Format (PDF) and Tagged Image File Format (TIFF). Only aspects of these extensions specific to JFIF will be addressed as most of these container types and extensions are beyond the scope of this document.

1.3.2 Implementations

The Windows Graphics Device Interface (GDI) was the application programming interface (API) for representing graphical objects in Windows®[1], prior to Windows XP. It was used by applications and the core operating system. In September 2004, Microsoft®[2] released a security bulletin regarding a flaw in the Windows Graphics Device Interface (GDI) library and how it decompressed JPEG image files. Processing a malformed JPEG file with a hidden payload would cause a buffer overrun and the payload to execute. Microsoft quickly released a patch to fix this exploit. The vulnerability was a combination of the GDI flaw and the malformed file. (Hornet 2004)

[1] Windows is a registered trademark of Microsoft Corp.
[2] Microsoft is a registered trademark of Microsoft Corp.

The Independent JPEG Group (IJG) distributes a free, open source and widely used development library for JPEG image compression. The IJG JPEG library does not contain any patented JPEG algorithms (i.e. arithmetic coding support). As of this writing, the current version of the library is release 8c.

1.4 Document Organization

This document describes the elements and syntax of JFIF-formatted data. It refers to these elements as constructs. In addition to describing each relevant element, this document also describes its potential flaws or ways data can be hidden, disclosed, or embedded maliciously.

The following table summarizes the organization of this document.

Table 1-1 Document Organization

Section	Description
Section 1: Scope	Background, organization, and limitations of this document.
Section 2: Constructs and Taxonomy	Definition of how the marker segments are represented as well as the terms defined in this document.
Section 3: JPEG Overview	General overview and description of the JPEG compression format.
Section 4: JPEG Interchange Format	General overview and description of the JPEG interchange format.
Section 5: JPEG File Interchange Format Structure	Marker and marker segments that appear in this document.
Section 6: Acronyms	List of acronyms used.
Section 7: Table of Document Markers and Segments	List of all document markers and segments in Section 5
Appendix A: Referenced Documents	List of documents referenced.
Appendix B: Summary of Risks	Summary of risks addressed.

1.5 Recommendations

1.5.1 Actions

The following are recommended actions for the data structures discussed in this document. They will be implemented as actions of Verification, Inspection and Sanitization (VIS) functions.

Table 1-2 Action Descriptions

Recommended Action	Description
Validate	Verify the data structure's integrity. Determine the adherence of the data to its ISO/IEC standard when possible.
Replace	Replace the entire data structure or one or more of its elements with alternate values that alleviate the risk (e.g., replacing a user name with a non-identifying, harmless value or just substituting a common name for all authors).
Remove	Remove the entire data structure or one or more of its elements. If the remove action is not viable because it would break the file format, then this must be indicated in the action.
External Filtering Required	Identify the type of data and pass the data block into an external action for handling that type of data. This data will generally be passed to another filter that is designed for that data type.
Review	Present the data structure or its constructs to review by a human. This action assumes that a system administrator or reviewer will be able to make an accept/reject decision about the data by visually reviewing it. The human review assumes that the reviewer has no knowledge of the internal structure of the data.
Reject	Discard the file entirely.

1.5.2 Naming Convention for Recommendations

Recommendations in this document are numbered sequentially, where applicable, and adhere to a standard naming convention identified by a single number x, where x is a sequential number followed by the recommendation keyword defined in Table 1-2. There may be multiple recommendations of the same type, which remain uniquely identified by its number. JFIF is the only format under review in this document.

- AR = Recommendation applies to All types - JFIF

1.6 Document Limitations

1.6.1 Markers and Segments

ISO/IEC 10918-1, Annex B, defines JIF as made up of parameters, markers, and entropy-coded data segments. Markers and marker segments are organized into images, frames, and scans with appropriate headers and processing tables as required. JFIF uses the same byte layout and is compatible with JIF.

This document will not address every detail of the JFIF definition, only the areas which are determined to present data disclosure, hiding, or attack risks.

1.6.2 Covert Channel Analysis

Covert channels are possible in JFIF. The lossy modes of JPEG encoding, based on the discrete cosine transform (DCT) function, make it possible to manipulate the DCT output values of the image data to embed a hidden message unnoticeable to the human eye. Modern steganography techniques are quite capable of manipulating digital images in such a manner. They can be difficult to identify using mathematical analysis techniques.

This document will address JFIF vulnerabilities related to data hiding and mitigation of vulnerabilities but not detection of steganography.

2. CONSTRUCTS AND TAXONOMY

2.1 Constructs

JFIF can be decomposed into numerous parts. Each part is specified as an ordered collection of parameters, markers, and entropy-coded data segments represented using byte-aligned codes. Markers can be stand alone parameters or parameters grouped together into related segments. Segments are either encoded image data (entropy coded segments) or groups of markers and parameters (marker segments).

Although this document will address the different marker segments defined in the JFIF specification, it does not serve as a complete reference to encode or decode JPEG compressed images. This document covers the particular areas which have been noted as concern for developers of file inspection and sanitization programs; however there is more complete detail in the standard that should be examined alongside this documentation.

- **Description**: A high level explanation of the data structure or element.
- **Concern**: An explanation of potential problems posed by the element. For example, some metadata elements can cause inadvertent data leakage and others can be used to exfiltrate data.
- **Location**: Provides a textual description of where to find the element in the format.
- **Examples**: If applicable, the definition will contain an example of the marker segment.
- **Recommendations**: Recommended actions as described in Section 1.5.1.

Recommendations appear within each of the JFIF marker segments. For the purposes of this document, these recommendations are "alternatives." Some recommendations may seem better than others and some recommendations may be more difficult to implement. Certain recommendations complement each other and can be grouped together (e.g., "Remove action object" and "Remove reference to action object"). Other recommendations may seem contradictory (e.g., "Remove Metadata" and "Replace Metadata").

2.2 Taxonomy

The terms that appear in this document are described in Table 2-1.

Table 2-1 Definition of Terms

Term	Definition
arithmetic coding	The means of using the arithmetic compression algorithm.
component	The elements which make up an image. The RGB colorspace has the components of R (red), G (green), and B (blue). The YCbCr colorspace has the components of Y (luminance), Cb (chrominance blue), and Cr (chrominance red). A gray scale image has only one component, Y (luminance).
continuous-tone image	An image whose components have more than one bit per sample.
differential frame	A frame in a hierarchical process in which differential components are either encoded or decoded.
discrete cosine transform (DCT)	Either the forward discrete cosine transform or the inverse discrete cosine transform.
entropy coding	A lossless data compression scheme. Huffman and arithmetic are types of entropy coding.
entropy-coded (data) segment	An independently decodable sequence of entropy encoded bytes of compressed image data.
forward discrete cosine transform	A mathematical transformation using cosine basis functions which converts a block of samples into a corresponding block of DCT coefficients.
frame	A group of one or more scans (all using the same DCT-based or lossless process) through the data of one or more of the components in an image.
frame header	A marker segment that contains a start of frame marker and associated frame parameters at the beginning of a frame.
hierarchical	A mode of operation for coding an image in which the first frame for a given component is followed by frames which code the differences between the source data and data from the previous frame for that component.
Huffman coding	A variable length coding algorithm written by David A. Huffman.
Huffman table	The set of variable length codes required in a Huffman encoder and Huffman decoder.
image data	Either source image data or decoded image data.
interchange format	The representation of compressed image data for exchange between application environments.
interleaved	The descriptive term applied to the repetitive multiplexing of small groups of data units from each component in a scan in a specific order.
inverse discrete cosine transform	A mathematical transformation using cosine basis functions which converts a block of DCT coefficients into a corresponding block of samples.
lossless	A descriptive term for encoding and decoding processes and procedures in which the output of the decoding procedure(s) is identical to the input of the encoding procedure(s).

lossy	A descriptive term for encoding and decoding processes where information is discarded; the output of the decoding procedure(s) is not identical to the input of the encoding procedure(s).
marker	A two-byte code in which the first byte is hexadecimal FF (0xFF) and the second byte is a value between 1 and hexadecimal FE (0xFE), inclusive.
marker segment	A marker and associated set of parameters.
modes (of operation)	The four main categories of image coding processes defined in ISO 10918-1 are sequential DCT-based, progressive DCT-based, hierarchical, and lossless (sequential).
parameters	Fixed length integers 4, 8 or 16 bits in length, used in the compressed data formats.
process	See coding process.
progressive (coding)	One of the DCT-based processes defined in ISO 10918-1 in which each scan typically improves the quality of the reconstructed image.
quantization	A lossy compression technique achieved by compressing a range of values to a single quantum value.
reconstructed image	The resulting image of a decoding process of encoded image data.
sample	One value of a two-dimensional component of an image.
scan	A single pass through data of one or more components in an image.
scan header	A marker segment that contains a start of scan marker and associated scan parameters at the beginning of a scan.
sequential (coding)	One of the lossless or DCT-based coding processes in which each component of the image is encoded within a single scan.
spectral selection	A method of grouping scans in the progressive mode of operation.
successive approximation	A method of grouping scans in the progressive mode of operation.
table specification data	The coded representation of tables used in the encoding and decoding processes.

Some excerpts from ISO 10918-1, 3.1 Definitions and abbreviations.

3. JPEG OVERVIEW

JPEG compression is a combination of different image and data compression techniques. The basis for these techniques takes advantage of how the human eye perceives color and brightness and discards less noticeable information about the image. The JPEG standard specifies two classes of compression (lossy and lossless) and four modes of operation (sequential DCT-based, progressive DCT-based, lossless sequential, and hierarchical). Additionally two entropy coding methods are stated, Huffman and arithmetic. The sequential DCT-based mode of operation and Huffman encoding make up the baseline sequential JPEG encoding process. Every JPEG decoder should be able to decompress the baseline sequential JPEG process.

This document will focus on the resulting file format after compressing the image data and not necessarily the encoding or decoding processes. However, each of the processing mode's compressed image data structure is formatted differently.

3.1 Sequential Encoding Process

The sequential DCT-based mode of operation uses a single frame and makes one pass (scan) through the image data, processing the image data units in raster order (from top left to bottom right). If a scan is made up of more than one component the encoder will alternate the data units of each component, grouping the data units of each component into an interleaved minimum coded unit (MCU).

From a security standpoint, sequential encoding is preferred over progressive or hierarchical because it contains a single frame and a single scan and therefore is less likely to have hidden data.

3.2 Progressive Encoding Process

The progressive DCT-based mode of operation is much like the sequential mode of operation with the main difference being multiple scans within the frame. Each scan in a progressive frame contains a portion of the quantized DCT coefficients. After the image data units go through the DCT and are quantized, the resulting coefficients are separated into multiple scans via two possible methods; spectral selection and successive approximation.

The decoding of a progressive encoded image will process the scans containing more general image data first to display a blurry image quickly which gradually comes into focus as the scans with more detailed image data are processed.

Since lower resolution scans may be skipped by the decoder for local images or high bandwidth connections, there is a risk that hidden data is contained in the skipped scans.

3.3 Lossless Encoding Process

The lossless mode of operation is a sequential process (one frame with one scan in raster order). Lossless sequential encoding does not use the DCT function and does not compress as efficiently as the DCT-based encoding processes. Because of this, lossless JPEG encoding is not nearly as popular as the JPEG baseline. The data unit for this process is one sample.

3.4 Hierarchical Encoding Process

The hierarchical mode of operation encodes the image in a sequence of two or more frames. Each frame can be an independent image or the difference of a preceding frame (differential). A differential frame contains image data which refers to the preceding frame's image data. This provides a gradual display of the image as each frame is decoded similar to the progressive encoding process but also allows for different resolutions of the image in each frame.

The multi-resolution capability of hierarchical encoding is commonly related to a pyramid, where the decoding of each frame reveals a progressively larger image. One example where hierarchical encoding may be advantageous would be to employ a high resolution frame used for printing and a lower resolution frame to be viewed in an application on a monitor screen. Each frame may use one of the different encoding processes previously mentioned (sequential, progressive or lossless).

Because independent images are allowed in frames, there is a high risk of data hiding in hierarchical images.

4. JPEG INTERCHANGE FORMAT

The JIF format was defined in Annex B of the JPEG standard (ISO 10918-1) and is the basis of the JFIF format. The typical structure of a JIF file, as illustrated in Figure 4-1, is enclosed between a Start of Image (SOI) marker and an End of Image (EOI) marker. The SOI and EOI segments consist of the markers only and do not specify a length.

SOI	Frame	EOI

Figure 4-1 Compressed image data for sequential DCT-based, progressive DCT-based, and lossless modes of operation

Between the **image markers** are one or more frames, depending on the mode of operation used to encode the original image data. The sequential (DCT-based and lossless) and progressive coding processes have only one frame. The hierarchical coding process may contain multiple frames which must be preceded by a Define Hierarchical Progression (DHP) marker segment, see Figure 4-2. In the hierarchical coding process, differential frames may be used where the differential frame references image data from the previous frame.

SOI	[Tables/misc.]	DHP	$Frame_1$	[…]	[$Frame_n$]	EOI

Figure 4-2 Compressed image data for hierarchical mode of operation

A **frame** contains a frame header marker segment, see Figure 4-4, where the Start of Frame (SOF) marker is one or more scan marker segments and an optional Define Number of Lines (DNL) marker segment which defines or redefines the image height within the frame. The frame header marker segment may be preceded by other table specification or miscellaneous marker segments, as shown in Figure 4-3.

[Tables/misc.]	Frame header	$Scan_1$	[DNL]	[$Scan_2$]	[…]	[$Scan_n$]

Figure 4-3 JIF frame

SOF	Header length	Parameters

Figure 4-4 Frame header marker segment

Different SOF markers are defined and identify the encoding process and entropy encoding procedure used to process the encoded image data within the frame. See ISO 10918-1 Table B.1, Marker code assignments, for a comprehensive list.

A **scan** contains a scan header marker segment, Figure 4-6, which includes the Start of Scan (SOS) marker, and Entropy Coded Segments (ECS). A scan header marker segment may be preceded by other table specification or miscellaneous marker segments. More than one encoded segment may exist depending on the presence of a

define restart interval (DRI) marker segment and are separated by one or more accompanying Restart (RST) markers.

[Tables/misc.]	Scan Header	ECS	[RST_0]	…	[ECS_{n-1}]	[RST_{n-1}]	[ECS_n]

Figure 4-5 JIF scan

SOS	Header length	Parameters

Figure 4-6 Scan header marker segment

Possible **table specification** marker segments include:

- Define Quantization Table (DQT) contains the table used to quantize the 8x8 coefficient matrices.
- Define Huffman Table (DHT) contains the Huffman code map.
- Define Arithmetic Conditioning (DAC) table contains the arithmetic code map.

Optional **miscellaneous** marker segments include:

- Define Number of Lines (DNL) defines or redefines the image height in pixels.
- Define Restart Interval (DRI) defines how many RSTn markers will exist between the ECSs, if any. The RSTn markers are used to detect the loss of data during transmission. The RSTn markers start at RST0 and increment to RST7 before restarting again at RST0. The RSTn segments consist of the markers only and do not specify a length.
- Application data (APPn) contain custom information applicable to specific applications. There are sixteen application marker segments (APP0–APP15) defined in ISO 10918-1 but no definition in regards to the structure of data stored within the marker segments. In addition, there can be multiple segments of each APPn type. For example, there may be multiple APP2 marker segments in a file.
- Comments (COM) are a null terminated string of characters associated with the encoded image. The character encoding to be used is unspecified.

The hierarchical coding process may also use an Expanded Reference Component (EXP) marker segment if it is required to expand the components of a differential frame and will only be present preceding a frame header marker segment. Except for the SOI, EOI, and RSTn segments, the two bytes following a marker define the length of that segment (including the two bytes used to store the length). Valid length values are listed in Table 5-2.

4.1 JPEG File Interchange Format

In order to exchange a JPEG encoded image between differing architectures and applications, the JPEG File Interchange Format (JFIF) further constrains JIF specifications. JFIF specifies a standard colorspace, YCbCr (defined by CCIR 601), and recommends using the JPEG baseline process to ensure compatibility with all ISO 10918-2 compliant decoders.

An APP0 marker segment is required immediately following the SOI marker to identify a JFIF file as well as provide the JFIF version, the pixel density, the measurement unit used (inches or centimeters), and an optional uncompressed 24 bit Red-Green-Blue (RGB) thumbnail image, see Figure 4-7. The JFIF APP0 marker segment is identified by the string "JFIF" immediately following the segment length parameter.

APP0	Length	JFIF\0	Version	Units	XDensity	YDensity	Thumbnail data

Figure 4-7 JFIF APP0 Marker Segment

Also defined in JFIF v1.02 is an optional JFIF extension APP0 marker segment. This segment immediately follows the JFIF APP0 marker segment and is identified by the string "JFXX" immediately following the segment length parameter, which contains additional thumbnail options; a JPEG encoded thumbnail (JIF), a one byte per pixel thumbnail with an RGB color palette, and an RGB thumbnail using three bytes per pixel.

APP0	Length	JFXX\0	Extension Code	Extension Data (Thumbnail Data)

Figure 4-8 JFIF extension APP0 Marker Segment

Other APP0 marker segments may be included but must not be identified as JFIF or a JFIF extension (JFXX) APP0 marker segment. In addition, subsequent APP0 marker segments must be located after the required JFIF APP0 marker segments.

4.2 Exchangeable image file format (EXIF)

The exchangeable image file format (Exif) extends the JPEG interchange format (JIF) by defining the structure of metadata inside APP1 and APP2 marker segments. Exif also limits the compression used to the JPEG baseline (Sequential processing with Huffman encoding). Practically all digital cameras add Exif information to images related to camera parameters used to take the picture, such as shutter speed. The data structure

located within the APP1 marker segment follows the Tagged Image File Format (TIFF) convention. The data structure inside the APP2 marker segments are for optional Flashpix extensions. Exif requires that its APP1 marker immediately follow the SOI marker, while JFIF requires the same, therefore these formats are technically incompatible. Many implementations embed both, with one or the other as the first marker after the SOI.

An APP1 marker segment identifies an Exif file and contains a TIFF header and metadata. The Exif identifier is the "Exif" followed by two 0 bytes of padding. After the Exif identifier is a TIFF header and Image File Directory (IFD) which is made up of attribute value pairs. Some of these attribute value pairs include the Exif version, image dimensions and orientation, colorspace profiles, thumbnail image data, global positioning system (GPS) information, and recording equipment details and settings. For more on TIFF and IFD, see TIFF Rev. 6.0 (ISO 12639). See JEITA CP-3451 for more detailed information about the Exif specification.

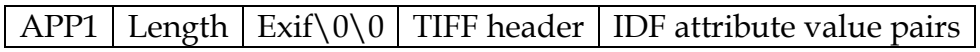


Figure 4-9 Exif APP1 Marker Segment

APP2 marker segments, identified by the string "FPXR", may follow the APP1 marker segment, containing FlashPix extensions. More than one APP2 marker segment may be used. APP2 FlashPix marker segments are included in the Exif specification but are not required. For more information about FlashPix, see FlashPix Format Specification version 1.0 (Eastman Kodak Company). FlashPix is out of scope for this ISG.

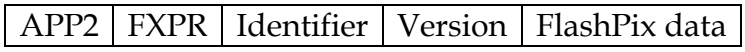


Figure 4-10 Exif APP2 FlashPix Marker Segment

5. JPEG FILE INTERCHANGE FORMAT STRUCTURE

JFIF.5.1: MARKER VALUES AND ORDER

DESCRIPTION:

Markers are identified by the value 0xFFxx, where xx is a hexadecimal value which specifies the marker or marker segment type. The two exceptions are the values 0xFF00 and 0xFFFF. 0xFF00 is a special value in regards to the entropy coded image data. If the encoded data contains the value 0xFF, zeros are added as padding in order to avoid misinterpretation as a marker in the encoded image data. The value 0xFFFF is ambiguous and undefined as multiple possibilities for interpretation exist. Table 5-1 lists the defined JPEG markers.

Table 5-1 JFIF markers

Marker	Hex Value	Description	Comments
TEM	0xFF01	For temporary private use in arithmetic coding	Miscellaneous, dependent, may exist only if arithmetic coding frame is present
RES	0xFF02 – 0xFFBF	Reserved	Miscellaneous, unspecified in ISO/IEC 10918-1
SOF_0	0xFFC0	Start of baseline DCT with Huffman coding frame	Start of Frame marker
SOF_1	0xFFC1	Start of extended sequential DCT with Huffman coding frame	Start of Frame marker
SOF_2	0xFFC2	Start of progressive DCT with Huffman coding frame	Start of Frame marker
SOF_3	0xFFC3	Start of lossless (sequential) with Huffman coding frame	Start of Frame marker
DHT	0xFFC4	Define Huffman table(s)	Table
SOF_5	0xFFC5	Start of differential sequential DCT with Huffman coding frame	Start of Frame marker
SOF_6	0xFFC6	Start of differential progressive DCT with Huffman coding frame	Start of Frame marker
SOF_7	0xFFC7	Start of differential lossless (sequential) with Huffman coding frame	Start of Frame marker
JPG	0xFFC8	Reserved for JPEG extensions	Miscellaneous, unspecified in ISO/IEC 10918-1
SOF_9	0xFFC9	Start of extended sequential DCT with arithmetic coding frame	Start of Frame marker
SOF_{10}	0xFFCA	Start of progressive DCT with arithmetic coding frame	Start of Frame marker
SOF_{11}	0xFFCB	Start of lossless (sequential) with arithmetic coding frame	Start of Frame marker
DAC	0xFFCC	Define arithmetic conditioning(s)	Dependent, may exist only if an arithmetic coding frame is present
SOF_{13}	0xFFCD	Start of differential sequential DCT with arithmetic coding frame	Dependent frame, may only exist if DHP marker segment is present
SOF_{14}	0xFFCE	Start of differential progressive DCT with arithmetic coding frame	Dependent frame, may only exist if DHP marker segment is present

SOF_{15}	0xFFCF	Start of differential lossless (sequential) with arithmetic coding frame	Dependent frame, may only exist if DHP marker segment is present
RSTm	0xFFD0 – 0xFFD7	Restart with modulo 8 count “m”	Dependent, will exist only if DRI marker segment is present
SOI	0xFFD8	Start of image	Image boundary
EOI	0xFFD9	End of image	Image boundary
SOS	0xFFDA	Start of scan	Scan
DQT	0xFFDB	Define quantization table(s)	Table
DNL	0xFFDC	Define number of lines	Optional
DRI	0xFFDD	Define restart interval	Miscellaneous
DHP	0xFFDE	Define hierarchical progression	Required for differential frames
EXP	0xFFDF	Expand reference component(s)	Dependent, will only exist if DHP marker segment is present
APPn	0xFFE0 – 0xFFEF	Reserved for application segments	Miscellaneous
JPGn	0xFFF7 – 0xFFFD	Reserved for JPEG extensions	Miscellaneous, unspecified in ISO/IEC 10918-1
COM	0xFFFE	Comment	Miscellaneous

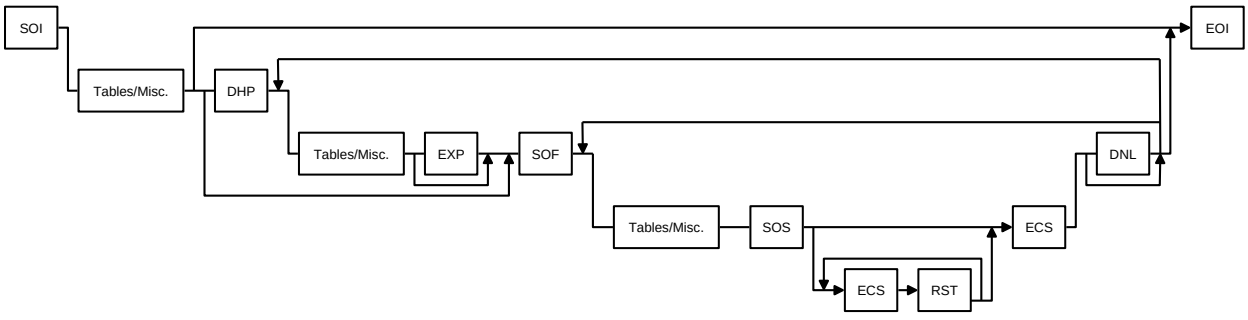
CONCERN:

An invalid marker indicates that the image file is malformed or corrupted and may indicate a data attack by causing the processing application to hang or crash.

PRODUCT: JFIF

LOCATION:

Markers are located at the beginning of every marker segment and also at the beginning and ending of the JPEG image format. Markers should be located in a specific order. For the appropriate order of marker locations, see Figure 5-1.



Abbreviated form of ISO/IEC 10918-1, Figure B.16

Figure 5-1 Order of markers

RECOMMENDATION:

AR.1 Validate: Validate that the hexadecimal value 0xFF00 is not present outside of a marker segment or an entropy coded segment.

AR.2 Validate: Validate that the hexadecimal value 0xFFFF is not present outside of a marker segment.

AR.3 Validate: Validate that all marker locations are in the correct order according to figure 5-1.

AR.4 Remove: N/A

AR.5 Replace: N/A

AR.6 External Filtering Required: N/A

AR.7 Review: N/A

JFIF.5.1: END

JFIF.5.2: IMAGE BOUNDARIES

DESCRIPTION:

The SOI marker indicates the beginning of an encoded image. It is paired with an EOI marker concluding the encoded image. The markers and marker segments between SOI and EOI contain the encoded image data and meta data necessary for processing.

CONCERN:

The length between SOI and EOI markers is unconstrained; therefore it is possible for extra bytes to exist between the various markers and marker segments that make up a valid JPEG image. Data hiding is possible by storing arbitrary data in between markers and marker segments. A data attack is possible if this hidden data is active or malicious in nature. Additionally any data after the EOI is typically ignored by decoders and represents hidden data.

PRODUCT: JFIF

LOCATION:

The SOI marker is the first marker at the beginning of a JPEG image. The EOI marker is the last marker of a JPEG image.

RECOMMENDATION:

AR.1 Validate: Validate that the SOI marker is the first byte values of the file.

AR.2 Validate: Validate that the EOI marker is the last byte values of the file.

AR.3 Validate: Validate that there is only one SOI marker and one EOI marker.

AR.4 Validate: Validate that every marker or marker segment boundary is directly adjacent to another marker or marker segment between the SOI and EOI markers.

AR.5 Remove: Remove any data which may exist before the SOI marker.

AR.6 Remove: Remove any data which may exist after the EOI marker.

AR.7 Replace: Rebuild the JFIF structure such that every marker or marker segment is directly adjacent to another marker or marker segment, removing the possibility for extra bytes to exist between marker segments.

AR.8 External Filtering Required: N/A

AR.9 Review: N/A

REFERENCE:

http://old.marcofolio.net/how_to/hide_files_in_jpg_files.html

http://www.online-tech-tips.com/computer-tips/hide-file-in-picture

JFIF.5.2: END

JFIF.5.3: MARKER SEGMENT LENGTH

DESCRIPTION:

A marker segment's marker type is followed by a length parameter of that segment's content size in bytes. The length includes the size of itself (two bytes) and the rest of the segment but does not include the size of the marker type preceding it. Some older JPEG image libraries have been found to process this length without any sanity checking to make sure the length is an appropriate value. For possible length values, see Table 5.2.

Table 5-2 Appropriate marker segment lengths

Marker Segment	Appropriate Length Values (Bytes)
SOFn	8 – 65,535
DHT	2 – 65,535
DAC	2 – 65,535
SOS	6 – 65,534
DQT	2 – 65,535
DNL	4
DRI	4
DHP	8 – 65,535
EXP	3
APPn	2 – 65,535
COM	2 – 65,535

CONCERN:

If the length specified is an invalid value (less than two or past the EOI) processing this marker segment is vulnerable to a data attack, possibly redirecting the processing application to an arbitrary address in memory resulting in a system crash or buffer overflow.

PRODUCT: JFIF

LOCATION:

A marker segment's length parameter follows directly after a marker segment's marker parameter.

RECOMMENDATION:

AR.1 Validate: Validate that every marker segment length is an appropriate value, as referenced in Table 5-2, and does not extend past the EOI marker.

AR.2 Remove: N/A.

AR.3 Replace: N/A

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

JFIF.5.3: END

JFIF.5.4: FRAMES

DESCRIPTION:
A frame consists of a frame header and contains one or more scans. Frames for every supported JPEG processing mode and entropy coding combination are defined in the specification Most JPEG images will contain only one frame. An image processed with the hierarchical processing mode will have multiple frames all preceded by a DHP marker segment.

CONCERN:
An image with more than one frame may be at risk of data hiding. In the hierarchical process, differential frames use other frames as a reference for processing. Non-differential frames are stand alone images. If more than one non-differential frame is present, data hiding may exist as unseen and unrelated images in a single image file. Additionally, hierarchical images are not well supported in open source libraries or commercial tools.

PRODUCT: JFIF

LOCATION:
Frames follow after the SOI marker and must come before an SOS marker segment.

RECOMMENDATION:

AR.1 Validate: Validate that only one non-differential frame exists.

AR.2 Validate: Validate that differential frames are present only if a DHP marker segment exists.

AR.3 Validate: Validate that the first differential frame is preceded by a DHP marker segment.

AR.4 Remove: N/A

AR.5 Replace: Replace the hierarchical image with a restructured, baseline JFIF image.

AR.6 External Filtering Required: N/A

AR.7 Review: Successively remove each differential frame, create an image from it, and present the resulting image for human review.

AR.8 Reject: As hierarchical images are not common, are not well supported, and represent a data hiding risk they may be rejected entirely.

JFIF.5.4: END

JFIF.5.5: FRAME HEADER MARKER SEGMENTS

DESCRIPTION:
Every frame has a frame header marker segment. A frame header marker segment contains parameters about the frame; image dimensions, sample precision, number of components, component sampling factors, and a quantization table selector for each component. The frame marker identifies the encoding process (sequential, progressive, lossless, or hierarchical) used and which entropy encoding procedure was used (Huffman or Arithmetic).

SOFn	Segment length	Sample precision	Number of lines	Samples per line	Number of components	Component parameters	…	[Component parameters]

Figure 5-2 Frame header parameters

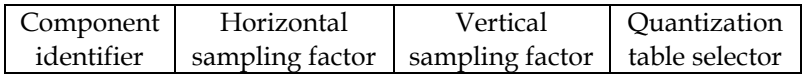


Figure 5-3 Frame header component parameters

CONCERN:
A malformed frame header, missing component parameters for example, may cause a JPEG decoder to crash or malicious code to execute and is considered a data attack risk. If there are more lines in the decompressed image than *Number of lines* viewers will truncate the image data shown. This represents a data hiding risk if there is additional image data not seen by the user or human reviewer.

PRODUCT: JFIF

LOCATION:
A frame header marker segment signifies the beginning of a frame within the image boundary markers.

RECOMMENDATION:

AR.1 Validate: Validate that the *Number of components* parameter matches how many component parameters are present.

AR.2 Validate: Validate that the *Number of lines* parameter matches how many lines there are within a decompressed image. The image must be decompressed to validate *number of lines*.

AR.3 Remove: Remove any extraneous bytes exceeding the *Number of lines* parameter.

AR.4 Replace: N/A

AR.5 External Filtering Required: N/A

AR.6 Review: N/A

JFIF.5.5: END

JFIF.5.6: SCANS

DESCRIPTION:
A scan consists of a scan header and contains one or more entropy coded segments of image data.

CONCERN:
Sequential processed frames will contain only one scan. Progressive processed frames will contain more than one scan. In progressive frames, the decoder may skip low resolution scans and only display the high resolution image if the image is stored locally or is loaded over a high speed connection. The presence of multiple scans is a data hiding risk. Converting an image with multiple scans to a baseline image does not sufficiently mitigate the data hiding issue since the scan selected to create the baseline image could be the scan containing the hidden data. Human review is necessary to compare the resulting baseline image with the original image to determine if the selected scan contained hidden data.

PRODUCT: JFIF

LOCATION:
A scan must follow an SOF marker.

RECOMMENDATION:

AR.1 Validate: Validate that the image contains only one scan.

AR.2 Remove: N/A

AR.3 Replace: Replace the image with a baseline JFIF image containing only one scan and present the resulting baseline image for human review.

AR.4 External Filtering Required: N/A

AR.5 Review: Replace the image with a baseline JFIF image containing only one scan and present the resulting baseline JFIF image for human review.

JFIF.5.6: END

JFIF.5.7: SCAN HEADER MARKER SEGMENTS

DESCRIPTION:
Every scan must have one header. A scan header marker segment contains parameters about the scan such as entropy coding settings and spectral selection or successive approximation settings required for decoding the scan.

SOS	Segment length	Number of components	Component parameters	Spectral selection parameters	Successive approximation parameters

Figure 5-4 Scan header parameters

Component identifier	Horizontal sampling factor	Vertical sampling factor	Quantization table selector

Figure 5-5 Scan header component parameters

Start of spectral or predictor selection	End of spectral selection

Figure 5-6 Scan header spectral selection parameters

Successive approximation bit position high	Successive approximation bit position low or point transform

Figure 5-7 Scan header successive approximation parameters

CONCERN:

A malformed scan header that may cause a JPEG decoder to crash and malicious code to execute is considered a data attack.

PRODUCT: JFIF

LOCATION:
A scan header marker segment signifies the beginning of a scan following a frame header marker segment. More than one scan will be present if progressive DCT processing was used.

RECOMMENDATION:

AR.1 Validate: Validate that the parameters of the scan header correspond to the specification (Section B.2.3).

AR.2 Validate: Validate that the *Number of components* parameter matches how many component parameters are present.

AR.3 Remove: N/A

AR.4 Replace: N/A

AR.5 External Filtering Required: N/A

AR.6 Review: N/A

JFIF.5.7: END

JFIF.5.8: TABLES

DESCRIPTION:
Marker segments that contain quantization, Huffman coding tables, or arithmetic conditioning tables (DQT, DHT, and DAC respectively) can be placed in various positions in the JPEG image's data structure. These tables are loaded into memory at the table destination identifier included in the define marker (DQT, DHT, or DAC) before being used to decode an image. Table destination identifiers may be reused, overwriting the previous table.

CONCERN:
Tables defined in a DQT, DHT, or DAC segment, but never referenced by a SOF or SOS segment are a data hiding risk.

PRODUCT: JFIF

LOCATION:
A table is usually located before the frame header or scan header with which it will be used.

RECOMMENDATION:

AR.1 Validate: Validate that there are no unreferenced marker segments.

AR.2 Remove: Remove unreferenced table marker segments.

AR.3 Replace: N/A

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

JFIF.5.8: END

JFIF.5.9: APPLICATION MARKER SEGMENTS

DESCRIPTION:

Application (APP0 – APP15) marker segments give the JIF specification flexibility for application developers to customize the data format and can contain any type of data. Some common uses of application marker segments are JFIF and JFXX extensions in APP0 marker segments as well as Exif data in APP1 marker segments. It is up to the application developer to decide how and which application marker is used. Specific application markers are discussed in detail in the following sections.

CONCERN:

Application developers can decide how to utilize an application marker segment. Any type of data may be placed in an application marker segment allowing for data hiding, data disclosure, and data attack concerns.

PRODUCT: JFIF

LOCATION:

Application marker segments can be anywhere within the image boundaries after the SOI marker and before the EOI marker. JFIF requires that its application marker directly follows the SOI marker.

RECOMMENDATION:

AR.1 Validate: N/A

AR.2 Remove: Remove all application marker segments.

AR.3 Remove: Remove all application marker segments that are not on an approved application marker segment list.

AR.4 Replace: N/A

AR.5 External Filtering Required: Identifiable data types within application marker segments, such as text strings, a JFIF thumbnail image, or Exif metadata may be externally filtered.

AR.6 Review: N/A

JFIF.5.9: END

JFIF.5.10: JFIF APP0 MARKER SEGMENT

DESCRIPTION:

An APP0 marker segment is required to identify the file as JFIF. The JFIF APP0 marker segment contains the zero terminated string “JFIF”. The 2 bytes following the JFIF identifier is the version. The first byte is the major version and the second byte is the minor. The current version of JFIF is 1.02 and therefore the bytes are 0x01 0x02. The next byte is the density units, 0, 1, and 2 are valid. The next 4 bytes are X and Y density followed by the 2 bytes for thumbnail width and height and finally the thumbnail data.

CONCERN:

A buffer overflow data attack may occur if “JFIF” is a non-zero terminated string. Additionally, unknown versions may contain different or more data that is unprocessed by applications. The

thumbnail data is uncompressed RGB data and therefore must be 3 * width * height or data may be hidden.

PRODUCT: JFIF

LOCATION:
The JFIF APP0 marker segment must immediately follow the SOI marker.

RECOMMENDATION:

AR.1 Validate: Validate that the "JFIF" string is a zero terminated string.

AR.2 Validate: Validate that the JFIF version is 1.02.

AR.3 Validate: Validate that the density unit is only 0, 1, or 2.

AR.4 Validate: Validate that the thumbnail data is exactly 3 * width * height.

AR.5 Remove: Remove the JFIF APP0 marker segment.

AR.6 Replace: Replace the byte following the "JFIF" string with the zero terminator.

AR.7 Replace: Replace the density unit with 0 if it is not 0, 1, or 2.

AR.8 External Filtering Required: N/A

AR.9 Review: N/A

JFIF.5.10: END

JFIF.5.11: JFIF APP0 THUMBNAIL IMAGE

DESCRIPTION:
JFIF supports a thumbnail image stored in the JFIF APP0 marker segment. The thumbnail is not compressed or encoded. The entire APP0 segment cannot exceed 65535 bytes. The thumbnail cannot exceed 65517 bytes.

CONCERN:
The image of the file and the thumbnail are not required to be of the same image. Data disclosure or data hiding are risks if the encoded image and thumbnail image are different. If the thumbnail is preserved from the original image the filter must present it for human review to ensure it is the same as the main image.

PRODUCT: JFIF

LOCATION:
Following the SOI marker, in a JFIF APP0 marker segment.

RECOMMENDATION:

AR.1 Validate: N/A

AR.2 Remove: Remove the JFIF thumbnail block and recalculate the APP0 segment length.

AR.3 Replace: Create a new thumbnail from the image data and recalculate the APP0 segment length.

AR.4 External Filtering Required: N/A

AR.5 Review: Present the thumbnail for human inspection.

JFIF.5.11: END

JFIF.5.12: JFIF EXTENSION APP0 MARKER SEGMENT

DESCRIPTION:

The JFIF extension marker allows the thumbnail to be encoded. It is always used in conjunction with a JFIF APP0 marker. It must always follow the JFIF APP0 marker segment and must contain the zero terminated string "JFXX". The 2 bytes following the JFXX identifier is the extension code. The extension code is 1 byte long and can be one of the following: 0x10 for a thumbnail coded using JPEG, 0x11 for a thumbnail stored using 1byte/pixel, and 0x13 for a thumbnail stored using 3 bytes/pixel. The next block is the extension (thumbnail) data. If the extension code is 0x10, the thumbnail is compressed using JPEG and conforms to the syntax of JIF, but must not contain JFIF or JFXX marker segments.

CONCERN:

A buffer overflow data attack may occur if “JFXX” is a non-zero terminated string. An unknown extension type would allow a block of data not processed by viewers, representing a data hiding risk.

PRODUCT: JFIF

LOCATION:

The JFXX APP0 marker segment must immediately follow the JFIF APP0 marker.

RECOMMENDATION:

AR.1 Validate: Validate that the “JFXX” string is a zero terminated string.

AR.2 Validate: Validate that the extension code is one of: 0x10, 0x11, or 0x13.

AR.3 Remove: Remove the JFIF extension segment if the extension code is not of a known type.

AR.4 Remove: Remove the JFIF extension segment.

AR.5 Replace: Replace the byte following the “JFXX” string with the zero terminator.

AR.6 External Filtering Required: N/A

AR.7 Review: N/A

JFIF.5.12: END

JFIF.5.13: JFIF EXTENSION THUMBNAIL IMAGE

DESCRIPTION:

JFIF supports a thumbnail image stored in an optional JFIF extension APP0 marker segment. The thumbnail may be stored using JPEG encoding, one byte per pixel and a color palette, or 3 bytes

per pixel, as defined in the JFIF extension segment.

CONCERN:
The image of the file and the thumbnail are not required to be of the same image. Data disclosure or data hiding are risks if the encoded image and thumbnail image are different. Additionally, there may be thumbnail data in both the JFIF and JFXX markers and applications may use either or neither of them. If the thumbnail is preserved from the original image the filter must present it for human review to ensure it is the same as the main image.

PRODUCT: JFIF

LOCATION:
Following the SOI marker and the JFIF APP0 marker, in a JFIF extension APP0 marker segment.

RECOMMENDATION:

AR.1 Validate: Validate that the thumbnail image has no JFIF markers present.

AR.2 Remove: Remove the JFIF extension block (removing the thumbnail).

AR.3 Replace: Create a new thumbnail from the image data and recalculate the APP0 segment length.

AR.4 External Filtering Required: N/A

AR.5 Review: Present the thumbnail for human inspection.

JFIF.5.13: END

JFIF.5.14: NXPowerLite APP MARKER

DESCRIPTION:
NXPowerLite is a Microsoft Office®[3], PDF, and image compression application heavily used by the DoD and military to compress documents prior to sending them across slow network connections. NXPowerLite adds an APP0 marker to the image when it compresses a JPEG image. The NXPowerLite marker may appear anywhere in the image aside from before the SOI, after the EOI, or as the first APP marker after the SOI. The current marker is APP0 followed by two bytes for the length, which is 8, followed by the 0 terminated string NXPL, and finally one byte.

CONCERN:
As with all APP markers there is potential for data hiding, data disclosure, and data attack. The NXPowerLite APP marker currently only includes an integer; however, there is no guarantee that the software will not expand the field in the future.

PRODUCT: NXPOWERLITE

LOCATION:
Following the SOI marker and the JFIF APP0 marker, in a JFIF extension APP0 marker segment.

RECOMMENDATION:

[3] Microsoft Office is a registered trademark of Microsoft Corp.

AR.1 Validate: Validate that the NXPL APP marker appears only in an appropriate place.

AR.2 Validate: Validate that the NXPL APP marker length is 8.

AR.3 Validate: Validate that the NXPL APP marker identifier is 0 terminated.

AR.4 Validate: Validate that only one NXPL APP marker is present in the file.

AR.5 Remove: Remove the NXPL APP marker segment.

AR.6 Replace: N/A

AR.7 External Filtering Required: N/A

AR.8 Review: N/A

JFIF.5.14: END

JFIF.5.15: ICC (International Color Consortium) Profile

DESCRIPTION:

An ICC profile is a data format for describing the color attributes of any device that captures or displays color [7]. Essentially, an image's ICC profile determines its color scheme. ICC profiles are used across a variety of data formats such as Bitmap, JPEG, and PDF. Within JFIF, the ICC profile is found in an APP2 marker. The marker must use the 0 terminated tag ICC_PROFILE

CONCERNS:

An ICC profile is an embedded object adhering to an independent data format. As such, its data disclosure, hiding, and attack risks must be subjected to the same level of scrutiny as the JFIF file that contains it. As an independent format there are risks of data attack and data hiding.

PRODUCT: JFIF

LOCATION:

After the Exif or JFIF APP marker.

RECOMMENDATIONS:

AR.1 Validate: Verify the ICC profile against the ICC format specification.

AR.2 Remove: Remove the ICC profile. This may result in a color scheme that renders the image unviewable.

AR.3 Replace: Replace the ICC profile with a 4-byte enumerated color space (ECS) field and set the Method value to 1. This may result in a color scheme that renders the image unviewable.

AR.4 External Filtering Required: Pass the contents of the ICC Profile field to a filter capable of handling ICC profiles.

AR.5 Review: N/A

REFERENCE:

See ISO 15444 Part 1, Annex I.5.3.3. Also, see this ISG, Chapter 4.3.2.

JFIF.5.15: END

JFIF.5.16: Exif APP MARKER

DESCRIPTION:

An APP1 marker is required to identify a file as an Exif. Exif is used primarily by camera manufacturers to add relevant data to the image. This information can include the camera brand, model, shutter speed, aperture, geolocation data and so on. The only limit to the data is the 64k length limit for APP markers imposed by JIF. The data is encoded as a TIFF and can include a thumbnail. The Exif standard requires that its APP1 marker be present immediately after the SOI, making it incompatible with JFIF, which has the same requirement. However, many Exif files have both JFIF and Exif APP markers. An Exif APP marker is APP1 followed by the length, followed by the 0 terminated string Exif followed by one byte of 0 padding followed by TIFF data.

CONCERN:

Exif markers can add any kind of data that TIFF supports making it a high risk for data hiding and data disclosure. Additionally, since a processing application must use a TIFF parser to parse the contents, an attacker could utilize a vulnerability in the TIFF parser. The TIFF may also include a thumbnail. If an image has both JFIF and Exif thumbnails, it is not defined which will be shown by an image viewer. Because of the high risks associated with Exif the standard recommendation is to remove it entirely, however, there may be circumstances where some data is required, such as geolocation information. In these cases the Exif data must be externally filtered by a TIFF filter.

PRODUCT: EXIF

LOCATION:

Following the SOI marker.

RECOMMENDATION:

AR.1 Validate: Validate that only one Exif APP marker is present.

AR.2 Validate: Validate that either a JFIF APP marker or an Exif APP marker is present, but not both.

AR.3 Validate: Validate that the Exif APP marker identifier is 0 terminated with an additional 0 byte of padding.

AR.4 Remove: Remove the Exif APP marker segment.

AR.5 Replace: N/A

AR.6 External Filtering Required: Filter the TIFF data in a TIFF filter.

AR.7 Review: Present the thumbnail for human inspection.

JFIF.5.16: END

JFIF.5.17: DEPENDENT MARKER SEGMENTS

DESCRIPTION:

Some marker segments are dependent upon and support other marker segments. Such dependent

marker segments include Define Arithmetic Conditioning (DAC), Expanded Reference Components (EXP), Restart (RST), Temporary private arithmetic coding use (TEM), and differential frame headers.

CONCERN:
The presence of a dependent marker segment without the marker segment which is depended upon may indicate data hiding or a corrupted JPEG image.

PRODUCT: JFIF

LOCATION:
Dependent marker segments have various positions between the image boundaries depending upon the marker segment.

RECOMMENDATION:

AR.1 Validate: Validate that an Arithmetic coding frame is present if a TEM marker segment exists.

AR.2 Validate: Validate that a DHP marker segment is present if a differential frame header exists.

AR.3 Validate: Validate that a DRI marker segment is present if an RST marker segment exists.

AR.4 Validate: Validate that a DHP marker segment is present if an EXP marker segment exists.

AR.5 Remove: N/A

AR.6 Replace: N/A

AR.7 External Filtering Required: N/A

AR.8 Review: N/A

JFIF.5.17: END

JFIF.5.18: RESERVED MARKER SEGMENTS

DESCRIPTION:
The way a reserved marker segment (RES, JPG, JPGn) is structured has not been specified in ISO/IEC 10918-1. These reserved market segments are for future revisions to the JPEG specification.

CONCERN:
The presence of a reserved marker segment may indicate data hiding.

PRODUCT: JFIF

LOCATION:
Reserved marker segments are located where miscellaneous or table marker segments are found.

RECOMMENDATION:

AR.1 Validate: Validate that no reserved marker segments exist.

AR.2 Remove: Remove any reserved marker and any data up to the next valid marker.

AR.3 Replace: N/A

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

JFIF.5.18: END

JFIF.5.19: ENTROPY CODED IMAGE DATA SEGMENTS

DESCRIPTION:

Processed image data are stored in entropy coded segments (ECS). Image data go through different states when processed by a DCT-based encoder; raw image data, DCT coefficients, quantized DCT coefficients, and entropy coded data. Raw image data is what is usually presented for human viewing. Entropy coded data is the format usually stored to disk.

CONCERN:

Known steganography tools manipulate the DCT coefficients by embedding information (data hiding) within the entropy coded image data. There are multiple identified methods to embed steganography in the entropy coded image data but the most popular found is to use the least significant bits (LSBs) of the DCT coefficients. If a binary program were located within or completely replaced the entropy coded image data, attempting to process the image can be a data attack.

PRODUCT: JFIF

LOCATION:

Scans contain one or more entropy coded segments.

RECOMMENDATION:

AR.1 Validate: N/A

AR.2 Remove: N/A

AR.3 Replace: Replace LSBs of the DCT coefficients with pseudo-random bit values. This may degrade the quality of the image.

AR.4 Replace: Decompress the image and perform lossy compression of the image data. For imagery that requires fidelity to a very high resolution, this may not be acceptable.

AR.5 External Filtering Required: Filters for known executable signatures do not exist in any entropy coded segments.

AR.6 Review: Decode image and present results for human review.

JFIF.5.19: END

JFIF.5.20: COMMENT MARKER SEGMENTS

DESCRIPTION:

The comment (COM) marker segment contains comments or meta data associated with the encoded image such as the author and application used to generate or process the image. The intended content might be a text string but any data may be stored in a COM marker segment.

CONCERN:

The comments may be information not intended for distribution, resulting in inadvertent data disclosure. Data hiding is also possible since arbitrary data can be purposefully placed in the comment marker segment.

PRODUCT: JFIF

LOCATION:
Comment marker segments can be anywhere within the image boundaries after the SOI marker and before the EOI marker.

RECOMMENDATION:

AR.1 Validate: N/A

AR.2 Remove: Remove all comment marker segments.

AR.3 Replace: Substitute the contents of a comment marker segment with a known string such as "The Comments of this image have been removed."

AR.4 External Filtering Required: Pass the string of characters to a filter that processes text strings.

AR.5 Review: Present the contents of the comment marker segment for human review.

JFIF.5.20: END

6. ACRONYMS

Table 6-1 Acronyms

Acronym	Denotation
AR	All Reference
CCITT	International Telephone and Telegraph Consultative Committee
DCT	Discrete cosine transform
ECS	Entropy coded segment
EXIF	Exchangeable image file format
ICC	International Color Consortium
IJG	Independent JPEG Group
ISO	International Organization for Standardization
JFIF	JPEG File Interchange Format
JIF	JPEG Interchange Format
JPEG	Joint Photographic Experts Group
PDF	Portable Document Format
RGB	Red-Green-Blue (light)
SPIFF	Still Picture Interchange File Format
TIFF	Tagged Image File Format
VIS	Verification, Inspection and Sanitization

7. TABLE OF DOCUMENT MARKERS AND SEGMENTS

Appendices

APPENDIX A: REFERENCED DOCUMENTS

Referenced Documents

The following publications are referenced in this document or suggested reading as resource material.

Eastman Kodak Company (1996) *FlashPix Format Specification* v1.0

Hass, Calvin (2008) *Impulse Adventure* http://www.impulseadventure.com/photo

Hamilton, E. (1992) *JPEG File Interchange Format* v1.2

Hornet, C. (2004) *JPEG Vulnerability: A day in the life of the JPEG Vulnerability*

Independent JPEG Group (2010) *Free JPEG Encoding Library* http://www.ijg.org

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

Japan Electronics and Information Technology Industries Association (2002) *Exchangeable image file format for digital still cameras: Exif* v2.2 JEITA CP-3451

APPENDIX B: SUMMARY OF RISKS

Table B-1 Summary of Risks

JFIF Feature	Spec Section	ISG Section	Attack	Hiding	Disclosure
Application Marker Segments	ISO 10918-1 B.2.4.6	JFIF.5.9		X	X
Comment Marker Segments	ISO 10918-1 B.2.4.5	JFIF.5.20		X	X
Dependent Marker Segments		JFIF.5.17		X	
Entropy Coded Image Data Segments	ISO 10918-1 B.2.1	JFIF.5.19		X	
Frame Header Marker Segment	ISO 10918-1 B.2.2	JFIF.5.5		X	
Frames		JFIF.5.4	X	X	
Image Boundaries	ISO 10918-1 B.2.1	JFIF.5.2		X	
Marker Segment Length	ISO 10918-1 B.1.1.4	JFIF.5.3	X		
Marker Values and Order	ISO 10918-1 B.1.1.3	JFIF.5.1	X		
Reserved Marker Segments		JFIF.5.18		X	
Scan Header Marker Segment	ISO 10918-1 B.2.3	JFIF.5.7		X	
Scans		JFIF.5.6		X	
Tables	ISO 10918-1 B.2.4	JFIF.5.8		X	
JFIF APP0 Marker Segment	JFIF 1.02	JFIF 5.10	X	X	
JFIF APP0 Thumbnail Image	JFIF 1.02	JFIF 5.11		X	X
JFIF Extension APP0 Marker Segment	JFIF 1.02	JFIF 5.12	X	X	
JFIF Extension Thumbnail Image	JFIF 1.02	JFIF 5.13		X	X
NXPowerLite APP0 Marker Segment	N/A	JFIF 5.14		X	X
Exif APP1 Marker Segment	JEITA CP-3451	JFIF 5.16	X	X	X
ICC Profiles		JFIF 5.15	X	X	X