Inspection and Sanitization Guidance for the Department of Defense (DoD) Electronic Biometric Transmission Specification (EBTS) File Format

Version 1.0

12 June 2015

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

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

DOCUMENT REVISION HISTORY

Date	Version	Description
6/12/2015	1.0	Final Release
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 (ISG) for Department of Defense (DoD) Electronic Biometric Transmission Specification (EBTS)* document provides guidelines and specifications for developing inspection and sanitization software for EBTS files, as defined by both the American National Standards Institute (ANSI)/National Institute of Standards and Technology (NIST) and the DoD Electronic Biometric Transmission Specification.

EBTS files contain graphic and textual information about a particular subject. This information can be useful for identification purposes. EBTS files are containers for other file formats. The file format is a series of custom records that contain both metadata and data. Metadata can provide textual information about the data and its contents. Metadata can always introduce risk because the information can be ignored by software applications. In addition, certain versions of EBTS viewers may only process a certain subset of metadata. The data in EBTS is typically an image of a fingerprint, iris, palm, or face of a person. This report identifies these image records along with other records that provide non-image data (e.g., fingerprint identification data). Information in each record is used to identify a subject. EBTS files embed several image types including Joint Photography Experts Group (JPEG), Wavelet Scalar Quantization (WSQ), and Portable Network Graphics (PNG) image files. When a file format introduces an embedded file, it inherits all the risks from the embedded file format. EBTS files introduce the same type of risks associated with common image file formats plus the risk introduced by its custom data structure and metadata.

This document examines numerous EBTS specifications for data attack, data hiding, and data disclosure risks that exists within the data structure. It provides a breakdown of each component of an EBTS file and provides recommendations that help assure an EBTS file is compliant with the specifications and does not contain any hidden data.

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 sanitization and analysis software for Department of Defense (DoD) Electronic Biometric Transmission Specification (EBTS) biometric files. This document analyzes elements and objects contained within the EBTS file structure and then discusses the data hiding, data attack, and data disclosure risks. It describes how identified elements can be a cause for concern for hiding sensitive data or attempting to exploit a system. This report provides recommendations to ensure EBTS files are safer for users to open and conform to the specification.

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

1.2 Introduction

Biometric information is used by law enforcement and criminal justice agencies to determine (or verify) the identity of a subject. EBTS was designed to help format and exchange identity data across jurisdictional boundaries and between different systems. It is designed as a common format for data exchange of biometric and personal data regarding a particular subject.

The biometric data contained within EBTS consists of a wide array of both behavior and physical traits of an individual to establish or verify the identity of a person. Examples of these traits include fingerprints, plantars (footprints), palmprints, facial images, deoxyribonucleic acid (DNA) data, and iris images. EBTS also provides for the inclusion of other data, e.g., details about scars, marks, or tattoos; these can be used manually by an operator to verify the identity of a subject.

EBTS also supports data in different formats, this includes: compressed or uncompressed image data, audio and video clips, and processed minuatiae data from friction ridge images[1]. The aggregate of this data can help build a better background for a particular subject, which ultimately helps with identifying the person.

[1] Per www.nist.gov, a friction ridge is “a raised portion of the epidermis on the palms of the hand or the soles of the feet, consisting of one or more connected ridge units of friction ridge skin.” http://www.nist.gov/forensics/EFSTrainingTool/ResourcesTab/Glossary html

1.3 Background

There are many varying implementations and standards of EBTS. The baseline standard, the American National Standards Institute (ANSI)/National Institute of Standards and Technology (NIST) Information Technology Laboratory specification number 1-2011, defines the basic record types and overall file structure. This version is preceeded by a 2007 ANSI/NIST standard. The information provided by ANSI/NIST is a subset of all EBTS data. Other standards implement additional information that supplements the baseline standard. The DoD EBTS implements additional data with custom extensions in user-defined locations in the baseline EBTS.

The DoD version of the EBTS was designed as an interface to the DoD Automated Biometric Identification System (ABIS). The DoD ABIS consists of a database and software tools that can search, store, and retrieve fingerprint and latent data collected from various suspects. The design of ABIS is similar to the Federal Bureau of Investigation (FBI) Criminal Justice Information Services (CJIS) Integrated Automated Fingerprint Identification System (IAFIS). The interface for IAFIS was based on the Electronic Fingerprint Transmission Specification (EFTS). Since the DoD expanded the use of new modality biometric data (e.g., iris) and required unique metadata to meet DoD needs, DoD biometric files are defined in the DoD EBTS rather than the FBI standard. The FBI EFTS evolved and was renamed to EBTS to include additional modalities, however the two (DoD and FBI) specifications remain distinct.

The DoD EBTS is based on the American National Standards Institute (ANSI)/National Institute of Standards and Technology (NIST) Information Technology Laboratory (ITL) specification number 1-2011 (ITL 1-2011). The DoD's EBTS builds upon the ITL 1-2011 to meet DoD requirements via additions to and customizations of the ITL 1-2011 data format.

Following extensive expert review and multiple revisions, the first widely distributed version (1.2) of the DoD EBTS was released in November 2006. An updated version (2.0) was released in 2009, and a minor update (2.1) was released in 2010. Due to the need for the DoD EBTS to be useable for communications with other DoD biometric repositories in addition to DoD ABIS, a major update version (3.0) was released in 2011.

1.4 Document Organization

The following table summarizes the organization of this document.

Table 1-1. Document Organization

Section	Description

Section	Description
Section 1: Scope	This section decribes the scope, organization, and limitations of this document.
Section 2: Construct Overview	This section describes construct information for EBTS.
Section 3: EBTS Overview	This section describes an overview of EBTS.
Section 4: EBTS - Constructs	This section describes the details of EBTS and provides constructs per marker and section in the file.
Section 5: Summary of Risks	Contains a table summary of the risks identified in this document.
Section 6: Acronyms	This section lists the acronyms that appear in this document.
Section 7: Referenced Documents	This section lists the sources used to prepare or cited in this document.

1.5 Actions

Each construct description lists recommended actions for handling the construct when processing a document. 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 of 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.

NOTE

> The recommendations in this document are brief explanations rather than a How-To Guide. Readers should refer to the construct description or the DoD Electronic Biometric Transmission Specification for additional details.

Table 1-2 summarizes the recommendation actions.

Table 1-2. Recommendation Actions

Recommendation Action	Comments
Validate	Verify the data structure’s integrity, which may include integrity checks on other components in the file (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 user name 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 areas.
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 data.

NOTE

No recommendations for logging all actions and found data are included here because all activity logging in a file 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

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 files and protocol messages may 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 filters that specialize in steganography should be used to handle embedded content, such as text, images, videos, and audio.

1.6.2 Scope

This document covers the baseline ANSI/NIST 2007 and 2011 standards for EBTS. There are additional constructs that also cover both version 1.2 and 2.0 of the DoD EBTS standard. The DoD version 3.0 standard of EBTS is not covered by this document since it is not widely used throughout the community.

2. CONSTRUCT OVERVIEW

2.1 Constructs

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

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

2.2 Taxonomy

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

Table 2-1. Document Taxonomy

Term	Definition
Construct	An object that represents some form of information or data in the hierarchy in EBTS data.
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. ELECTRONIC BIOMETRIC TRANSMISSION SPECIFICATION (EBTS) OVERVIEW

The EBTS standard defines the structure of the data format and the number of data types that can exist. EBTS is a series of data records comprising a transaction that is transmitted to another site or agency. The receiving agency sets the requirements for number and type of records, scanning resolution, and other user-specific data in order to consider the transaction valid. Transactions are typically used for one subject or identity, in which all the records in a transaction pertain to a single subject. There are exceptions, such as a candidate list, where a transaction or an EBTS file may contain records pertaining to multiple subjects. Some records may include biometric data from another person, if that data is used to corroborate the identity of the subject in the transaction. A transaction is comprised of records; all the possible record types are listed below in Table 3-1 EBTS Record Types and Contents.

EBTS defines a transmission file format, which defines the structure and organization of a block of data that contains metadata followed by record data. EBTS is a series of records that each provide some type of meaningful data. Record data depends on the type of record that the EBTS file implements. An example EBTS file is shown in the graphic below. The Type-1 record defines the structure, the size, and (optionally) the list of all character encodings used in the file.The Type-2 record provides a textual description about the subject and other pertinent metadata. The other records provide binary or textual metadata followed by the biometric or image data that pertain to the subject.

EBTS records can be divided into two categories: tagged-field records and pure binary records. A tagged-field record is defined as “a logical record containing unique ASCII field identifiers for variable length data fields that is capable of being parsed based on the field identifier and the data contents of each field.”[2] EBTS may also implement binary records that contain fixed binary length fields followed by a block of binary data. Software processing EBTS must be able to switch between parsing ASCII tagged-field records and binary records. In addition, some tagged-field records may contain a mix of text and binary data. All of the fields within the tagged-field record are text except for the final field record, which is a binary data field used to embed a binary file within the data record.

While discussing each record type, it is important to understand the character set and encoding set that are used by each record. EBTS files may contain a combination of American Standard Code for Information Interchange (ASCII) and Unicode textual

[2] ANST/NIST ITL1-2007

data, as well as binary data. The default encoding of the file is 7-bit ASCII with a zero inserted at the high bit position. A Type-1 (Transaction information) record is always the first record in the file and all of its fields and subfields must use 7-bit ASCII. Other records may implement textual fields and subfields with a different encoding. A Type-1 record may contain a field that identifies all the possible encodings used in the file. If this field is not present, it is assumed that, when text is present through the entire EBTS file, it is in ASCII. Table 3-1 introduces each record type and the default encoding. As mentioned before, some tagged-field records contain a mix of text and binary data, where the record begins with textual metadata and concludes with a binary field (e.g., a binary image file). This is referred to in the table below as an ASCII/Binary encoding record. The table shows all the possible record types, their contents, their record type and default encoding. A more thorough discussion on character encoding is available in Section 3.4.

Table 3-1 EBTS Record Types and Contents

Record Type	Record Contents	Record Type (Default Encoding)
1	Transaction information	Tagged-Field (ASCII)
2	User-defined descriptive text	Tagged-Field (ASCII)
3	Low-resolution grayscale fingerprint image (Deprecated)	Deprecated
4	High-resolution grayscale fingerprint image	Binary
5	Low-resolution binary fingerprint image (Deprecated)	Deprecated
6	High-resolution binary fingerprint image (Deprecated)	Deprecated
7	User-defined image Note: Used for fingerprint, face, or iris image data in various DoD EBTS implementations before ANSI/NIST-ITL1-2007	Binary
8	Signature image	Binary
9	Minutiae data	Tagged-Field (ASCII)
10	Face, other body part, or scar, mark tattoo (SMT) image	Tagged-Field (ASCII/Binary)
11	Voice data (future addition to the standard)	Reserved for Future Use
12	Dental record data (future addition to the standard)	Reserved for Future Use
13	Variable-resolution latent friction ridge image	Tagged-Field (ASCII/Binary)
14	Variable-resolution fingerprint image	Tagged-Field (ASCII/Binary)

15	Variable-resolution palmprint image	Tagged-Field (ASCII/Binary)
16	User-defined variable-resolution testing image Note: Used for iris image data in various DoD EBTS implementations before ANSI/NIST-ITL1-2007	Tagged-Field (ASCII/Binary)
17	Iris image	Tagged-Field (ASCII/Binary)
18	DNA data	Tagged-Field (ASCII/Binary)
19	Variable-resolution plantar image	Tagged-Field (ASCII/Binary)
20	Source representation	Tagged-Field (ASCII/Binary)
21	Associated context	Tagged-Field (ASCII/Binary)
22-97	Reserved for future use	Tagged-Field (ASCII/Binary)
98	Information assurance	Tagged-Field (ASCII/Binary)
99	Common Biometric Exchange Formats Framework (CBEFF) biometric data record	Tagged-Field (ASCII/Binary)

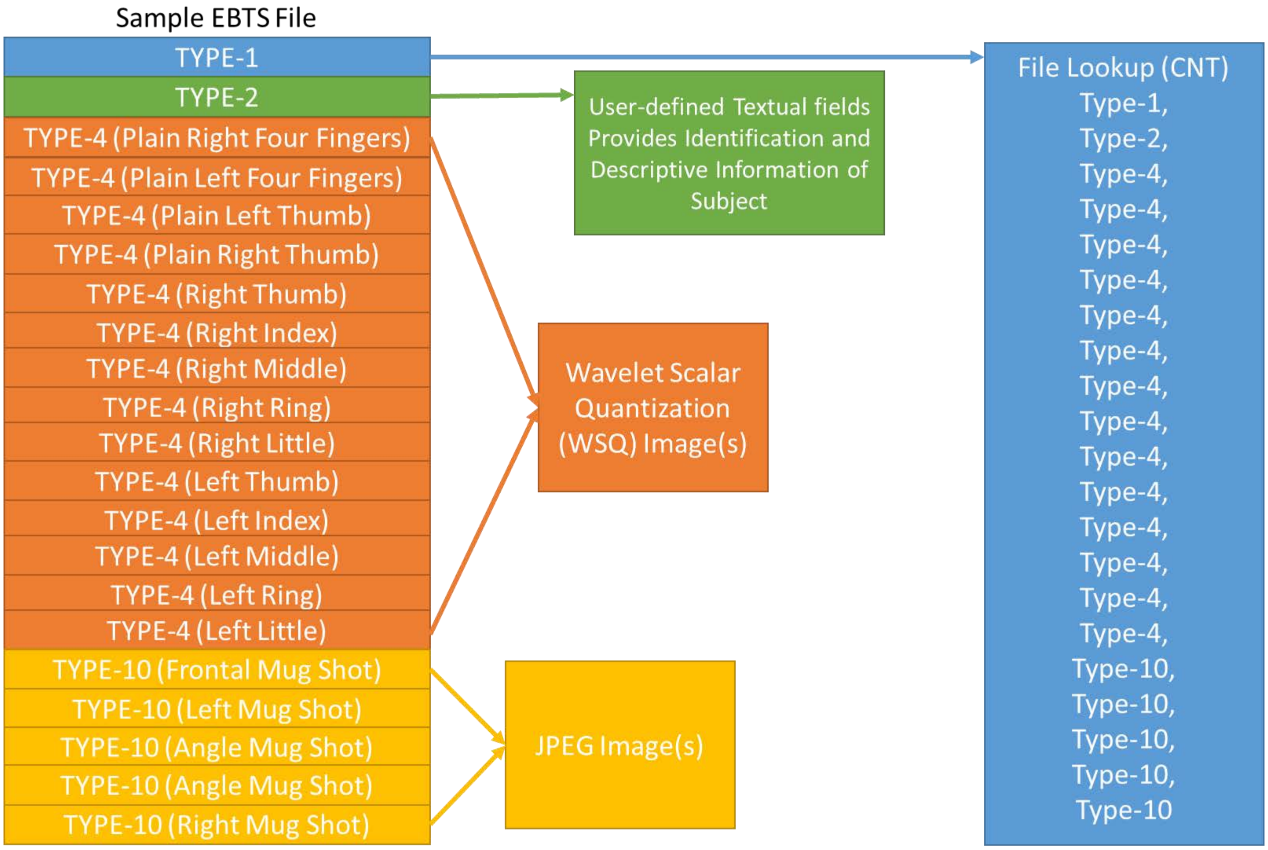


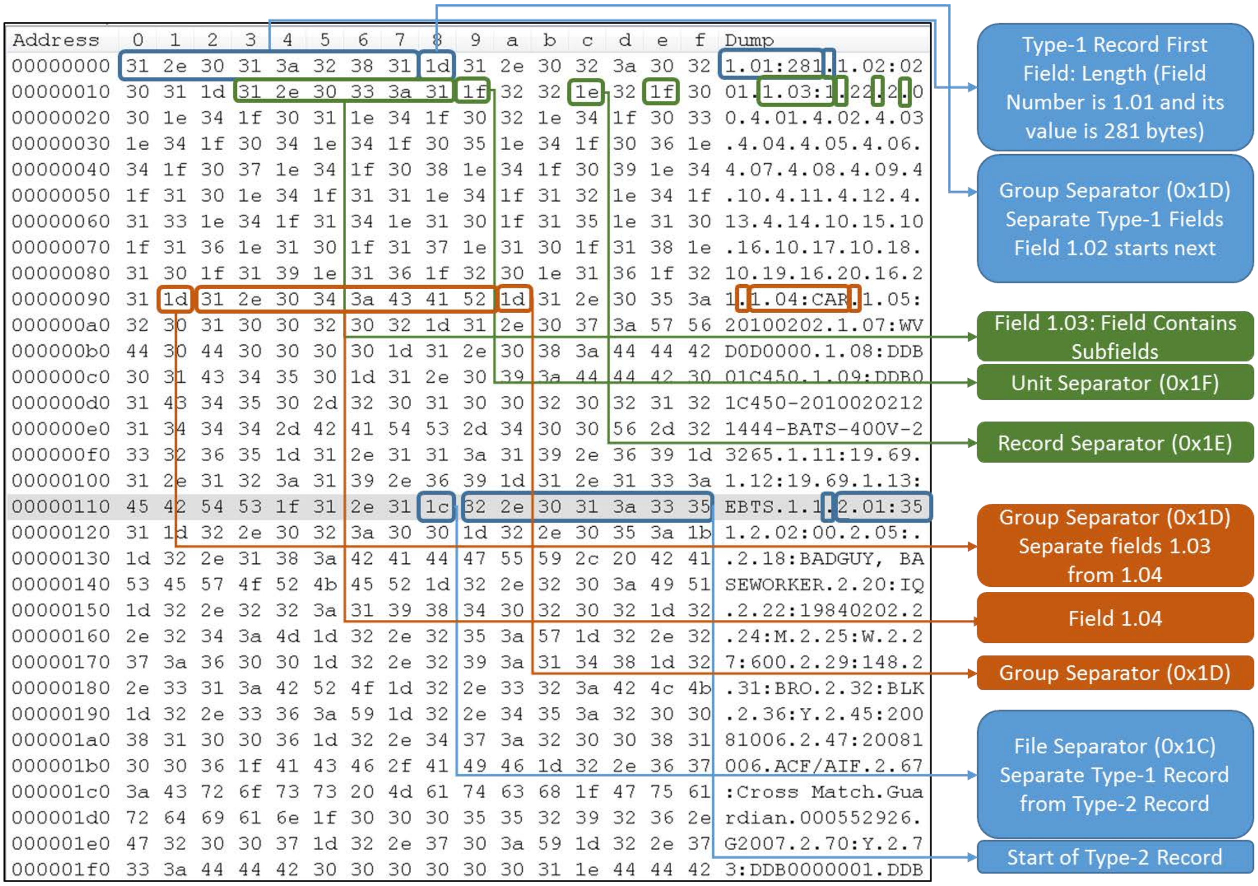
Figure 3-1. Example EBTS File

EBTS implements four different information separators to instruct parsers where one data record ends and a new record begins. Separators are used only in tagged-field records as they are ASCII characters. The four information separators are the File Separator (FS, ASCII=0x1C), Group Separator (GS, ASCII=0x1D), Record Separator (RS, ASCII=0x1E), and Unit Separator (US, ASCII=0x1F). Each information separator serves a different purpose. An FS is used to separate tagged-fields records. The end of every tagged-field record is the FS character. A GS is used to separate fields within a tagged-field record. A tagged-field record implements a number of text fields. Each text field may be further broken down into subfields or information items. The standard defines that a US is used to "separate multiple items within a field or subfield." The RS is used to "separate multiple subfields."

Pure binary records (Types 3-8) do not implement information separators. Each metadata field in the record has a fixed length that is strictly defined within the standard; therefore, the byte boundaries are known. The length of the entire binary record is variable and is the first field of the record; the length is the total number of bytes in the record (metadata plus image data). The next record begins immediately after the specified number of bytes in the EBTS file. The following sections cover example records from the file presented in Figure 3-1. There are numerous records and record types from the example. Each example in the following sections illustrate how to identify separators and boundaries within an EBTS file, as well as how to identify the embedded data that can exist. The first section covers an example tagged-field record using the Type-1 record as an example. This record contains no binary data and is completely ASCII data. The next example shows a binary Type-4 record that follows a Type-2 record (tagged-field ASCII). The last example shows a Type-10 record, a textual record with ASCII metadata that also contains a binary field (JPEG image).

3.1 Example Tagged-Field Record

A Type-1 record is a tagged-field that must be the first record of the file and must be represented in ASCII. Figure 3-2 shows a detailed breakout of a Type-1 record, which is then followed by a Type-2 record. The first field in a Type-1 record (the first few bytes of the file) is the string “1.01:”. This means that there is Type-1 record, field number 1. Following the colon in this example is the value 281. This means that the Type-1 record is 281 bytes in length. This can be validated in Figure 3-2 since the Type-2 record begins at byte offset 0x119 (281). The next byte after the first field is the GS (0x1D) which means that the next field begins after that marker (Field 1.02). Field 1.03 is a field within a Type-1 record that implements a table so it provides two other markers: an RS and US .This allows for the field 1.03 to be broken into a table of individual elements. The Type-1 record concludes with the FS marker (0x1C). The beginning of the Type-2 record (“2.01:”) immediately follows.



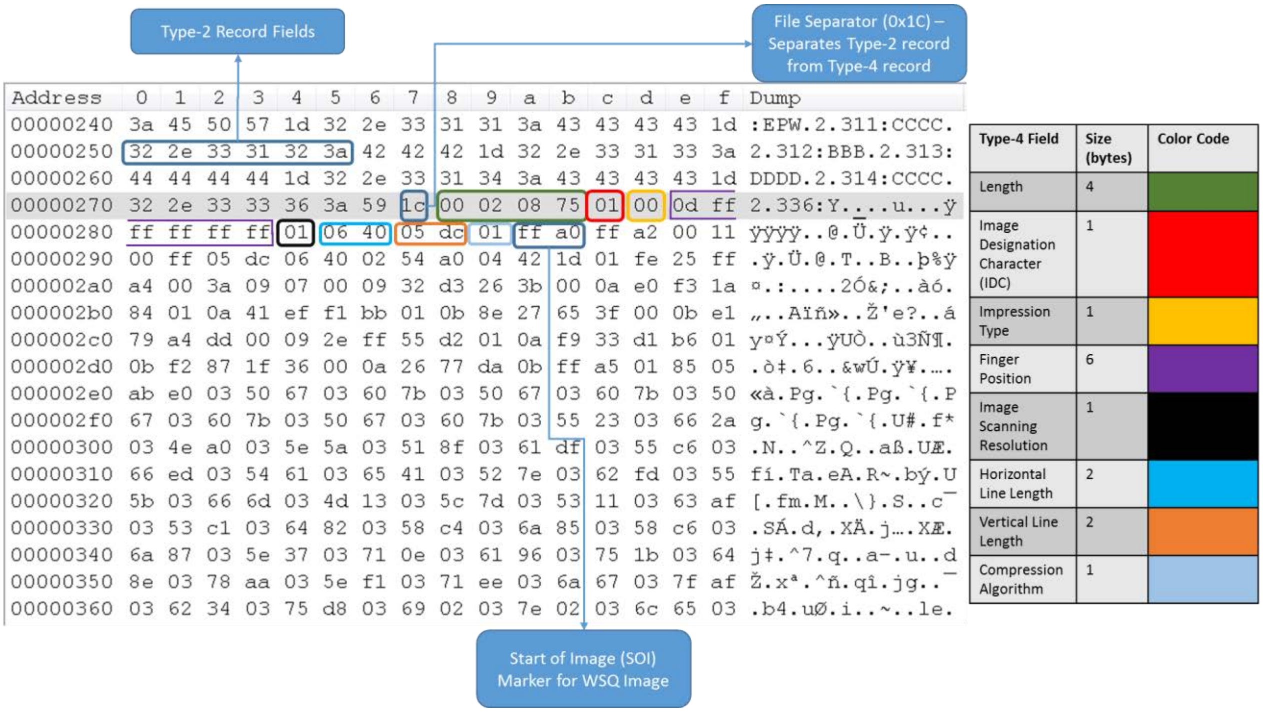
```
Address   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  Dump
00000000 31 2e 30 31 3a 32 38 31 1d 31 2e 30 32 3a 30 32 1.01:281.1.02:02
00000010 30 31 1d 31 2e 30 33 3a 31 1f 32 32 1e 32 1f 30 01.1.03:1.22.2.0
00000020 30 1e 34 1f 30 31 1e 34 1f 30 32 1e 34 1f 30 33 0.4.01.4.02.4.03
00000030 1e 34 1f 30 34 1e 34 1f 30 35 1e 34 1f 30 36 1e .4.04.4.05.4.06.
00000040 34 1f 30 37 1e 34 1f 30 38 1e 34 1f 30 39 1e 34 4.07.4.08.4.09.4
00000050 1f 31 30 1e 34 1f 31 31 1e 34 1f 31 32 1e 34 1f .10.4.11.4.12.4.
00000060 31 33 1e 34 1f 31 34 1e 31 30 1f 31 35 1e 31 30 13.4.14.10.15.10
00000070 1f 31 36 1e 31 30 1f 31 37 1e 31 30 1f 31 38 1e .16.10.17.10.18.
00000080 31 30 1f 31 39 1e 31 36 1f 32 30 1e 31 36 1f 32 10.19.16.20.16.2
00000090 31 1d 31 2e 30 34 3a 43 41 52 1d 31 2e 30 35 3a 1.1.04:CAR.1.05:
000000a0 32 30 31 30 30 32 30 32 1d 31 2e 30 37 3a 57 56 20100202.1.07:WV
000000b0 44 30 44 30 30 30 30 1d 31 2e 30 38 3a 44 44 42 D0D0000.1.08:DDB
000000c0 30 31 43 34 35 30 1d 31 2e 30 39 3a 44 44 42 30 01C450.1.09:DDB0
000000d0 31 43 34 35 30 2d 32 30 31 30 30 32 30 32 31 32 1C450-2010020212
000000e0 31 34 34 34 2d 42 41 54 53 2d 34 30 30 56 2d 32 1444-BATS-400V-2
000000f0 33 32 36 35 1d 31 2e 31 31 3a 31 39 2e 36 39 1d 3265.1.11:19.69.
00000100 31 2e 31 32 3a 31 39 2e 36 39 1d 31 2e 31 33 3a 1.12:19.69.1.13:
00000110 45 42 54 53 1f 31 2e 31 1c 32 2e 30 31 3a 33 35 EBTS.1.1.2.01:35
00000120 31 1d 32 2e 30 32 3a 30 30 1d 32 2e 30 35 3a 1b 1.2.02:00.2.05:.
00000130 1d 32 2e 31 38 3a 42 41 44 47 55 59 2c 20 42 41 .2.18:BADGUY, BA
00000140 53 45 57 4f 52 4b 45 52 1d 32 2e 32 30 3a 49 51 SEWORKER.2.20:IQ
00000150 1d 32 2e 32 32 3a 31 39 38 34 30 32 30 32 1d 32 .2.22:19840202.2
00000160 2e 32 34 3a 4d 1d 32 2e 32 35 3a 57 1d 32 2e 32 .24:M.2.25:W.2.2
00000170 37 3a 36 30 30 1d 32 2e 32 39 3a 31 34 38 1d 32 7:600.2.29:148.2
00000180 2e 33 31 3a 42 52 4f 1d 32 2e 33 32 3a 42 4c 4b .31:BRO.2.32:BLK
00000190 1d 32 2e 33 36 3a 59 1d 32 2e 34 35 3a 32 30 30 .2.36:Y.2.45:200
000001a0 38 31 30 30 36 1d 32 2e 34 37 3a 32 30 30 38 31 81006.2.47:20081
000001b0 30 30 36 1f 41 43 46 2f 41 49 46 1d 32 2e 36 37 006.ACF/AIF.2.67
000001c0 3a 43 72 6f 73 73 20 4d 61 74 63 68 1f 47 75 61 :Cross Match.Gua
000001d0 72 64 69 61 6e 1f 30 30 30 35 35 32 39 32 36 2e rdian.000552926.
000001e0 47 32 30 30 37 1d 32 2e 37 30 3a 59 1d 32 2e 37 G2007.2.70:Y.2.7
000001f0 33 3a 44 44 42 30 30 30 30 30 30 31 1e 44 44 42 3:DDB0000001.DDB
```

Type-1 Record First Field: Length (Field Number is 1.01 and its value is 281 bytes)
Group Separator (0x1D) Separate Type-1 Fields Field 1.02 starts next
Field 1.03: Field Contains Subfields
Unit Separator (0x1F)
Record Separator (0x1E)
Group Separator (0x1D) Separate fields 1.03 from 1.04
Field 1.04
Group Separator (0x1D)
File Separator (0x1C) Separate Type-1 Record from Type-2 Record
Start of Type-2 Record

Figure 3-2. Example Type-1 Record

3.2 Example Binary Record

A common binary record is a Type-4 record which implements a number of metadata fields (in binary) and is immediately followed by an embedded image file. The example in Figure 3-3 shows the ending of a Type-2 record and the beginning of the Type-4 record. A FS (0x1C) marker is used to terminate the Type-2 record. The beginning of the Type-4 has no marker or special bytes. The first field is a 4-byte length field which identifies the length of the binary metadata fields plus the length of the binary image that follows. The rest of the binary fields follow in order, as defined by the EBTS standard. These fields and their definitions are covered in later constructs but are listed in the table in Figure 3-3. There are a total of nine fields in a Type-4 record. The ninth field is the actual image data itself. In the image below, the Start of Image (SOI) marker for WSQ is shown because an embedded WSQ file follows the Type-4 binary metadata. This record is one of many fingerprint records in the EBTS file.



Type-2 Record Fields

File Separator (0x1C) – Separates Type-2 record from Type-4 record

```
Address   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  Dump
00000240 3a 45 50 57 1d 32 2e 33 31 31 3a 43 43 43 43 1d :EPW.2.311:CCCC.
00000250 32 2e 33 31 32 3a 42 42 42 1d 32 2e 33 31 33 3a 2.312:BBB.2.313:
00000260 44 44 44 44 1d 32 2e 33 31 34 3a 43 43 43 43 1d DDDD.2.314:CCCC.
00000270 32 2e 33 33 36 3a 59 1c 00 02 08 75 01 00 0d ff 2.336:Y....u...ÿ
00000280 ff ff ff ff 01 06 40 05 dc 01 ff a0 ff a2 00 11 ÿÿÿÿ..@.Ü.ÿ.ÿ¢..
00000290 00 ff 05 dc 06 40 02 54 a0 04 42 1d 01 fe 25 ff .ÿ.Ü.@.T..B..þ%ÿ
000002a0 a4 00 3a 09 07 00 09 32 d3 26 3b 00 0a e0 f3 1a ¤.:....2Ó&;..àó.
000002b0 84 01 0a 41 ef f1 bb 01 0b 8e 27 65 3f 00 0b e1 „..Aïñ»..Ž'e?..á
000002c0 79 a4 dd 00 09 2e ff 55 d2 01 0a f9 33 d1 b6 01 y¤Ý...ÿUÒ..ù3Ñ¶.
000002d0 0b f2 87 1f 36 00 0a 26 77 da 0b ff a5 01 85 05 .ò‡.6..&wÚ.ÿ¥.….
000002e0 ab e0 03 50 67 03 60 7b 03 50 67 03 60 7b 03 50 «à.Pg.`{.Pg.`{.P
000002f0 67 03 60 7b 03 50 67 03 60 7b 03 55 23 03 66 2a g.`{.Pg.`{.U#.f*
00000300 03 4e a0 03 5e 5a 03 51 8f 03 61 df 03 55 c6 03 .N..^Z.Q..aß.UÆ.
00000310 66 ed 03 54 61 03 65 41 03 52 7e 03 62 fd 03 55 fí.Ta.eA.R~.bý.U
00000320 5b 03 66 6d 03 4d 13 03 5c 7d 03 53 11 03 63 af [.fm.M..\}.S..c¯
00000330 03 53 c1 03 64 82 03 58 c4 03 6a 85 03 58 c6 03 .SÁ.d,.XÄ.j….XÆ.
00000340 6a 87 03 5e 37 03 71 0e 03 61 96 03 75 1b 03 64 j‡.^7.q..a–.u..d
00000350 8e 03 78 aa 03 5e f1 03 71 ee 03 6a 67 03 7f af Ž.xª.^ñ.qî.jg..¯
00000360 03 62 34 03 75 d8 03 69 02 03 7e 02 03 6c 65 03 .b4.uØ.i..~..le.
```

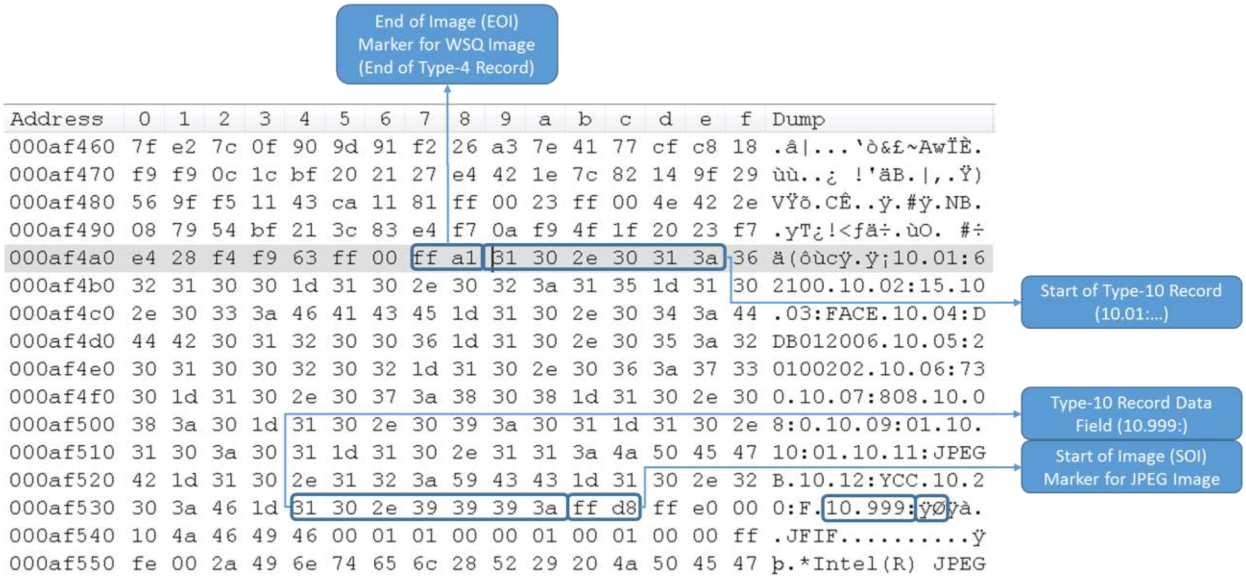
Type-4 Field	Size (bytes)	Color Code
Length	4	🟩
Image Designation Character (IDC)	1	🟥
Impression Type	1	🟨
Finger Position	6	🟪
Image Scanning Resolution	1	⬛
Horizontal Line Length	2	🟦
Vertical Line Length	2	🟧
Compression Algorithm	1	🟦

Start of Image (SOI) Marker for WSQ Image

Figure 3-3. Example Binary Type-4 Record with WSQ Image

3.3 Example Tagged-Field with Binary Data Record

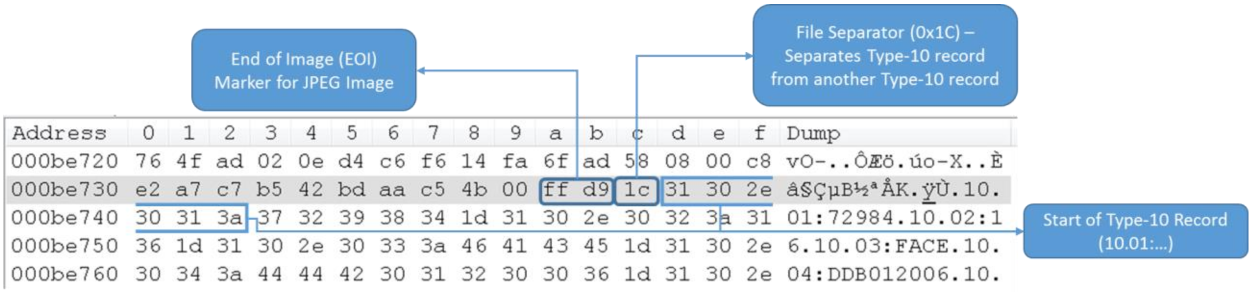
Earlier in this section, Table 3-1 identified some records as ASCII/Binary encoding. This means that the record is a combination of textual tagged-field records and one field that provides binary data (e.g., an image). A common example of this record type is a Type-10 record. Type-10 records are commonly used to provide mugshots of suspects and textual metadata fields such as eye color and hair color. The last field in a Type-10 record is the image data. The example in Figure 3-4 shows the transition between the ending of a Type-4 record and the beginning of a Type-10 record. An End of Image (EOI) marker for a WSQ image is shown highlighted as bytes 0xFFA1. Immediately following the EOI marker is the value "10.01:" which means that this is the beginning of a Type-10 record. The Type-10 record implements tagged-fields that are similar to the Type-1 record shown earlier in this section. The last field of the Type-10 is the field "10.999:" which is the data field for the embedded image. The next two bytes that immediately follow this field are the value 0xFFD8 which is the SOI marker for a JPEG image.



```
Address  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  Dump
000af460 7f e2 7c 0f 90 9d 91 f2 26 a3 7e 41 77 cf c8 18 .â|...'ò&£~AwÏÈ.
000af470 f9 f9 0c 1c bf 20 21 27 e4 42 1e 7c 82 14 9f 29 ùù..¿ !'äB.|,.Ÿ)
000af480 56 9f f5 11 43 ca 11 81 ff 00 23 ff 00 4e 42 2e VŸõ.CÊ..ÿ.#ÿ.NB.
000af490 08 79 54 bf 21 3c 83 e4 f7 0a f9 4f 1f 20 23 f7 .yT¿!<ƒä÷.ùO. #÷
000af4a0 e4 28 f4 f9 63 ff 00 ff a1 31 30 2e 30 31 3a 36 ä(ôùcÿ.ÿ¡10.01:6
000af4b0 32 31 30 30 1d 31 30 2e 30 32 3a 31 35 1d 31 30 2100.10.02:15.10
000af4c0 2e 30 33 3a 46 41 43 45 1d 31 30 2e 30 34 3a 44 .03:FACE.10.04:D
000af4d0 44 42 30 31 32 30 30 36 1d 31 30 2e 30 35 3a 32 DB012006.10.05:2
000af4e0 30 31 30 30 32 30 32 1d 31 30 2e 30 36 3a 37 33 0100202.10.06:73
000af4f0 30 1d 31 30 2e 30 37 3a 38 30 38 1d 31 30 2e 30 0.10.07:808.10.0
000af500 38 3a 30 1d 31 30 2e 30 39 3a 30 31 1d 31 30 2e 8:0.10.09:01.10.
000af510 31 30 3a 30 31 1d 31 30 2e 31 31 3a 4a 50 45 47 10:01.10.11:JPEG
000af520 42 1d 31 30 2e 31 32 3a 59 43 43 1d 31 30 2e 32 B.10.12:YCC.10.2
000af530 30 3a 46 1d 31 30 2e 39 39 39 3a ff d8 ff e0 00 0:F.10.999:ÿØÿà.
000af540 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 ff .JFIF..........ÿ
000af550 fe 00 2a 49 6e 74 65 6c 28 52 29 20 4a 50 45 47 þ.*Intel(R) JPEG
```

Figure 3-4. Example Type-10 Record Beginning

The rest of the data in the Type-10 record in Figure 3-4 is the remainder of the JPEG file. Since the JPEG image data was in the field labeled "10.999:", the record must terminate with an FS marker (0x1C) as shown in Figure 3-5. The last marker before the FS is the value 0xFFD9 which is the EOI marker for a JPEG file. The FS that immediately follows terminates the Type-10 record. A new Type-10 record is defined after the FS. This demonstrates the transition from a Type-10 record (mixed ASCII/Binary) to another record.



```
Address  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  Dump
000be720 76 4f ad 02 0e d4 c6 f6 14 fa 6f ad 58 08 00 c8 vO-..ÔÆö.úo-X..È
000be730 e2 a7 c7 b5 42 bd aa c5 4b 00 ff d9 1c 31 30 2e â§ÇµB½ªÅK.ÿÙ.10.
000be740 30 31 3a 37 32 39 38 34 1d 31 30 2e 30 32 3a 31 01:72984.10.02:1
000be750 36 1d 31 30 2e 30 33 3a 46 41 43 45 1d 31 30 2e 6.10.03:FACE.10.
000be760 30 34 3a 44 44 42 30 31 32 30 30 36 1d 31 30 2e 04:DDB012006.10.
```

Figure 3-5. Example Type-10 Record Ending

.

3.4 EBTS Character Encoding

EBTS data originally implemented 7-bit ASCII for textual data. It is the default character encoding for text in all EBTS files. EBTS also implements binary data for other record types and other fields that provide an embedded data type (e.g., an image). Almost all records (except for Types 3-8) are implemented as a tagged-field record that defines numerous textual fields containing metadata. Fields within a record use various information separators from the ASCII standard character set. Some records (e.g., Type-10 record) provide additional binary data (e.g., a binary JPEG image) that follows the textual metadata. This means that a number of records in EBTS are mixed with both ASCII and binary encoded data. Type 4, 7, and 8 records are implemented in binary only, which requires a strict byte representation from the standard. A parser is required to switch between ASCII, binary embedded data, and pure binary records when they are all in one transaction.

In order to support international character sets (e.g. for foreign names and locations), EBTS relies on Unicode to represent certain characters. EBTS officially supports ASCII, UTF-16[3], UTF-8, and UTF-32 for character encodings in textual records in the 2011 ANSI/NIST standard. This was not always the case. The 2007 standard only supported UTF-16, UTF-8, and ASCII. In the 2007 standard, the field 1.015 (Character Encoding) in a Type-1 record is present to identify all of the possible character encodings used in the file. However, any field not located in a Type-1 record could switch and implement a Unicode string instead of an ASCII string. This was possible by using special control characters to denote the switch to Unicode. To switch from ASCII to Unicode, older versions of EBTS would use the Start of Text (STX, ASCII=0x02) character followed by the ASCII equal sign '='. After these two ASCII characters, Unicode text would follow. The block of Unicode text would be terminated by the special control character End of Text (ETX, 0x03). In addition to the control characters, all text between the STX and ETX markers are base64 encoded. The use of STX/ETX characters and converting to base64 is not required when using UTF-8; thus, the usage of UTF-8 is highly encouraged.

In the 2011 ANSI/NIST standard, the field 1.015 is used but allows only one entry in the field to specify a single alternate encoding. The 2011 standard lists ASCII (7-bit), UTF-16, UTF-8, and UTF-32 as the official character encoding schemes. It also provides for a range of "user-defined character encoding sets." The same switching mechanism from ASCII to Unicode is still available in the 2011 standard, but the standard suggests that any new transaction not use this technique and should, instead, leverage UTF-8 as the Unicode standard.

Fields and subfields within tagged-field records may implement different character types. Character types are different from character encoding. Table 3-2 shows the list of

[3] UTF-XX, **U** from Universal Coded Character Set +**T**ransformation **F**ormat – XX bit, where XX is 8, 16, or 32. http://en.wikipedia.org/wiki/UTF-8 and http://www.unicode.org. Readers should also review the "Unicode Security Risks" ISG v1.0 from 20 June 2013.

character types that are defined in the 2011 ANSI/NIST EBTS standard. The EBTS standards provide a large set of tables describing the contents of each record. Each field and subfield within a record may implement a different character type from the list shown in Table 3-2. Any field that is labeled as “U” may use the alternate encoding specified in the character encoding field 1.015.

Table 3-2. List of EBTS Character Types

Character Type Label	Description
A	Alphabetic: English characters, upper and lower case, or spaces.
AN	Alphanumeric: Alphabetic, plus 0-9 (Numeric)
ANS	Alphanumeric and special characters
AS	Alphabetic and special characters
B	Binary or Base64 for XML
Base64	Base-64 encoded data (exclusively)
H	Hexadecimal representation: 0-9, A-F
N	Numeric: 0-9
NS	Numeric with special characters
U	Unicode characters or ‘User-Defined’

All information separators such as the FS, GS, RS, and US, as well as the STX and ETX characters are all reserved characters in all encodings.

4. EBTS CONSTRUCTS

The following sections present the EBTS constructs, the risks associated with each one, as well as recommendations to help mitigate each risk. The initial constructs are focused on the file structure and common issues that are found in several EBTS record types.

Recommendations in constructs that describe a removal or replacement option may require a complete build of a new file or the new record. Any change to an EBTS record or field will most likely require realignment of the file, which means adjusting length values and and possibly changing the Type-1 or Type-2 record (from Table 3-1) to reflect the new data. Type-1 and Type-2 records contain structural and descriptive information about the remainder of content in the EBTS file. Replacement or removal of a field may also involve zeroing out (0x30) or injecting spaces (0x20) into the fields. This can be done while preserving the original length of the field and will not require restructuring the entire file. This option can be taken to simplify the filtering process while still replacing or removing content. It is the filter developer's responsibility to understand that manipulation of records or fields may require restructuring of the entire EBTS data.

4.1 Record/Field Numbers and Data Types

OVERVIEW:
This section focuses on generic field records and various different field types including numeric, alphabetic, alphanumeric, and binary. All EBTS files start out with a single Type-1 record (covered in construct 4.10). This construct covers the formatting of the record and field numbers for all fields in EBTS. For tagged-field logical records (Type-1, 2, 9, 10-17, and 99), the field number is defined in the form of "TT.xxxxxxxxx". The first number is the logical record type number and can be up to two digits long. For example, a Type-99 record will start with "99.x" to designate that this field belongs to a Type-99 record. In the case of a Type-2 record, the field will start with "2.x". The second number is the field number and can be variable in length, but is a maximum of 9 digits long. It is valid for a number to contain preceding zeros before the non-zero field number. This number is an unsigned integer and should be previously defined in a specification. Following the field number is a colon which defines that the next portion is the field value itself (which could be in a variety of formatted data).

An example of the First Field Number and Type is shown in Table 4-1 as Field Number "1.001". Other examples from a Type-1 Record and a Type-2 Record are also shown in the table.

Table 4-1. Example Field Numbers

Record and Field Number	Field Name	Character Type
1.001:281	Logical Record Length	Numeric (281 is numeric)
1.002:0201	Version Number	Numeric

1.003:1	File or Transaction Content	Numeric
2.002:00	Image Designation Character	Numeric

Many EBTS specifications will declare a character type which specifies the content of the field. The content of the field could be a simple number or string, or it could be a more detailed structure or table that contains data up until the next Record/Field number combination. Basic validation should be done to ensure that when a Record/Field number defines a character type, then the data following that should adhere to that character type.

RISKS AND RECOMMENDATIONS:

Data Hiding: Unknown, deprecated, or unused record/field numbers could be ignored by applications. Fields with invalid character types may be an attempt to hide data within EBTS data.

1. **Validate:** Check that each record and field number for any record is valid from one of the EBTS specifications.

2. **Remove:** Remove any deprecated fields in the EBTS file.

3. **Validate:** Check that the remainder of the field contents aligns with a character type as denoted by the appropriate standard.

4. **Validate:** If the field content has a type or format (such as a date), check that the field adheres to the correct format.

5. **Validate:** If the field specifies a minimum and/or maximum value, verify that the value meets that criteria.

Data Attack and Data Hiding: An improperly formatted record and field number combination could be an attempt to target a parser and cause a buffer overflow if an application fails to account for the maximum sizes of these fields. Improper formatting may also be an attempt to hide fields within the record.

6. **Validate:** Check that the record and field number is in the format of “TT.xxxxxxxxx” (at a maximum), followed by a colon (‘:’). At a minimum, a record/field number could be in the format of “T.x”

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Record and Field numbers are present throughout the entire file in tagged-field ASCII records.

4.2 Information Separators

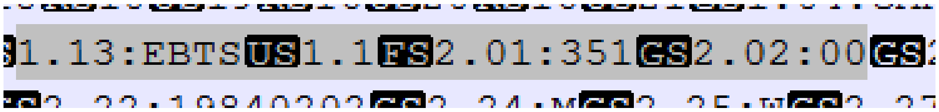
OVERVIEW:

EBTS defines four ASCII information separators that are used as delimiters for tagged-field logical records (Types-1, 2, 9-17, and 99). These separators are not used in binary records; if they are discovered in binary records they are simply part of the binary data. Separators can be used within a field or a subfield, fields within a logical record, or multiple occurrences of subfields. There is a hierarchy to the separators as the File Separators (FS) are the top most level separator and they go down in order as shown in Table 4-2.

Table 4-2. Information Separators [1]

ASCII Character	ASCII Hex Value	Column/Row Position	Description
FS	0x1C	1/12	File separator: Separates logical records of a file or is the terminating character of a transaction.
GS	0x1D	1/13	Group separator: Separates fields of a logical record.
RS	0x1E	1/14	Record separator: Separates multiple data entries (subfields) of an Information field.
US	0x1F	1/15	Unit separator: Separates individual information items of the field or subfield.

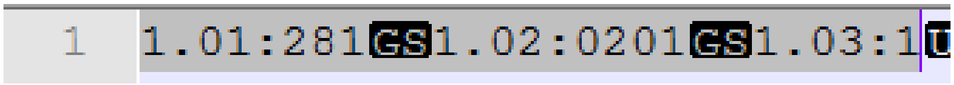
File separators (FS) exist in between logical records of a file. The image below shows the end of a Type-1 record denoted by a FS and the start of a Type-2 record, which is denoted by "2.01:351".



```
1.13:EBTSUS1.1FS2.01:351GS2.02:00GS
```

Figure 4-1. File Separators

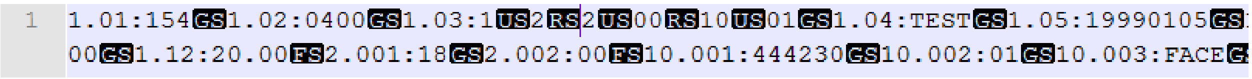
Group separators (GS) are common as they separate fields that are defined in section 4.1. This is shown in Figure 4-2, where the "GS" is shown between the fields "1.01:281" and "1.02:0201".



```
1.01:281GS1.02:0201GS1.03:1U
```

Figure 4-2. Group Separators

Record separators (RS) are used to define a table of multiple entries as shown in Figure 4-3 below, while a FS signals the end of the record. Unit Separators (US) are used to separator individual items within a single field or a subfield defined by the RS. These fields will follow a GS since they are used within fields. An empty subfield can be found when valid separators are sequential in the file (e.g., two US means that the first subfield is empty).



```
1.01:154GS1.02:0400GS1.03:1US2RS2US00RS10US01GS1.04:TESTGS1.05:19990105GS
00GS1.12:20.00FS2.001:18GS2.002:00FS10.001:444230GS10.002:01GS10.003:FACEG
```

Figure 4-3. Type-1 Record with Record Separators

RISKS AND RECOMMENDATIONS:

Data Hiding and Data Attack: Records and fields that are not bounded by their appropriate separator could lead to a buffer overflow by parsing application. Data added to the end of the file is a common attack method. It may also lead to skipping over fields or misinterpreting data depending on the parsing application.

1. **Validate:** Check that the FS character is the last byte of the file or transaction.

2. **Validate:** Check that information separators are present in the correct order as defined by the table in this section.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Information separators are located throughout the entire file in ASCII records.

4.3 Mandatory and Optional Fields

OVERVIEW:

Within logical records, there are several fields defined by numerous EBTS specifications, some fields are mandatory and others are optional. For example, a mandatory field number is “1.001” which is the Logical Record Length in the ANSI/NIST 2007 standard. It defines the length of the Type-1 record. The field “1.006” is optional and called the Priority field. It provides a number between “1” and “9”, signifying the priority of the transaction. There is no complete list of fields in a single specification, so checking for all possible fields is only possible when all EBTS specifications are considered. In certain applications, it is possible to accept only fields from one specification. Inspecting EBTS data requires that every field is accounted for, is defined formally in one of the specifications, and its data is validated. Furthermore, mandatory fields shall be present, otherwise the EBTS file has not been created correctly.

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: Invalid records with missing mandatory or unknown fields could be an attack risk.Unknown fields could lead to a data hiding risk if they are not parsed by an application.**Validate:** Check, for each record in EBTS, that each and every mandatory field is implemented.

1. **Remove:** Remove undefined fields not present in any EBTS specification supported by this ISG.

2. **Remove:** Remove optional fields from the EBTS record. This will remove useful information needed by some applications and may disrupt the transaction between two endpoints.

3. **Reject:** Reject files with unkown or undefined fields.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Tagged field records often contain a mix of mandatory and optional fields, while some binary records may only contain mandatory binary fields.

4.4 Record Field Size

OVERVIEW:

The data that follows a Record and Field Number is often limited to a maximum byte count. For example, the Type-1 Record, Field Number "002", or "1.002" as it appears in the file has a maximum size of 11 bytes per the ANSI/NIST specification. This includes the "1.002:" that precedes the value. EBTS can define a minimum field size per occurrence and a maximum field size per occurrence. It may also define a maximum byte count which includes the record and field number, in addition to the actual data in the field. An example from ANSI/NIST is shown in Table 4-3. This section focuses on the fields Field Size per Occurrence and Max Byte Count.

Table 4-3. Example Field Sizes and Max Byte Counts [1]

Identifier	Conditional Code	Field Number	Field Name	Character Type	Field Size Per Occurrence		Occurrence Count		Max Byte Count
					Min	Max	Min	Max	
VER	M	1.002	Version Number	N	**5**	**5**	1	1	**11**
CNT	M	1.003	File Content (or Transaction Content)	AN	**4**	**6**	2	*	*****
TOT	M	1.004	Type of Transaction	A	**4**	**5**	1	1	**11**

The Field Size per Occurrence and the Max Byte Count define formal limitations for each field type. These values should be checked for correctness, when possible. Some fields such as the File Content field have a Max Byte Count of '*', which means that there is no established limit. The same field has a Max Occurrence of '*', which means there can be any number of File Content fields of any length. The File Content field is unlimited because EBTS may contain any number of records, and this field is responsible for listing the record types contained within the rest of the file. Despite this field having an unlimited size, it can still be validated for sanity. For example, the File Content field implements a table that defines the total number of records in the file. Therefore, the size of the File Content field should be relative in size to the number of records in the EBTS file; therefore, the size of the File Content field can be validated.

RISKS AND RECOMMENDATIONS:

Data Attack: Any field that attempts to go above its maximum limit might be the sign of an attack risk targeting a buffer overflow of an application parsing EBTS data.

1. **Validate:** Check all fields such that the length of the entire field (record, field number, and data) is within the maximum byte count, if defined.

2. **Validate:** Check all fields such that the data portion of the field is within the minimum and maximum range, if defined.

3. **Validate:** Check the length of unlimited size fields for consistency. For example, if the length of the field is relative to the number of records in the file, check that this is correct. This might not be possible for all fields as the content of the field might be easily defined or restricted.

4. **Reject:** Reject files that contain fields that exceed their maximum byte count.

5. **Reject:** Reject files that provide data that is not within their minimum and maximum field lengths.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Each record will contain numerous fields that may have a minimum and maximum field length.

4.5 Field Occurrence

OVERVIEW:

Within a record, a field may appear multiple times. In some specifications, there is a minimum and a maximum occurrence count for each field type. While optional fields must have a minimum of 0 occurrences and mandatory fields often have a minimum occurrence of 1. There are some fields such as the Type-1 Record, Field Number 1.003 (File Content) which has a minimum occurrence of 2. While each field should be examined as mandatory or optional (See Section 4.3) as discussed earlier, the occurrence of each field should be checked as well.

RISKS AND RECOMMENDATIONS:

Data Hiding: Too many occurrences of a field may be ignored by an application. It may also be the sign of a bad EBTS writer application or hidden data.

1. **Validate:** Check that, for each field type in an EBTS record, the number of occurrences is within the minimum and maximum values provided by the specification (if given).

2. **Remove:** Remove extra occurrences of a field from a record. Note that removing the extra occurences may require a rebuild of the file due to new length values.

3. **Reject**: Reject files with field occurrence that extend past the defined maximum or below the minimum.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:
Each field may have numerous occurrences within their record.

4.6 Record Order

OVERVIEW:
An EBTS file or transaction is a series of records, some of which are the tagged-field ASCII records, while others are in a pure binary data format. Applications that parse EBTS data expect a certain number and order of each record type based upon the CNT field in the Type-1 record. A filter must examine EBTS data for correctness, which includes finding duplicate or undefined records. For example, all EBTS data must begin with a Type-1 record, which is then followed by a mandatory Type-2 record. The Type-1 record will define the order and number of records that follow the Type-2 record. The records contained in an EBTS transaction must follow the order specified in the 1.003 File (or Transaction) Content (CNT) field of the Type-1 record. Later versions of EBTS refer to this field as a Transaction Content field as opposed to File Content. If the order of records is incorrect or if there are duplicate records somewhere in the file, this may introduce a problem for parsers.

RISKS AND RECOMMENDATIONS:
Data Attack and Hiding: An incorrect order or number of records may lead to parsing problems. Records that are out of order or duplicate records may cause parsers to ignore other records which could lead to hidden data.

1. **Validate:** Validate order of each record against the transaction record count field 1.003 (CNT).

2. **Reject:** Reject files that do not have the exact correct order of records as defined by the CNT field.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:
Records are located sequentially in a file as specified in the record count field 1.003 (CNT) discussed in later sections.

4.7 Trailing Data

OVERVIEW:

EBTS data is a series of records. These records must follow the order defined in the CNT filed in a Type-1 record. Data can exist after the end of the last valid record in the file. This problem is common in many file formats and is known as the trailing data problem, where data can be appended after the last legal byte in the file.

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: Malicious content could be appended to the end of the file. The trailing data content might be used as part of an attack if the valid EBTS data is ignored. Trailing data can also introduce hidden data that might not be detected by parsers.

1. **Remove:** Remove data following the last byte of legal EBTS data.

2. **Reject:** Reject files that contain any trailing data.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Trailing data would exist at the very end of the file.

4.8 Record Lengths

OVERVIEW:

All records in EBTS begin with a length field. This defines the length of the record and can inform the parser when the next record should begin. If a length field is incorrect, this could lead to problems for parser trying to locate the next record. This is especially true in binary records where an information separator is not present to mark the record boundaries.

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: If a length is incorrect it may cause a parser to incorrectly allocate memory or could lead to an overflow attack. If the length is incorrect, a parser may still ignore content up until the next valid record.

1. **Validate:** Check that for each record in the file that each length field is accurate and that the next valid record immediately follows the previous valid record.

2. **Remove:** Remove any arbitrary data in between records that should not exist in these locations and check that the length fields are correct.

3. **Reject:** Reject EBTS data with invalid length values (in any record).

4. **Reject:** Reject EBTS data that contain any arbitrary data in between records.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Length fields are located in each record, including ASCII and binary records.

4.9 Generic Image Records

The ANSI/NIST standard defines numerous image records. Types-3-8, 10, 13-17, and 19 are all image based EBTS records. The record implements some metadata about the image, and then is followed by the actual image data. A summary of each record type is shown in Table 4-5 below. Some records are entirely binary data (both the image and metadata). There are other records that implemented a tagged-field structure (in ASCII), but the data field (image) is in a binary format.

Table 4-4. EBTS Image Records

Type Record	Description
3	Deprecated.
4	Fingerprint image record.
5	Deprecated.
6	Deprecated.
7	User-defined image record.
8	Scanned binary or vectored signature image record.
10	Scars, Marks, and Tattoos (SMT) image record.
13	Variable-resolution latent image record.
14	Variable-resolution fingerprint image record.
15	Variable-resolution palmprint image record.
16	User-defined image record.
17	Iris image record.
19	Plantar print image record.

Image records listed in Table 4-4 may implement a tagged-field with binary image data appended to the record, or the entire record may be listed as binary data. Type-4 records are entirely binary with fixed byte length values for each field in the record. There are several common fields that are present in all records. This section addresses those fields and the risks that they introduce. The subsections will address those that are specific to that record.

Image records implement some of the fields listed in Table 4-5 below. It is important to stress that some records may contain this data in a binary format, while others may implement a tagged-field structure (in ASCII). It is up to the record type and its internal structure to determine how to extract this information. Image records do not have to implement every one of the fields in the table below, so the appropriate standard should be consulted for the exact record structure.

Table 4-5. Common Image Record Fields

Field Name	Description
Logical Record Length	Length of tagged-fields plus image data, covered in earlier constructs.
Image Designation Character (IDC)	Unique ID for record, covered in earlier constructs.

Source Agency/ORI	The source agency that captured the information, this can be free-text information that is entered by a person. Covered by construct **Error! Reference source not found.**.
Capture Date	Image records provide a capture date when the fingerprint/palmprint/latent image was captured.
Horizontal Line Length	Number of horizontal lines in the image.
Vertical Line Length	Number of vertical lines in the image.
Scale Units	0 – no scale given, 1 – pixels per inch, 2 – pixels per centimeter.
Horizontal Pixel Scale	Mandatory field to specify integer pixel density for the horizontal direction.
Vertical Pixel Scale	Mandatory field to specify integer pixel density for the vertical direction.
Compression Algorithm	Code values from 0-6, see Table 4-7.
Bits per pixel	The resolution of the image. For example, if this value is '8', then each pixel has a value range from 0-255.
Print Position Coordinates	The Print Position Coordinates field contains a set of coordinates to identify a bounding box where the latent finger or palmprint resides. The field contains a series of (X, Y) values identifying the bounding box. These are common in Type-13-17 records.
Comment	Some image records support a free-text comment field. Comments have been addressed in construct 4.14.
User-defined fields	Some image records support several hundred user-defined records, which are defined outside the ANSI/NIST standard. Covered by construct **Error! Reference source not found.**.
Image Data	The actual binary image data.

The compression algorithm codes are standard codes from the EBTS standard. They are listed in Table 4-6 below.

Table 4-6. Compression Algorithm Codes in EBTS [1]

Compression Algorithm	Code	Code Representation as ASCII String
Uncompressed	0	NONE
WSQ Version 2.0	1	WSQ20
JPEG ISO/IEC 10918 (Lossy)	2	JPEGB
JPEG ISO/IEC 10918 (Lossless)	3	JPEGL
JPEG 2000 ISO/IEC 15444-1 (Lossy)	4	JP2
JPEG 2000 ISO/IEC 15444-1 (Lossless)	5	JP2L
Portable Network Graphics	6	PNG

RISKS AND RECOMMENDATIONS:

Data Attack, Data Hiding, and Data Disclosure: Since image records provide complex image data (data in another format); it presents all three types of risk that should be handled by another filter.

1. **External Filtering Required** – Pass the content of the Image Data field to an external filter.

Data Hiding: Field values that are incorrect according to the standard format may contain hidden data or an attempt to hide the image data. An invalid or incorrect compression algorithm (not reflecting the DATA field) can lead to an EBTS viewer to skip the image.

2. **Validate -** Check that the capture date field is a valid date.

3. **Validate:** Check that the Compression Algorithm is from the whitelist defined in this section.

4. **Validate:** Check that the Compression Algorithm field matches the DATA that is provided.

5. **External Filter Required:** Use an external filter to validate the DATA is the correct image format as defined in the Compression Algorithm field.

6. **Reject:** Reject files with any invalid compression algorithm code listed in any image record.

Data Hiding and Data Attack: If the image viewer for EBTS relies on the Image Resolution (Horizontal Line Length (HLL) and Vertical Line Length (VLL)) to render the image, an incorrect value could lead to problems obscuring or hiding the image. It may also be the sign of a data attack risk, targeting applications that rely on them for memory allocation. Deprecated image records could be a source for malicious data or hidden data.

7. **Validate:** Check that all the image parameters align with the image provided in the data field, if possible.

8. **Validate:** If the Compression Algorithm field is 0 (Uncompressed), then check that the size of the DATA field is correct with the image parameters (BPP * HLL * VLL)

9. **Remove:** Remove deprecated image records.

10. **External Filtering Required:** Use an external filter to validate the image parameters are correct.

11. **Reject:** Reject files with deprecated image records.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

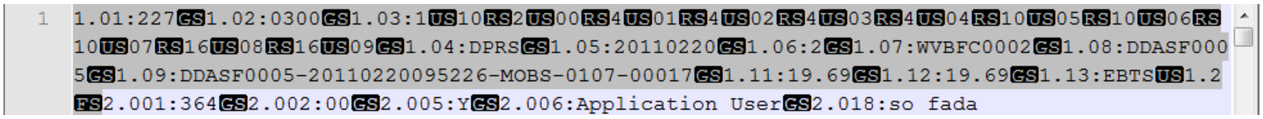
Image records are located throughout the EBTS files, they can be found by looking at the CNT field in the Type-1 record, which is discussed in the next construct.

4.10 TYPE-1 Record

OVERVIEW:
Every EBTS data contains a Type-1 record at the beginning of the file. It is a tagged-field record that implements fields in ASCII. It is mandatory and the first field is the length field of the Type-1 record. This serves as a "Magic Number" for the data, as all EBTS data must start with this byte sequence.

The following example in Figure 4-4 is a dump of the Type-1 record starting from the beginning of an EBTS file. A Type-2 record is shown beginning at the end of this block with the record and field number "2.001."

A Type-1 record implements a Type of Transaction (ToT) field. This defines the type of EBTS transaction. This field is from an enumerated list in the EBTS Transactions construct and can be found in an applicable EBTS standard. In the example below, the ToT is equal to 'DPRS'. The first field is the Logical Record Length, which defines the length of the entire Type-1 record.



```
1.01:227GS1.02:0300GS1.03:1US10RS2US00RS4US01RS4US02RS4US03RS4US04RS10US05RS10US06RS
10US07RS16US08RS16US09GS1.04:DPRSGS1.05:20110220GS1.06:2GS1.07:WVBFC0002GS1.08:DDASF000
5GS1.09:DDASF0005-20110220095226-MOBS-0107-00017GS1.11:19.69GS1.12:19.69GS1.13:EBTSUS1.2
FS2.001:364GS2.002:00GS2.005:YGS2.006:Application UserGS2.018:so fada
```

Figure 4-4. Example Type-1 Record

RISKS AND RECOMMENDATIONS:
Data Attack and Data Hiding: A Type-1 record is located at the very beginning of the file. As with many file types, files can appear as other files or act as many file types depending on how a parser interprets the file. There is a general risk that software could interpret the file as some other type, possibly leading to an attack. Length fields in the Type-1 record have caused application crashes in applications. Data may also be hidden at the very beginning of the file if a parser ignores content up until a valid start of the file header later.

1. **Validate**: Check that the first few bytes of the file begin with the Logical Record Length field (This can range from 1.1: through 1.000000001: as the tagged-field number can be a 1 to 9 digit number between the period and the colon, provided it stays within the Max Byte Count as discussed in construct 4.4).

2. **Validate**: Check that the Logical Record Length field in a Type-1 record is correct with the remainder of the record data.

3. **Validate**: Check that the ToT field is a valid entry from a whitelist in the appropriate EBTS standard for this data type.

4. **Remove:** Remove any data at the beginning of the file up to a valid Type-1 Record, as this should never occur with EBTS data.

5. **Reject:** Reject files that do not start with a valid Type-1 record.

6. **Reject:** Reject files with more than 1 Type-1 Record.

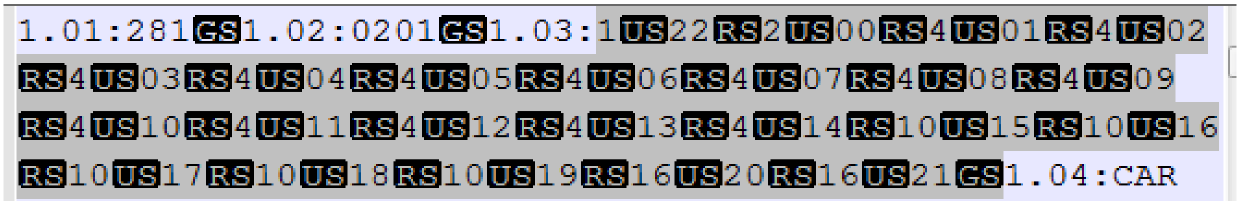
PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:
Type-1 records are located at the very beginning of the EBTS data.

4.10.1 TYPE-1 Record – File Content Field

OVERVIEW:
The CNT field is located in a Type-1 Record, it is the third field of the record. The ANSI/NIST specification defines it as follows: "This mandatory field shall list and identify each of the logical records in the file by record type. It also specifies the order in which the remaining logical records shall appear in the file." The CNT field defines the entire layout and structure of the remaining file. This is an important field that must be correct. When sanitizing or modifying a file (e.g., removing a record), it is an important field to consider as it may need to change to reflect the new or modified data.

The CNT field itself consists of two or more subfields. The first field represents the current Type-1 record, and the first entry in the CNT field is always '1' as illustrated below in Figure 4-5.



```
1.01:281GS1.02:0201GS1.03:1US22RS2US00RS4US01RS4US02
RS4US03RS4US04RS4US05RS4US06RS4US07RS4US08RS4US09
RS4US10RS4US11RS4US12RS4US13RS4US14RS10US15RS10US16
RS10US17RS10US18RS10US19RS16US20RS16US21GS1.04:CAR
```

Figure 4-5. Example CNT Field in File

In the example above, the highlighted section is the CNT field, followed by "1.03:" which identifies the field. There are several subfields that exist in this field which ends before the next field defined by the value of "1.04", the next field in the Type-1 Record. The Unit Separator and Record Separators are shown in a way that this field translates to the following table. If using each Record Separator as delimiting a row in a table and the Unit Separator for delimiting a column, this data in Table 4-7 can be seen.

Table 4-7. Example in Table Form of a CNT Field

Record Type	2nd Field	Explanation of 2nd Field
1	22	22 = Sum of all Type-2 through Type-99 Records in the file (and entries remaining in this table)
2	00	IDC (Image Designation Character)
4	01	IDC

4	02	IDC
4	03	IDC
… (4)	… (04-12)	… (IDC)
4	13	IDC
4	14	IDC
10	15	IDC
… (10)	… (16-19)	… (IDC)
16	20	IDC
16	21	IDC

For all non-Type-1 Records, the 2nd field is an Image Designation Character, which matches a field in the record itself. This file consists of a single Type-1 and Type-2 record, 14 Type-4 records (fingerprints), 5 Type-10 records (images), and 2 Type-16 records (variable resolution images).

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: If the CNT field is incorrect, a parser may incorrectly parse the remainder of the file, possibly leading to an attack. For example, in one test file, the number of records was modified to be incorrect and caused an application crash in a commonly used EBTS library. If the CNT field does not list all the records in the file, some parsers could ignore some records and parse only the content described in the table.

1. **Validate:** Check the First field for every entry is a valid number between 1 and 99.
2. **Validate:** Check that the First entry is 1, and no other Type-1 Record should be listed in the table.
3. **Validate:** Check that the count of all Type-2 – 99 records (both in the file and in the CNT table) equal the 2nd field number.
4. **Replace:** If possible, replace the CNT field with a table that represents the actual file content.
5. **Reject:** Reject files that contain CNT tables that either not formatted correctly or do not line up with the remainder of the file content.

PRODUCT: ANSI/NIST-ITL 2007

LOCATIONS:

The CNT field is denoted by "1.003" and is the third field in a Type-1 Record.

4.10.2 TYPE-1 Record – Character Encoding

OVERVIEW:

Type-1 records introduce a field called Character Encoding, field number 1.015. This field allows the user to specify a character set that can be used in later fields. These fields must be defined with a "U" symbol based on the EBTS standard; this means that the character set is Unicode or "User-Defined". Switching between 7-bit ASCII and Unicode should be handled carefully. An EBTS parser must understand each field that it parses and the content of that field. Type-1 records are always in 7-bit ASCII. If a Character Encoding field is defined, then it is possible that other fields are in a different encoding. The following table shows the list of supported encodings as defined from the 2011 ANSI/NIST EBTS standard. The first column is the range of values allowed for the Character Encoding field.

Table 4-8. Character Encoding for EBTS

Character Encoding Value	Character Encoding Name	Description
0	ASCII	7-bit ASCII with zero in the high bit position (Default encoding)
2	UTF-16	16-bit Unicode Transformation Format (Variable length encoding)
3	UTF-8	8-bit Universal Character Set Transformation Format (Variable length encoding)
4	UTF-32	32 bit Unicode Transformation Format (Fixed length encoding)
5-127	--	Reserved
128-999	--	User-defined character encoding sets.

RISKS AND RECOMMENDATIONS:

Data Hiding and Data Attack: A user-defined encoding will not be able to undergo inspection and could be the source of a data hiding risk. Applications may not handle different encodings correctly and may be the source an attack risk.

1. **Validate:** Check that the character encoding is one of the valid supported encodings from the standard. Other Unicode fields throughout the EBTS file should also be checked for correctness.

2. **Reject:** Reject files with custom or user-defined encoding.

3. **Reject:** Reject files that implement UTF-16 or UTF-32, as UTF-8 is a better alternative because it is compatible with ASCII (for English text).

PRODUCT: ANSI/NIST-ITL 2007/2011

LOCATIONS:

The Character encoding field is in a Type-1 record, denoted by the number "1.015"

4.11 TYPE-2 Record

OVERVIEW:

Type-2 records contain user-defined textual fields providing identification and descriptive information associated with the subject of the transaction. Each entry in a Type-2 record is expected to have a definition and format that is listed with the Domain owner. Data contained in this record are expected to conform in format and content to the specifications of the domain name(s) as listed in Field 1.013 Domain name (DOM) found in the Type-1 record, if that field is in the transaction. If not defined, then the default domain is the North American Domain Implementation (NORAM). Field 1.016 (Application profile specifications (APS)) allows the user to indicate conformance to multiple specifications. If Field 1.016 is specified, the Type-2 record must conform to each of the application profiles.

A DOM or APS reference uniquely identifies data contents and formats. Each domain and application profile shall have a point of contact responsible for maintaining this list. The contact shall serve as a registrar and maintain a repository including documentation for all of its common and user-specific Type-2 data fields. As additional fields are required by specific agencies for their own applications, new fields and definitions may be registered and reserved to have a specific meaning. When this occurs, the domain or application profile registrar is responsible for registering a single definition for each number used by different members of the domain or application profile. There may be more than one Type-2 record included in each transaction.

A Type-2 record contains numerous free-text fields and could contain significant amounts of hidden data, simply because some standards may support certain fields while others may not. A filter developer may wish to go so far as to treat every field in the Type-2 record as a block of free-text and inspect everything. A filter developer should consult the actual specification for a full list of each field in this record.

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: Type-2 records contain multiple User-Defined fields for each registered Domain and/or APS. Collection systems may also attempt to insert User-Defined fields that are not defined in the registered Domain or APS.

1. **Validate:** Check data type and syntax for each field in each Type-2 record for compliance with field type (alphabetic, numeric, special characters, or binary content) according to registered Domain and / or APS.

2. **Validate:** For each field in a Type-2 record, check that they do not contains field(s) which are not defined in the registered Domain and/or APS.

3. **Remove:** Remove undefined fields not present in any EBTS specification supported by this ISG.

4. **Replace:** Replace potentially sensitive contents of Type-2 record fields with pre-approved replacement text for specific fields (including location and Operational Personnel Identifier).

5. **Remove:** Remove optional fields from the EBTS record; this may remove some useful information needed by some applications.

6. **External Filtering Required:** Extract the values of each Type-2 field and pass to an external filter, many of them are free-text.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:
Each Type-2 record contains mandatory and optional fields. Undocumented fields may be present at any location within the entire length of the record, designated by FS markers.

4.11.1 Domain-Specific Implementations

4.11.1.1 FBI EBTS 7.0-10.0 Domain Type-2 Fields

The FBI EBTS versions 7.0 through 10.0 include mandatory and optional fields that vary according to the Type-1 record Type of Transaction (ToT) data field. FBI EBTS [2] defines the field tag, field description, field length, and syntax requirements for over 1000 individual fields. The majority of the fields are optional; however, a subset of the fields must be inspected when present. A summary of the most common fields is provided below in Table 4-9. Highlighted fields represent data hiding opportunities due to large field lengths.

Table 4-9. Type-2 Record Fields

Tag (Field Name)	Record Number	Field Description	Allowed Values	Max Length
LEN	2.001	Logical Record Length	Numeric	7
IDC	2.002	Image Designation Character (IDC)	Numeric	2
FFN	2.003	FBI File Number	Numeric	10
RET	2.005	Retention Code	Alphabetic	1
ATN	2.006	Attention Indicator	Alphabetic, Numeric, or Special Characters (ANS)	30
ORI	2.009	Originating Agency Case Number	ANS	20
FBI/UCN	2.014	FBI Number	Alphabetic, Numeric	9
NAM	2.018	Name	Alphabetic, Special Characters	50
POB	2.020	Place of Birth	Alphabetic	2

CTZ	2.021	Citizenship	Alphabetic	2
DOB	2.022	Date of Birth	Numeric	8
SEX	2.024	Sex	Alphabetic	1
RAC	2.025	Race	Alphabetic	1
HGT	2.027	Height	Alphabetic, Numeric	3
HTR	2.028	Height Range	Alphabetic, Numeric	6
WGT	2.029	Weight	Numeric	3
WTR	2.030	Weight Ranges	Numeric	6
EYE	2.031	Color Eyes	Alphabetic	3
HAI	2.032	Hair Color	Alphabetic	3
PHT	2.036	Photo Available Indicator	Alphabetic	1
RFP	2.037	Reason Fingerprinted	ANS	75
DPR	2.038	Date Printed	Numeric	8
DOA	2.045	Date of Arrest	Numeric	8
ASL	2.047	Arrest Segment Literal	SET	
DOO	2.047A	Date of Arrest	Numeric	8
AOL	2.047B	Arrest Offense Literal	ANS	300
SRF	2.059	Search Result Findings	Alphabetic	
MSG	2.060	Status/Error Message	ANS	300
RAP	2.070	Request for Electronic Rap Sheet	Alphabetic	1
CRI	2.073	Controlling Agency Identifier	ANS	9
ERS	2.075	Electronic Rap Sheet	ANS	200,000
PEN	2.078	Penetration Query Response	Numeric	2

RISKS AND RECOMMENDATIONS:

Data Hiding and Data Disclosure: Sensitive data or information may be inserted and/or hidden in multiple fields. Specific fields of concern due to length are ATN, RFP, AOL, MSG, and ERS.

1. **Remove:** Remove optional Type-2 fields.

2. **External Filtering Required:** Extract the ATN, RFP, AOL, MSG, and ERS fields (if present) and pass to an external filter.

4.11.1.2 DoD EBTS 1.2 Domain Type-2 Fields

The DoD EBTS standard for version 1.2 implements a number of custom fields in a Type-2 records [4]. The following data in Table 4-10 provides a few Type-2 fields from this standard. This is not a comprehensive list of fields from the DoD EBTS 1.2 standard. Filter developers should understand that there could be many optional additional fields that contain a large amount of free text. In general, every Type-2 field should be treated as a location to hide data. Many other fields may contain codes such as an enumerated type, numbers, or dates, but a significant number of Type-2 fields simply contain free text.

Table 4-10. DoD EBTS v1.2 Type-2 Fields

Field Number	Field Description	Concern
2.301	Location	Contains the free-text location of the biometric collection.
2.302	Internment Serial Number	Contains serial number of a prisoner/detainee.
2.303	DoD Number	May contain prisoner Internment Serial Numbers
2.304	Passport Information	178 bytes of passport information (passport date, expiry, passport number, country, and location).
2.306	Geographic Coordinate Latitude/Longitude	Covered by Geolocation construct. This field shall undergo inspection according to the EBTS Construct 4.12.
2.318	XML-Based Rap Sheet	Contains text fields with unknown length.
2.319	Name of Latent Technician	Name of the person that captured the biometric data. This could contain sensitive data about the capture.
2.323-2.331	Miscellaneous family name fields	Numerous free-text fields for family and relative names.

RISKS AND RECOMMENDATIONS:

Data Hiding and Data Disclosure: Many optional fields in the Type-2 record for DoD EBTS 1.2 contain free text. This could be a location to hide information if the receiving end does not recognize these fields but it may also contain accidently disclosed information if used properly.

1. **Remove:** Remove optional fields from the Type-2 record.

2. **External Filtering Required:** Pass free-text fields in a Type-2 record to an external filter.

PRODUCT: EBTS DOD 1.2 AND ABOVE

LOCATIONS:
Type-2 records will be located in the file according the CNT field in the Type-1 record. All custom DoD EBTS 1.2 fields are above 2.300. Type-2 records for DoD EBTS 1.2 will still implement the lower field numbers covered in previous sections for ANSI/NIST.

4.11.1.3 DoD EBTS 2.0 Type-2 Fields

DoD EBTS domains generally follow FBI EBTS field names for items 2.001-2.075. However, DoD EBTS files contain fields with significantly different content for fields higher than 2.075. Notable fields from the Version 2.0 standard are listed in Table 4-11 below. This is not a comprehensive list, and there are other fields not listed that may introduce a hidden data risk.

In the EBTS DoD Version 2.0 standard [5], there are several new fields added to the Type-2 record in the 2.8000 and above range. Every field in that range is optional and many may contain a significant amount of metadata information.

Table 4-11. DoD EBTS v2.0 Type-2 Fields

Record Number	Field Description	Concern
2.8000-2.8016	Biometric Subject Metadata Fields	Various fields identifying metadata about the subject including address, birthday, citizenship, height, weight, vital status, blood type, marital status, etc. Most fields are free text information.
2.8017	Biometric Subject Associated Individual	Multiple occurrences of this field may exist listing associated individuals. This may be free-text information or contain sensitive information.
2.8018	Biometric Subject Group Membership	Identifies the associations of a subject with a group, there can be multiple occurrences of this field for multiple groups. This may contain free-text or sensitive information of 100 bytes per occurrence.
2.8019	Collected identification	Contains information about how the biometric data was collected including organization, identifier, and a comment field for free-text.
2.8022	Biometric Subject Compartments	Contains the compartment that the subject is cleared for (or claims to be cleared for). The value may be any text information. It also contains a subfield for validity (CLA = claimed, VER = verified).
2.8100	Collection location	This field contains 250 bytes of data identifying where the biometric data was collected. There are several subfields defining the address value and address type. The address value contains free-text.

RISKS AND RECOMMENDATIONS:

Data Hiding and Data Disclosure: Many optional fields in the Type-2 record for DoD EBTS 2.0 contain free text. This could be a location to hide information if the receiving end does not recognize these fields, but it may also contain accidently disclosed information if used properly.

1. **Remove:** Remove optional fields from the Type-2 record.

2. **External Filtering Required:** Pass free-text fields in a Type-2 record to an external filter.

PRODUCT: EBTS DOD 2.0

LOCATIONS:
Type-2 records will be located in the file according the CNT field in the Type-1 record. All custom DoD EBTS 2.0 fields are above 2.8000. Type-2 records for EBTS 2.0 will still implement lower field numbers covered in previous sections.

4.12 Common EBTS Constructs

The following section identifies common fields within many EBTS records. Rather than documenting and addressing each of these fields individually within each record construct, they are discussed once and apply for each instance discovered across all EBTS records.

4.12.1 Image Designation Characters

OVERVIEW:
The Image Designation Character (IDC) field is present in all records. It is an identification field, or a unique number, that is used to define the record. In Type-1 records, more specifically in the CNT field, there is a table that defines all the records in the file and their IDC field values. The NIST standard defines the IDC field as a "mandatory ASCII field [that] shall be used to identify the fingerprint image data contained in the record. This IDC shall match the IDC found in the file content (CNT) field of the Type-1 record. "An IDC is an identification field that relates to each record in the EBTS file. For every entry in the lookup table in the Type-1 record, the IDC field should match the respective IDC field that is defined in the actual record. The IDC field in each EBTS record is the second field in every record.

RISKS AND RECOMMENDATIONS:
Data Hiding: Although a minimal risk, a data hiding risk could arise when the IDC values are incorrect. Some parsers may handle the problem differently, however, the worst case is an ignored record due to an invalid IDC value.

1. **Validate:** Check that the lookup table IDC values match the records in the rest of the file.

2. **Remove:** Remove records and IDC entires that have a mismatch in the IDC value from the lookup table. This may require rebuilding the entire EBTS file.

3. **Reject:** Reject files that contain invalid IDC fields.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:
The IDC field is present as a subfield in the CNT field in a Type-1 record. Each EBTS record contains an IDC field that is typically the second field in the record.

4.12.2 Agency and Source Information

OVERVIEW:
There are many record types, in several different standards of EBTS, which provide some type of agency identifier. For example, a Type-1 record (defined in later constructs) implements two fields: an Originating Agency Identifier and a Destination Agency Identifier (similar to a TO/FROM tag in a message). These fields are free-text and allow users to document the users of this biometric data. A valid value for this field can be "Not Specified," while other agencies or organization may have their own ways to list this information.

There are several different types of fields throughout many EBTS standards that provide this information. Fields such as Originating Agency Identifier, Destination Agency Identifier, Source Agency Field (SRC), Controlling Agency Identifier, Agency Names, Originating Agency Case Number, and Source Agency Name are present in various EBTS records. The most common field is the SRC field (Source Agency Field), which is located in most image record types.

RISKS AND RECOMMENDATIONS:
Data Disclosure: Any field that implements an agency identifier is considered free-text, where users could enter any information. If this field is implemented, it may reveal information not intended for release.

1. **Remove:** If the field in question is optional, remove the field from the record.

2. **Replace:** If the field in question is mandatory, replace it with "Not Specified".

3. **Replace:** Replace the field with a pseudonym or an approved replacement value.

4. **External Filtering Required:** This field is free-text, pass to an external text filter.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:
These fields are located throughout EBTS data in numerous records.

4.12.3 Record Hash Field

OVERVIEW:
A Hash (HAS) field was introduced in ANSI/NIST-2011; this allows many records to provide a SHA-256 hash of their record data (typically not the metadata, just the final data field). Some

examples of the Hash fields are listed in fields 10.996, 13.996, 14.996, and 15.996. It is a major concern for any filter developer to understand that a hash field exists and that any changes to the file require recalculation and updating the HAS field.

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: An incorrect hash may mean that data has been modified in transit.

1. **Validate:** Check that the hash field is the correct SHA-256 hash of the actual data.

2. **Replace:** Any changes to the data in the record will require replacing the hash with a correct value.

3. **Reject:** Reject the entire file with any invalid hash in any record.

PRODUCT: ANSI/NIST 2011 EBTS

LOCATIONS:

Located in only a few EBTS records, they are listed as XX.996 in their record/field number.

4.12.4 Geographic Sample Acquisition Location

OVERVIEW:

Implemented in ANSI-NIST-2011, many records provide an optional set of geolocation subfields. There are several geolocation subfields that are present throughout many record types. The subfields provide the time and location where biometric data was collected. There are a total of 15 subfields in the main field called the Geographic Sample Acquisition Location field, and a few subfields that are of interest are listed in Table 4-12. The Geographic Sample Acquisition Location field is referred by the mnemonic "GEO".

Table 4-12. Geographic Sample Acquisition Location Subfields

Subfield Name	Description	Length	Constraints
Latitude Degree value (LTD)	Specifies the degree of latitude. This value can be expressed as an integer or real number (decimals).	Undefined.	Value shall be inclusively between -90 and +90. If exists, LTM subfield must also exist.
Latitude Minute Value (LTM)	Specifies the minute of a degree. This value can be expressed as an integer or real number (decimals)	Undefined.	Value shall be between 0 (inclusive) and 60 (exclusive). If exists, LGM subfield must also exist.
Latitude Second Value (LTS)	Specifies the second of a minute. This value can be expressed as an integer or real number (decimals)	Undefined.	Value shall be between 0 (inclusive) and 60 (exclusive). If this field exists, the LGS field must also exist.

Longitude Degree Value (LGD)	Specifies the degree of a longitude. This value can be expressed as an integer or real number (decimals).	Undefined.	Value shall be inclusively between -180 and +180. If the LTD field is present, the LGD must also be present.
Longitude Minute Value (LGM)	Specifies the minute of a degree. This value can be expressed as an integer or real number (decimals)	Undefined.	Value shall be between 0 (inclusive) and 60 (exclusive). If exists, LTM subfield must also exist.
Longitude Second Value (LGS)	Specifies the second of a minute. This value can be expressed as an integer or real number (decimals)	Undefined.	Value shall be between 0 (inclusive) and 60 (exclusive). If exists, LTS subfield must also exist.
Elevation (ELE)	A numeric value expressed in meters.	Undefined.	Minimum: -422 meters Maximum: 8848 meters
Geodetic Data Code (GDC)	Alphanumeric value to indicate the coordinate system for latitude and longitude.	6 bytes.	Table 6 of the ANSI/NIST 2011 ITL provides a whitelist of possible values. [2]
Geographic Coordinate Universal Transverse Mercator Zone (GCM)	Alphanumeric value to specific a coordinate location in UTM.	3 bytes.	xxC-X (omitting I and O). Where xx is a one-two digit UTM zone number.
Geographic Coordinate Universal Transverse Mercator Easting (GCE)	Integer value	1-6 digits.	Range limited by 6 digit number.
Geographic Coordinate Universal Transverse Mercator Northing (GCN)	Integer value.	1-8 digits.	Range limited by 8 digit number.
Geographic Reference Text (GRT)	Alphanumeric entry that can be free-text.	150 bytes	No constraints, other than an alphanumeric field.
Geographic Coordinate Other System Value (OCV)	Used for free text depending on the OSI subfield defined before.	126 bytes	Must be present if OSI field is present.

RISKS AND RECOMMENDATIONS:

Data Attack, Data Hiding, and Data Disclosure: Geographic subfields and metadata contain numerical values that could be used as an overflow or for targeting a system that does not perform bounds checking. Many fields are undefined in length or contain hundreds of bytes of data, a place where information could be hidden. Geographic metadata about where the biometric data was collected might also be inadvertently placed within the file and could be sensitive.

1. **Validate:** Check that the Universal Time Entry field is a valid time.

2. **Validate:** Combine the implemented latitude and longitude subfields and check that the values are within an acceptable range or specified region.

3. **Validate:** Combine the implemented latitude and longitude subfields and check that the values are within the possible ranges defined in this construct.

4. **Validate:** Combine the implemented Mercator zone fields and check that the values are within an acceptable range or specified region.

5. **Validate:** Combine the implemented Mercator zone fields and check that the values are within possible ranges.

6. **Validate:** Check that the elevation value is within the range specified above.

7. **Remove:** Remove all the Geographic Sample Acquisition Location fields and subfields.

Data Hiding and Data Disclosure – Free text fields may contain either hidden information or may contain sensitive data not seen in the file.

8. **Remove:** Remove the Geographic Reference Text subfield as it may contain free text.

9. **External Filtering Required:** Pass the contents of the Geographic Reference Text subfield to an external filter; this content can be free text.

10. **Remove:** Remove the OCV field as it may contain free text.

11. **External Filtering Required:** Pass the contents of the OCV subfield to an external filter, this content can be free text.

PRODUCT: ANSI/NIST 2011 EBTS

LOCATIONS:
Geographic sample acquisition location fields (GEO) are used in most Record Types-10 and above. They are embedded within records as an optional field (xx.998, where xx is the record number).

4.12.5 Annotation Information

OVERVIEW:
A new field called Annotation Information (ANN) was implemented in ANSI/NIST-2011. There are several records that can implement this field. Some example record/field numbers are 9.902, 10.902, and 13.902 (all are in the format of XX.902). It is an optional field, "used to store annotation, logging, or processing information...", that is typically associated with the biometric

data represented in that record [2]. It contains four sequential fields: the first is the Greenwich Mean Time (GMT), the second is the processing algorithm name or version, the third is the owner of the algorithm, and the fourth field (255 bytes) contains the process description. The last field can be considered free-text.

There is more Annotation Information defined in a Type-9 record: 9.901. This field contains Universal Latent Annotation or Universal Latent Workstation Annotation Information. It is an optional field that provides logging or information related to how the information was processed. It contains information such as: how the image was imported (including the file path); how it was cropped and saved into a new file; and the checksum of the image, before and after any type of processing done by the workstation.

RISKS AND RECOMMENDATIONS:

Data Disclosure: The last subfield in the Annotation Information contains 255 bytes of free-text; it may contain hidden information or information regarding a process that could be accidently disclosed. The 9.901 field contains processing steps and filepath information about the processing of the data.

1. **Validate:** Check that the XX.902 Annotation Information follows the four subfield format.
2. **Remove:** Remove the Annotation Information field from the record.
3. **Remove:** Remove the Universal Latent Annotation field from the record.
4. **External Filtering Required:** Pass the entire annotation field to an external filter.
5. **External Filtering Required:** Pass the entire Universal Latent Annotation field and its subfields to an external filter.

PRODUCT: ANSI/NIST 2011 EBTS

LOCATIONS:

Located in only a few EBTS records, they are listed as XX.902 or 9.901 in their record/field number.

4.12.6 Comments

OVERVIEW:

Many EBTS records provide a field for a user to supply comments. Image records often provide a field for text comments to allow an operator to add any free-text they wish to describe the content in the image. A comment field is often over 100 bytes in length and may contain any content. There are many comment fields and subfields (comments nested within an EBTS field) that are present throughout numerous EBTS record types. This section applies to any comment located in any EBTS record. Comment fields do not have a consistent field number across records, but many records list comments as field number 20 (xx.020).

RISKS AND RECOMMENDATIONS:

Data Hiding and Data Disclosure: Comments are free-text and allow someone to hide information in the field or could accidently disclose information not intended for release.

1. **Validate:** Check that the comment field and its characters are text from the right character set. If a Unicode encoding is supported by field 1.015, it can be a block of Unicode text.

2. **Remove:** If the Comment field is optional, remove the field or subfield from the record.

3. **Replace:** If the Comment field is mandatory, replace with an empty string or pre-approved value.

4. **External Filtering Required:** Pass the free-text comment to an external text filter.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

These comment fields and subfields are located throughout EBTS data in numerous records.

4.12.7 User Defined Fields

OVERVIEW:

User-defined fields are located in many different EBTS records. They are often labeled as UDF fields in many EBTS standards, and they are often defined as a range of several hundred field numbers or a block of user-defined fields. User-defined fields may indicate that the field may be present but only in certain EBTS implementations. Each standard may define fields located in a block of user-defined fields. There are some standards that will not implement user-defined fields from another standard. Inspecting these fields depends on the filter and its intended version of EBTS. A filter should be cautious when handling user-defined fields, because removing these fields may have an impact on other EBTS viewers. If a filter is only handling ANSI/NIST files, the appropriate ANSI/NIST standard should be consulted to determine which fields are User-Defined Fields.

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: If any undefined fields exist (i.e., a field within the user-defined range defined by the standard, but not in use by the actual standard that this file implements), then it is likely the sign of a bad EBTS writer, hidden data, or possibly malicious content in the EBTS file.

1. **Validate:** If an ANSI/NIST user-defined field is present, then check that it is defined in the appropriate specification for the correct environment (DoD version, FBI version, etc.). This will mean that the field has a definition in one of the applicable specifications.

2. **Remove:** Remove any undefined (i.e., not documented) field in the EBTS record.

3. **Reject:** Reject any EBTS file with any invalid or unknown user-defined fields.

4.12.8 Reserved Fields

OVERVIEW:

Many different EBTS records contain fields that are defined as "Reserved" or "Reserved for Future Use." These are different from user-defined fields, because user-defined fields may be implemented by other specifications of EBTS. Reserved fields have not been formally implemented in any specification, but the field numbers are listed as reserved. They should not be implemented in any EBTS file. This may require keeping track of different standard versions, as reserved fields in one standard might be implemented in later versions of the standard.

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: If any "Reserved" fields are implemented in a file it is likely the sign of a bad EBTS writer, hidden data, or possibly malicious content in the EBTS file.

1. **Reject:** Reject any EBTS file with any "Reserved" field implemented.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Reserved fields are located in numerous fields in numerous EBTS records throughout the file.

4.13 TYPE-3-6 Records

OVERVIEW:

Type-3 through Type-6 record types contain binary fingerprint images. Types-3, 5, and 6 are considered deprecated, leaving the Type-4 as the most common binary fingerprint record found in EBTS data. All four record types use the same binary field structure, so they are consolidated in this section. Unlike Type-1 and Type-2 records, Types-3-6 records are entirely in binary and their fields are shown below [1]. They do not implement the tagged-field metadata structure that is present in many other EBTS records; their metadata is simply in byte fields defined by the standard. A summary of the fields is shown in Table 4-13 below.

Table 4-13. Type-3-6 Record Layout [1]

Field Tag	Field Description	Byte Count	Byte Position
LEN	Logical Record Length	4	1-4
IDC	Image Designation Character (IDC)	1	5
IMP	Impression Type	1	6
FGP	Finger position	6	7-12
ISR	Image scanning resolution	1	13
HLL	Horizontal Line Length	2	14-15
VLL	Vertical Line Length	2	16-17
GCA/BCA	Compression algorithm	1	18

DATA	Image Data	<LEN>-18	19-<LEN>

The Image resolution is provided in Type-3-6 records, as well as the size in the VLL and HLL fields. This identifies the height and width of the fingerprint image. This information is also available in the data format, unless the image is uncompressed in a raw form. Compressed images should not solely rely on these fields for image rendering, as this information should be in the DATA field. Uncompressed images must rely on these fields since the image dimensions are not known with simple raw pixels. The ANSI/NIST 2007 version of the specification indicates that when an image is uncompressed "each pixel of the uncompressed grayscale image shall be quantized to eight bits (256 gray levels) contained in a single byte" [1] and each byte is left justified.

RISKS AND RECOMMENDATIONS:

All the recommendations from construct 4.9 apply to this construct as well.

Data Attack and Hiding: Type-3, 5, and 6 records have been deprecated. Any of those records that are included could be malicious content or hidden data.

1. **Remove:** Remove all Type-3, 5, and 6 record types.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-3-6 records will be located in the file according the CNT field in the Type-1 record.

4.14 TYPE-7 Record

OVERVIEW:

A Type-7 record contains a user-defined image. It was designed as a temporary measure to support exchanging user-defined images that did not fit into the categories of other records. It is still used to exchange arbitrary user-defined image data. The record contains two mandatory fields: the logical record length and the IDC. The remainder of the Type-7 record's fields are completely user-defined.

RISKS AND RECOMMENDATIONS:

All the recommendations from construct 4.9 apply to this construct as well.

Data Hiding, Data Disclosure, and Data Attack: Type-7 records provide arbitrary image data which may introduce each type of risk. Since Type-7 records and many of its fields are not formally defined they may be an attempt at hidden data. It is also a binary image record which may present an attack risk.

1. **Remove:** Remove the Type-7 record from the EBTS file.

2. **External Filtering Required:** Pass the Type-7 record data to an external filter, if possible to determine the file type.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:
Type-7 records will be located in the file according the CNT field in the Type-1 record.

4.15 TYPE-8 Record

OVERVIEW:
A Type-8 Record is a binary-only record that contains a scanned or a vectored signature image. The following fields shown in Table 4-14 are defined in a Type-8 record according to the ANSI/NIST specification.

Table 4-14. Type-8 Record Fields [1]

Tag (Field Name)	Field Description	Byte Count	Byte Position
LEN	Logical record length	4	1-4
IDC	Image designation character	1	5
SIG	Signature type	1	6
SRT	Signature representation type	1	7
ISR	Image scanning resolution	1	8
HLL	Horizontal line length	2	9-10
VLL	Vertical line length	2	11-12
DATA	Signature image data	<LEN>-12	13-<LEN>

An important field from this table above is the Signature representation type (SRT) because it indicates the format of the image data. This field equals zero if the image is scanned and not compressed. When uncompressed, the data is packed at eight pixels per byte. If the image is scanned and compressed, then the SRT field is equal to one. If the SRT field contains the value '1', then the image data conforms to the ANSI/EIA-538-1988 facsimile compression algorithm. If the image is vector data, then the SRT field is equal to two.

The vector data is a custom implementation that provides a list of vectors. This information is unique and is not addressed by construct 4.9. Each vector in the list is 5 bytes long and contains three parts: the first is a 2-byte X coordinate value, the second is a 2-byte Y coordinate value, and the third is a 1-byte pen pressure value. When the pen pressure value is 0, there is no line (the pen is lifted up). The value can range from 1 to 254 for different levels of pressure. A value of 255 is special and is used to terminate the vector list. The X, Y values are present in a space that covers 1000 mm^2. The specification only mentions that the X and Y axis have units in terms of 0.0254 mm. In the case of a perfect square for the vector region, each X, Y dimension is approximately 31.622776mm. The specification is clear that each unit in either axis is 0.0254 mm.

So the X and Y values could reach 1,245. (1,245 * 0.0254 = 31.623). These are calculated values from information given in the specification.

The ANSI/NIST specification indicates that there are typically up to 2 Type-8 records: one for the person supplying the fingerprint (suspect) and the other that is recording the fingerprint (captor).

RISKS AND RECOMMENDATIONS:

Since a Type-8 record is an image record, all the recommendations from construct 4.9 apply as well.

Data Attack and Data Hiding: An invalid field regarding the image type might be an attack or an attempt to hide or obscure the image data.

1. **Validate:** Check that the SRT field matches the appropriate data in the Type-8 record.

2. **Validate:** Check the vector list contains 5 byte entries and terminates with a value of 255 for the final pen pressure value.

3. **Validate**: Check each entry in the vector list for correctness: a correct and within range (X, Y) coordinate and a value of 0-254 for pen pressure (unless the final entry which shall terminate the list with a value of 255).

Data Attack and Data Hiding: Additional Type-8 records may be the sign of malicious content or hidden data.

4. **Validate:** Check that there are up to 2 Type-8 records in the EBTS file.

5. **Reject:** Reject files with more than 2 Type-8 records.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-8 records will be located in the file according the CNT field in the Type-1 record.

4.16 TYPE-9 Record

OVERVIEW:

The NIST specification states that Type-9 records "shall contain and be used to exchange geometric and topological minutiae templates and related information encoded from a finger or palm." A Type-9 record contains ASCII data and implements the Tagged-Field metadata structure. A record can exist for each finger present in the EBTS file. For example, ten Type-9 records may exist within a tenprint EBTS file (ten fingerprints). There can also exist up to 8 Type-9 records for 1-2 palmprints listed in the EBTS file.

Type-9 records are critical for fingerprint matching; the standard format for minutiae data is organized as X, Y, Theta (θ) values. These three values define the location at the coordinates X, Y (ranging from 0 upward) and a Theta value (ranging from 000-359). Fingerprint matching uses the minutiae data points to match as many points as possible to guarantee a higher level of certainty for a match. ANSI/NIST EBTS files implement the standard minutiae data in field 9.12, while others may use other custom fields and custom minutiae data for their specific applications.

Type-9 records define minutiae data, which is commonly stored in a "standard format"--an encoded table of tagged fields. There are four fields that apply to every Type-9 record and then seven more fields for "Standard Format Features." There can be an additional 163 vendor specific or custom tagged fields (175 fields in total). Each flavor of the specification can implement different fields. A summary of the Type-9 record fields are shown below in Table 4-15.

Table 4-15. Type-9 Record Fields [1]

Field Numbers	Implementations
9.1-4	ALL
9.5-12	Standard Format Features
9.13-30	IAFIS Features
9.31-55	Cogent Systems Features
9.56-70	Motorola Features
9.71-99	Sagem Morpho Features
9.100-125	NEC Features
9.126-150	M1-378 Features
9.151-175	Identix Features

The mandatory first four fields in a Type-9 record as shown below in Table 4-16:

Table 4-16. Mandatory Type-9 Record Field Descriptions

Tag (Field Name)	Record Number	Field Description	Allowed Values
LEN	9.001	Logical Record Length	Numeric
IDC	9.002	Information Designation Character	Numeric
IMP	9.003	Impression Type	1-2 byte ASCII Values (range from 0 – 29)
FMT	9.004	Minutiae Format	S or U (S- Standard Minutiae) (U-Vendor Specific)

RISKS AND RECOMMENDATIONS:

Data Hiding: A record that does not adhere to the correct format could be hidden data.

1. **Validate:** Check that the first four fields (9.1 – 9.4) are present at the very beginning of the Type-9 record. They are mandatory for all EBTS data.

2. **Validate:** Check that the first four fields (9.1-9.4) have an appropriate value as defined from the table above.

3. **Remove:** Remove any field numbers that should not exist or overlap with another vendor specific feature set. The Standard, IAFIS, and one vendor set can be present in a Type-9 record. Remove any extra vendor set that does not belong.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:
Type-9 records will be located in the file according the CNT field in the Type-1 record.

4.16.1 TYPE-9 Records – General Minutiae Data

OVERVIEW:
Vendors may implement different types of minutiae data, some in their own data format. This section highlights an example to understand the complexity of filtering minutiae data. In the example from the ANSI/NIST ITL 2007 standard, the field 9.012 (Minutiae and Ridge Count Data (MRC)) is implemented. Other feature sets are available and may appear in other EBTS files. The next construct illustrates the DoD version of minutiae data.

A Type-9 record may describe numerous minutiae data points, each defining a location used for fingerprint matching. A basic single block of fingerprint minutiae data consists initially of an index number, as is incremented for each block. Following the index number are the X, Y, and Theta (θ) values. X and Y are two four-digit positive values that relate to a location on the fingerprint image. Theta is a value between 000 and 359, representing the angle that the ridge follows. The next piece of information is a quality measure, which is a value ranging from 0 to 63. The next field defines the minutiae type (Ridge Ending, Bifurcation, Compound, or undetermined). This is a code value (letter) from the ANSI/NIST standard. The last portion of information is called the Ridge count data, which can be a series of information items.

RISKS AND RECOMMENDATIONS:
Data Attack and Data Hiding: While not observed, it could be possible for an attacker to attempt to provide false values in minutiae data to target EBTS readers. Since there are many minutiae data points, small bits of information could be hidden within each point similar to steganography in images.

1. **Validate:** Check that each minutiae data point has been properly formatted.

2. **Validate:** Check that each minutiae data point (x,y) is located within the fingerprint image.

3. **Validate:** Check that the number of minutiae field is accurate with the minutiae data.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Minutiae data will be embedded into possibly serveral fields within a Type-9 record.

4.16.2 TYPE-9 Records – DoD EBTS v2.0 Minutiae Data

OVERVIEW:

In the DoD version 2.0 of the EBTS standard, there are several fields that are added to define minutiae data. In the DoD version of EBTS, the field number 9.137 contains the finger minutiae data. Field number 9.138 contains the Ridge Count information and field number 9.139 contains Core Information. Delta information is found within field number 9.140.

The finger minutiae data field contains the following subfields: minutia index number, minutia X coordinate, minutia Y coordinate, minutia angle, minutia type, and quality measure.

The ridge count information field contains a Ridge Count Extraction method, followed by two zeros, or it contains the Minutia Index Numbers (x2) and a subfield defining the Number of Ridges.

Both the Core and Delta information fields define an X, Y, and Theta (θ) value as part of its field for a Type-9 record.

Lastly, there is a Number of Minutiae field (9.136) that contains the number of data points. This is important because it defines how many fields follow this field in the Type-9 record.

All of these fields are critical to support fingerprint matching. They are all mandatory fields that must be validated for correctness.

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: While not observed, it could be possible for an attacker to attempt to provide false values in minutiae data to target EBTS readers. Since there are also many minutiae data points, small bits of information could be hidden within each point similar to steganography in images.

1. **Validate:** Check that each minutiae data point has been properly formatted.

2. **Validate:** Check that each minutiae data point (x, y) is located within the fingerprint image.

3. **Validate:** Check that the number of minutiae field is accurate with the minutiae data.

PRODUCT: DOD EBTS VERSION 2.0

LOCATIONS:

Minutiae data for EBTS DoD Version 2.0 is primarily located in the fields 9.136 – 9.140. There may be other related minutiae data fields located in Type-9 records.

4.17 TYPE-10 Record

OVERVIEW:

Type-10 records contain facial or SMT information in the form of greyscale or color images. Information about the image is also part of the record and listed in the ASCII format in numerous tagged-fields.

Type-10 records implement additional fields to define more information about the image supplied in the data. Some fields can be a source of hidden or disclosed data, while others could be used in an attack to target image-processing software.

There are several fields that are listed as reserved for future definition (10.014-10.015, 10.018-10.019, 10.031-10.039, and 10.044-10.199). These fields are not implemented in the ANSI/NIST standard and could be used as a source for hidden data since they are undefined and ignored by many EBTS viewers.

The ANSI/NIST standard also defines several user-defined fields, which are different from those reserved for future definition. They are listed as 10.200-10.998. They present some issues, as the ANSI/NIST standard does not define the format of these fields, which means that it could provide a source of hidden data. They may also be used by other standards outside the scope of ANSI/NIST. (e.g., 10.998 is the GEO field is not listed under the NIST standard). Type-10 records will be discussed later in more specific EBTS constructs.

Type-10 records provide fields for description of the image content. An example field, such as 10.022, provides the “PHOTO DESCRIPTION”, a 196 byte text field where information about the photo such as GLASSES, HAT, or SCARF is listed to help describe the subject in the photo. As this could be a source of hidden or disclosed data, it should be a concern for a filter. The following data in Table 4-17 lists several fields in the Type-10 record that provide textual (alpha character set) description of the SMT image provided in the record.

Table 4-17. Type-10 Record Fields

Field Number	Field Name	Maximum Number of Bytes	Description	Character Set
10.022	Photo Description	196	Contains subfields with attribute codes such as GLASSES, HAT, SCARF, or OTHER	Alpha
10.026	Subject Facial Description	1057	Lists code to describe the face of the subject such as NEUTRAL, SMILE, TEETH VISIBLE, LEFT EYE PATCH, or just unformatted text, etc.	Alpha
10.029	Facial Feature Points	1591	An optional ASCII field	Numeric

			used for exchanging facial image data. Consists of x, y coordinates and a feature point code.	
10.042	SMT Descriptors	466	A list of codes that identify the scar, mark, or tattoo. For example, the string “TATTOO FLAG USA” is a tattoo of the American flag, located by the field in 10.040.	Alphanumeric
10.043	Colors Present	196	Contains a series of subfields with the colors listed in the tattoo.	Alpha

The actual image field is located in 10.999 (Image Data). The size of the image itself has a maximum byte count equal to (6*HLL*VLL) + 8. An uncompressed color image can be expressed as 24-bit or 48-bit per pixel (6 bytes). Since each pixel can be up to 6 bytes in size, the maximum size is equal to (6*HLL*VLL) + 8. The Horizontal Line Length (HLL) is defined in the field 10.006 and the Vertical Line Length (VLL) is defined in 10.007. These fields have already been addressed in the Generic Image Records construct, but the specification provides a formula to compute the maximum data size specifically for Type-10 records. Compressed images should yield a smaller data size than the uncompressed data types.

RISKS AND RECOMMENDATIONS:

All the recommendations from construct 4.9 apply to this construct as well.

Data Hiding and Data Disclosure: The fields listed in the table above provide some level of free text information where information could be disclosed or purposely hidden.

1. **Validate:** If a list or code is available for a field in a Type-10 record, check that the code present is on the whitelist provided by the standard.

2. **Remove:** Remove all optional description fields from the Type-10 record.

3. **External Filtering Required:** Pass all textual description fields to an external filter.

Data Attack and Data Hiding: Invalid length fields may be the sign of an attack or an attempt to hide information.

4. **Validate:** Check that the size of the image is at a maximum equal to (6*HLL*VLL)+8.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-10 records will be located in the file according the CNT field in the Type-1 record.

4.18 TYPE-11 and 12 Records

OVERVIEW:

According to the 2007 ANSI/NIST specification, Type-11 and Type-12 records are reserved for future use. In the 2011 version of the NIST specification, Type-11 records are to store voice data and Type-12 records are to store dental records. At the current time of this ISG, neither record types are implemented or formally defined yet.

RISKS AND RECOMMENDATIONS:

Data Hiding: Any EBTS file that contains records that are not formally defined could be an attempt to hide data.

1. **Remove:** Remove Type-11 and Type-12 records.

2. **Reject:** Reject files that contain Type-11 or Type-12 records.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-11 and Type-12 records could exist in between any valid record and could possibly (or not) be listed in the CNT field in a Type-1 record.

4.19 TYPE-13 Variable-Resolution Latent Image Records

OVERVIEW:

A Type-13 record contains image data that has been acquired from latent fingerprint or palmprint images. The record itself is similar to other image records which provide a number of tagged-fields and metadata, following by a binary image. Most of the Type-13 fields have been addressed in the earlier section, but this record implements unique fields which can store a considerable amount of data. Two unique fields that can be extracted are the Search Position Descriptors field (total 82 bytes), and the Latent Quality Metric fields (total 156 bytes).

The Search Position Descriptors field is an optional ASCII field that defines an image code for the latent fingerprint image. The code is useful for limiting database searches when comparing against the latent image. The field contains a number (0-10) followed by one of the following codes: EJI, TIP, FV1, FV2, FV3, FV4, PRX, DST, and MED. These codes are defined by the ANSI/NIST standard in Table 32 [1].

The Latent Quality Metric field is an optional ASCII field that implements four items or subfields. The first item is a code from the ANSI/NIST standard. The other three items provide the algorithm in use and the actual quality score. The second field contains an ASCII representation of an integer value (0-100) for the score. The third item provides the vendor ID, which identifies the algorithm in use. The fourth item provides a numeric product code assigned by the vendor; this can distinguish which algorithm from the vendor ID (third field) is in use. The third and fourth fields combined identify the full details of the algorithm that was used.

RISKS AND RECOMMENDATIONS:

All the recommendations from construct 4.9 apply to this construct as well.

Data Hiding: Some fields in Type-13 records can provide optional information, which could be used to hide data.

1. **Validate:** Check that the Search Position Descriptors field contains a valid number and valid code from the standard.

2. **Validate:** Check that the Latent Quality Metric field contains four subfields with each item having the correct content type (a valid numeric or valid code from the vendor).

3. **External Filtering Required –** Pass the entire Search Position Descriptors field to an external filter as it may contain free text instead of the number/code combination.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-13 records are located as defined in the CNT field in the Type-1 record.

4.20 TYPE-14 Variable-Resolution Fingerprint Image Records

OVERVIEW:

A Type-14 record contains variable-resolution fingerprint images, used to exchange data from a rolled tenprint (all fingers), identification flat, or a major case print. The record itself is similar to other image records which provide a number of tagged-fields and metadata, followed by a binary image. Most of the Type-14 fields have been addressed in construct 4.9, but this construct implements additional unique fields.

The Fingerprint segment position field is an optional field with an undefined size present in Type-14 records. It is an ASCII field that contains offsets to locations within the images to indicate where the fingerprints are located. There can be multiple sets of offset values depending on the number of algorithms used to segment the image.

The Segmentation Quality Metric field is an optional ASCII field that contains four subfields. The first is the finger number (0-10). The second is the score (0-100). The third and fourth are vendor ID and product code that specify the vendor and the quality algorithm that was used.

The Fingerprint Quality Metric field is an optional ASCII field that contains four subfields. It is identical in structure to the Segmentation Quality Metric field.

RISKS AND RECOMMENDATIONS:

All the recommendations from construct 4.9 apply to this construct as well.

Data Hiding: The Fingerprint segment position field, Segmentation and Fingerprint Quality Metric fields are all optional ASCII fields that may contain hidden information.

1. **Validate:** Check that the Quality metric fields have the correct four subfield structure and the first two subfields contain the appropriate integer values.

2. **External Filtering Required:** Pass the fingerprint segment position field and both quality metric fields to an external filter.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-14 records are located as defined in the CNT field in the Type-1 record.

4.21 TYPE-15 Variable-Resolution Palmprint Image Records

A Type-15 record contains variable-resolution palmprint images. The record itself is similar to other image records which provide a number of tagged-fields and metadata, followed by a binary image. Most of the Type-15 fields have been addressed in the earlier section, but this record implements some unique fields. There are a number of small fields, but one to consider, that can store a considerable amount of data, is the Palmprint Quality Metric field. The EBTS file may contain up to four of these fields, each of 38 bytes in length. Combined with the field identifier, it is 159 bytes total. The field consists of ASCII characters that identify the score of the image and the algorithm that was used.

The Palmprint Quality Metric field is similar to other quality metric fields present in this section. It contains four subfields. The first subfield is the palm code from Table 35 of the ANSI/NIST Standard. The second subfield is an integer between 0 and 100. The third and fourth subfields contain the vendor ID and a product code from the vendor indicating the quality algorithm.

RISKS AND RECOMMENDATIONS:

All the recommendations from construct 4.9 apply to this construct as well.

Data Hiding: A Type-15 record could contain hidden information in the Palmprint Quality metric field since the field may contain any information.

1. **Validate:** Check the format of the Palmprint Quality metric field as it should contain four subfields, the first subfield a valid palm code, the second containing an integer between 0 and 100 and the other two fields are vendor supplied IDs.

2. **External Filter Required:** Pass the Palmprint Quality metric field to an external filter.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-15 records are located as defined in the CNT field in the Type-1 record.

4.22 TYPE-16 User-Defined Testing Image Records

OVERVIEW:

A Type-16 record contains user-defined or miscellaneous images. The ANSI/NIST standard defines it to be for developmental purposes for incorporating new images into EBTS data. The ANSI/NIST standard does not indicate its true purpose. Before the ANSI/NIST 2007 standard, it was used for iris images by the DoD. Iris images are now officially located in Type-17 records. Type-16 records implement fields that have been presented previously for general image record types. A unique field that is present in Type-16 records is the User-defined Testing Image Quality Score. The ANSI/NIST standard defines this field into three subfields (separated by the Unit Separator character). The first field is number between 0 and 100 for the score. The second field is the ID of the vendor, and the third field is a numeric product code from the vendor to indicate which algorithm was used.

In older versions of EBTS (specifically DoD Version 1.2), a Type-16 record was used to transmit iris image data. In later versions of the standard, a Type-17 record was used for iris image data. This construct is introduced to instruct filter developers that they may need to cover both User-defined Testing images and iris images in Type-16 records. EBTS files may contain different Type-16 record implementations. The recommendations are provided in the respective Type-16 and Type-17 record sections, and a filter developer should adhere to those recommendations. They should understand that a parser may need to handle multiple record structure types. There are no recommendations or risks for Type-16 iris image records because they are already covered in Type-16 and Type-17 record construct sections.

RISKS AND RECOMMENDATIONS:

All the recommendations from construct 4.9 apply to this construct as well.

Data Hiding: A Type-16 record could contain hidden information in the User-defined testing image quality score field since the field may contain any information.

1. **Validate:** Check the format of the User-defined Testing Image Quality Score as it should contain three subfields, the first subfield containing an integer between 0 and 100 and the other two fields are vendor supplied IDs.

2. **External Filter Required:** Pass the User-defined Test Image Quality Score to an external filter.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-16 records are located as defined in the CNT field in the Type-1 record.

4.23 TYPE-17 Iris Image Records

OVERVIEW:

Type-17 records are similar to other image records that have been discussed in this section. However, they contain additional fields such as Rotatation Angle of Eye, Eye Color, and Rotation Uncertainty. All of these fields are under 12 bytes in length. There is one unique field with significant size in Type-17 records called the Make/Model/Serial Number field. It is an optional 160 byte length field that may contain alphanumeric and special characters. The ANSI/NIST standard defines the field with 3 subfields, separated by the 'US' character.

RISKS AND RECOMMENDATIONS:

All the recommendations from construct 4.9 apply to this construct as well.

Data Hiding: The Make/Model/Serial Number field may contain any information and could contain hidden data.

1. **Validate:** Check that the Make/Model/Serial Number field contains 3 subfields.

2. **External Filtering Required:** Pass the Make/Model/Serial Number field to an external filter.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-17 records are located as defined in the CNT field in the Type-1 record.

4.24 TYPE-18 Records

OVERVIEW:

Type-18 records were implemented to exchange DNA data. The draft format of the ISO/IEC 19794-14 DNA data interchange format is used in this record. Due to privacy concerns, the ANSI/NIST 2011 ITL uses the non-coding regions of DNA and the regions that contain the subject's genetic charactersistics or traits are avoided.

Type-18 records contain metadata and mitochrondial DNA data. The following data in Table 4-18 defines some of the fields and subfields that may contain free text or contain large amounts of data that require further inspection.

Table 4-18. Type-18 Record Fields of Interest

Field Name	Field Number	Subfield Name	Size (bytes)	Description
DNA Lab Setting Field	18.003	Point of Contact	200	Free text field
DNA Lab Setting Field	18.003	Name or Organization	unlimited	Free text field
Sample Donor Information	18.006	Sample Collection Location Description	4000	Free text field

Pedigree Information	18.009	Pedigree Comment	2000	Free text field
Sample Collection Method	18.012	N/A	255	Free text field
DNA Profile Data	18.015	Supplemental Message	100	Free text field
DNA Profile Data	18.015	DNA Profile Comment	100	Free text field
Mitochondrial DNA Data	18.017	Mito Control Region 1	646	Character string with values A, G, C, T or, a value from Table 84 in ANSI/NIST standard.
Mitochondrial DNA Data	18.017	Mito Control Region 2	976	Character string with values A, G, C, T or, a value from Table 84 in ANSI/NIST standard.
Electropherogram Description	18.019	Image Data Descriptor	200	Free text field
Electropherogram Description	18.019	Electropherogram data	Unlimited	Base-64 encoded data field.
Electropherogram Description	18.019	Electropherogram screenshot	Unlimited	Base-64 encoded data field.
Electropherogram Ladder	18.023	Ladder Image Data Descriptor	200	Free text field
Electropherogram Ladder	18.023	Ladder Electropherogram Data	Unlimited	Base-64 encoded data field.
Electropherogram Ladder	18.023	Ladder Elctropherogram Screenshot	Unlimited	Base-64 encoded data field.

RISKS AND RECOMMENDATIONS:

Data Hiding: DNA fields could be used to hide data if the information is not used. Base-64 encoded data fields could be used to store hidden data since they are not visible until decoded.

1. **Validate:** Check that both mitochrondrial DNA data fields contain valid values (A, G, C, T, or a valid value from Table 84 in the ANSI/NIST ITL-2011 standard).

2. **Remove:** Remove all optional base-64 encoded data fields such as the screenshot subfields.

3. **Remove:** Remove the entire Type-18 record as they are not widely used.

4. **External Filtering Required:** Pass all base-64 encoded data fields from Type-18 records to an external filter. The decoded data might be an image screenshot which would require further inspection.

Data Hiding and Data Disclosure: Any free text field in a Type-18 record could be used to hide data or accidently disclosure information.

5. **Remove:** If any of the above free text fields are listed as Optional, then remove them from the Type-18 record.

6. **External Filtering Required:** Pass the contents of all free-text fields listed in this section to an external filter.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:
Type-18 records are located as defined in the CNT field in the Type-1 record.

4.25 TYPE-19 Records

OVERVIEW:
Type-19 records contain plantar (foot) print image data along with image metadata. The image metadata has been discussed in construct 4.9; it is similar in Type-19 records. There are several unique fields that are implemented in Type-19 that have not been addressed in the Generic Image Records section (Section 4.9). This construct identifies some of the unique fields to a Type-19 record that do not exist in many other records.

Type-19 records provide a field called Make/Model/Serial Number. It contains 50 bytes per each subfield (Make, Model, and Serial Number), which combined is 150 bytes of free text.

There is an optional field within Type-19 records called the Friction Ridge – Plantar Print Quality metric. It is similar to other quality metric fields discussed but the ANSI/NIST standard provides contraints for this field and its subfields. There can be nine occurrences of this field, so the total contents of these fields may expand in size over the course of several fields. There are four subfields defined in Table 4-19 below along with acceptable range values from the ANSI/NIST standard.

Table 4-19. Friction Ridge Plantar Print Quality Metric Field

Field Name	Field Mnemonic	Field Data Type	Constraint
Friction Ridge Metric	FRMP	Number	60<=FRMP<=79

Position			
Quality Value	QVU	Number	0<=QVU<=100 or QVU=245 or 255
Algorithm Vendor Identification	QAV	Hexadecimal	0000<=QAV<=FFFF
Algorithm Product Idnetification	QAP	Number	1<= QAP<=65535

RISKS AND RECOMMENDATIONS:

All the recommendations from construct 4.9 apply to this construct as well.

Data Hiding: The Make/Model/Serial Number field may contain any information and could contain hidden data or accidently disclosed information in free text areas.

1. **External Filtering Required:** Pass the Make/Model/Serial Number field to an external filter.

Data Hiding: The Friction Ridge-Plantar Print Quality Metric field is optional and may not be implemented which could allow for it to contain any hidden data.

2. **Validate:** Check that the Friction Ridge-Plantar Print Quality Metric field adhers to the constraints from the table in this section from the ANSI/NIST standard.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-19 records are located as defined in the CNT field in the Type-1 record.

4.26 TYPE-20 Records

OVERVIEW:

Type-20 records contain the "source representation(s) from which other Record Types were derived" [2]. An example that is provided in the standard is a Type-20 record which contains a group photo consisting of several individuals. A subject's face is extracted from the group photo and stored in a Type-10 record; therefore, the Type-10 record is derived from the Type-20 record. Another example is a color image of an object in a Type-20 record with latent fingerprint images on the object. The fingerprint portion of the image is extracted into Type-13 records (Latent fingerprint records). The standard indicates that there could be several levels of derivation. There could be several higher level Type-20 records and several derived lower level Type-20 records.

Record Types-10 and greater, except for Type-18, can be derived from a Type-20 record. They implement a field (xx.997) that is labeled Source Representation (SOR). It contains two sub-items: the first is a mandatory source representation number which is an index number mapping to the Type-20 record. The second optional sub-item may contain relative information such as coordinates (for an image) or time (for audio/visual recording) that indicates when or where this data was derived.

Within the Type-20 record itself there are a few fields that are present in other constructs presented in this document. A Type-20 record may provide an image, and the fields for a Type-20 record may closely follow the structure defined in construct 4.9. All recommendations in that construct apply to this record type as well. There are several unique Type-20 record fields which are identified below.

A Type-20 record provides an Acquisition Source field, a mandatory field with optional subfields. At its simplest is an Acquisition Source Type, a mandatory code value indicating how the data in this record was retrieved (digital image, video frame, screen capture, mobile telephone, landline telephone, FM radio transmission, etc.). The optional subfields contain information about the acquisistion and can include data for the Analog to Digital Conversion, Radio Transmission Format Description, and Acquisition Special Characteristics, all of which are 200 bytes in length. A filter should treat these optional fields as free text fields as they may contain metadata about how the data was acquired.

Type-20 records implement a field called the Source Representation Format. It contains two sub-items: the first is a mandatory file type field that contains the suffix (such as JPG) if it is a known file type. The second subfield is free text that identifies decoding instructions. It contains up to 1000 bytes of free text.

Type-20 records provide a field called Make/Model/Serial Number. It contains 50 bytes per each subfield (Make, Model, and Serial Number), which combined is 150 bytes of free text.

A field exists in Type-20 records called the External File Reference field. It is 200 bytes in length and contains the URI of the data if the record itself does not provide the data.

RISKS AND RECOMMENDATIONS:

Data Hiding and Data Disclosure: The Acquisition Source field, Source Representation Format field, the Make/Model/Serial Number field, and the External File Reference field may contain any information and could contain hidden data or accidently disclosed information in free text areas.

1. **External Filtering Required:** Pass the Make/Model/Serial Number field to an external filter.

2. **External Filtering Required:** Pass the Source Representation Format field to an external filter.

3. **External Filtering Required:** Pass the Acquisition Source Type field to an external filter.

4. **External Filtering Required:** Pass the External File Reference field to an external filter.

Data Hiding: Invalid Source Representation Fields indicate that data is not related and could be the sign of hidden data.

5. **Validate:** Check that the link between Type-20 records and their derived records is properly defined.

6. **Remove:** Remove Type-20 records which do not contain any references to other EBTS records (Type-20 record provides no use as there is no derivation from the Type-20 records to any other record.)

Data Attack, Data Hiding, and Data Disclosure: The data field of the Type-20 can be a complicated image or multimedia, it inherits all three risks from that separate data type.

7. **External Filtering Required:** Pass the data field contents to an external filter.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-20 records are located as defined in the CNT field in the Type-1 record. The Source Representation field (xx.997) exists in Type-10-17, and Type-19 records, outside of the Type-20 record data.

4.27 TYPE-21 Records

OVERVIEW:

Type-21 records are associated context record which contain information such as a context image or related audio/visual records. It does not provide any type of biometric data. An example from the ANSI/NIST standard is an image of the location or area where fingerprints were captured.

Type-21 records implement an Associated Content Format field. There are two subfields in this field. The first is a mandatory file type, with the suffix or extension. It may also contain "ANALOG" for an analog file or "OTHER" for digital data. The second subfield is a 1000 byte free text field that contains decoding instructions.

A field exists in Type-21 records called the External File Reference field. It is 200 bytes in length and contains the Uniform Resource Identifier (URI) of the data if the record itself does not provide the data.

The data portion of the field may contain any image or any other audio or visual recording of data that might be associated with the area where biometric data was captured.

RISKS AND RECOMMENDATIONS:

Data Hiding and Data Disclosure: The Associated Content Format field and the External File Reference field both allow for hidden data and possibly the location of accidently disclosed data.

1. **External Filtering Required:** Pass the Associated Content Format field to an external filter.

2. **External Filtering Required:** Pass the External File Reference field to an external filter.

Data Attack, Data Hiding, and Data Disclosure: The data portion of a Type-21 record may be any image or audio/visual recording. The inclusion of a complex data type means that it introduces all three types of risk.

3. **External Filtering Required:** Pass the data field to an external filter.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-21 records are located as defined in the CNT field in the Type-1 record.

4.28 TYPE-22 Records

OVERVIEW:

Type-22 records are included in the NIST Special Publication 500-290, Rev1, which is an updated version of the ANSI/NIST 2011 ITL standard. This version was released in 2013. It is intended for forensic imagery and associated data that is not included in a Type-10 record. Examples listed are 2D and 3D radiographs, Computerized Tomography (CT) scans, Magnetic Resonance Imaging (MRI) scans, and data for 3D printing.

RISKS AND RECOMMENDATIONS:

Data Attack, and Data Hiding: Since Type-22 records might not be implemented by some systems, it could possibly include hidden or malicious data. EBTS parsers may skip over records that are not implemented.

1. **Remove:** Remove all Type-22 records from the EBTS data, which will also require restructuring of the original EBTS data.

2. **External Filtering Required:** Extract and pass the data field of this record to an external filter.

PRODUCT: ANSI/NIST 2011 ITL - UPDATED 2013 VERSION

LOCATIONS:

Type-22 records are located as defined in the CNT field in the Type-1 record.

4.29 TYPE-98 Records

OVERVIEW:

Type-98 records were introduced in the ANSI/NIST ITL 2011 standard. They implement security information to ensure the integrity of the EBTS transaction and data. The record may include "binary data hashes, attributes for audit or identification purposes, and digital signatures" [2].

The ANSI/NIST standard defines an audit log field which may contain substantial information. Other fields of interest are located in User-defined fields, outside the scope of the ANSI/NIST standard. The Audit Log field is optional but also features two 200-byte fields (Event Reason and Agent) that are incorporated into the Audit Log. They are free-text fields that have no formal structure. There could any number of Audit Log fields repeated in the Type-98 record so these fields could add up to a large amount of data. These audit logs may be in an Extensible Markup Language (XML) format, in which case it should be treated as an embedded XML file that requires further inspection.

A digital signature is defined in User-defined fields since it is not officially part of the ANSI/NIST ITL 2011 standard. A Manifest field is defined in 98.200 and the Signature field is defined in 98.201 according to the ANSI/NIST ITL 2011 Type-98 Best Practice Implementation Guidance [6].

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: Since Type-98 records are not widely implemented, they could contain sensitive hidden data or malicious data.

1. **Validate:** Check that the digital signature is accurate with the rest of the EBTS data, if possible.
2. **Remove:** Remove the entire Type-98 record and reconstruct the EBTS data.
3. **External Filtering Required:** Pass the contents of the Audit Log field to an external filter.

PRODUCT: ANSI/NIST ITL 2011

LOCATIONS:

Type-98 records are located as defined in the CNT field in the Type-1 record.

4.30 TYPE-99 CBEFF Biometric Data Records

OVERVIEW:

Type-99 records contain biometric data that is not already covered by the ANSI/NIST standard (Type-1 through Type-17 records). This record is referred to as a Common Biometric Exchange Formats Framework (CBEFF) data record. The actual data portion of this record is binary and does not have a maximum size. There are several metadata fields in a Type-99 record (before the actual data) that are similar to other fields presented in previous sections; there are reserved fields and user-defined fields as well.

Other mandatory fields within the Type-99 record include a Source Agency/ORI (a free text field), a biometric creation date, a CBEFF header version, and a biometric type. The CBEFF header version (12 byte field) is the string “0101” and the biometric type field (16 byte field) is from a whitelist defined by Table 39 of the ANSI/NIST standard. The other two mandatory fields are the Biometric Data Block (BDB) Format Owner and BDB Format Type (both 12 bytes in length).

The BDB field is located at the end of the record within field 99.999. The BDB is the data portion of the CBEFF data format. The header of the CBEFF data is found within the tagged-field records in the Type-99 record. The structure of the BDB is dependent upon the value in the BDB Format Owner field. Each type of BDB must be registered with the International Biometrics Industry Association (IBIA).

RISKS AND RECOMMENDATIONS:

Data Attack and Data Hiding: The BDB field is a binary data block without a strict size limit. It must implement the matching structure defined in the IBIA. It could be used to inject binary executable, payload, or other hidden data.

1. **Remove:** Remove the entire Type-99 record and the biometric data block. This will require rebuilding the Type-1 record.

Data Hiding: Reserved and user-defined fields provide an area for someone to hide information, particularly if they are unimplemented by some viewers.

2. **Remove –** Remove all reserved and user-defined fields. Removing user-defined fields may disrupt functionality if defined by another standard.

3. **External Filtering Required** – Pass the contents of user-defined fields to an external filter.

PRODUCT: ALL EBTS PRODUCTS

LOCATIONS:

Type-99 records are located as defined in the CNT field in the Type-1 record.

4.31 EBTS Transactions

4.31.1 DoD EBTS Version 1.2

OVERVIEW:

EBTS data is sent via messages between two end point systems. Each message is considered a transaction by EBTS standards. Each transaction has an overall message data structure that defines the minimum and maximum number of record types. The transaction defines the structure of the EBTS data, something that can be validated and checked for correctness.

The DoD EBTS Version 1.2 standard defines the following transaction detail and requirements. The following data in Table 4-20 defines each EBTS Type of Transaction (ToT) from the Version 1.2 standard. The ToT field is in the 1.004 field in a Type-1 record.

Table 4-20. EBTS Transactions

EBTS ToT	Transaction Name	Description
CAR	Criminal 10-print Submission	Submission used for detainee or enemy prisoner of war.
FANC	Federal Applicant No Charge	Submission used as part of a background check for enlisting U.S. military, DoD civilians, and DoD contractors.
MAP	Miscellaneous Applicant	Submission used as part of a background check for local nationals and third country nationals who require access to U.S. military installations or other restricted areas.
DEK	Known Deceased	Submission used for deceased subject whose identity is known.
DEU	Unknown Deceased	Submission used for deceased subject whose identity is not known.
SRE	Submission Results – Electronic	Response containing an Ident/Non-Ident decision; will contain an electronic rap sheet if requested.
ERRT	10-print Transaction Error	Error response.
TPRS	10-print Rap Sheet Search	Performs a search only, non-retain, and can return an unconfirmed-identification (“yellow”) identification.
DPRS	DoD Flat Print Rap Sheet Search	Only used in special circumstances.
SRT	Search Result – 10-print	Response including a candidate list comprising names and DoD number of each candidate.
LFIS	Latent Fingerprint Image Search	Used for latent image submission and searches.
LFFS	Latent Fingerprint Feature Search	Used for latent feature submission and searches.
LRE	Latent Result	Latent response containing an Ident/Non-Ident decision.
ERRL	Latent Transaction Error	Error response.
IRQ	Fingerprint Image Request	Request for identification information (flat prints, mug shots, demographic/biographic information).
IRR	Fingerprint Image Request Response	Response containing requested identification information (flat prints, mug shots, demographic/biographic information).
ISR	Image Response Summary	Response indicating that the prints were not on file with DoD ABIS.
ERRI	Image transaction error.	Error response.
CPR	Subject Photo Request	Request for mug shot photos on file with DoD ABIS.
PRR	Photo Response	Response containing requested identification mug shot photos.

VER	Verification Electronic Submission	Used for 1:1 verification based on an identifier.
VRSP	Verification Response – Electronic	A verification response that contains Ident/Non-Ident information.
EVER	Verification Error Response	Error response.

For every possible transaction in EBTS from the table above, the DoD standard lists the structural requirements. The following data in Table 4-21 below is from the Version 1.2 standard and identifies this structure per Type of Transaction. This is important because it can be used to validate every EBTS transaction.

Table 4-21. EBTS Structure per Transaction (EBTS Version 1.2)

ToT	Type-1	Type-2	Type-4	Type-7	Type-9	Type-10	Type-13	Type-14	Type-16
CAR	1	1	0-14*	-	-	0-5	-	0-14***	0-6**
DEK	1	1	0-14*	-	-	0-5	-	0-14***	0-6**
DEU	1	1	0-14*	-	-	0-5	-	0-14***	0-6**
MAP	1	1	0-14*	-	-	0-5	-	0-14***	0-6**
FANC	1	1	0-14*	-	-	0-5	-	0-14***	0-6**
VER	1	1	1-14	-	-	-	-	-	-
LFFS	1	1-2	0-10	0-10	1-10****	-	0-10	0-10	-
LFIS	1	1-2	0-10	0-10	-	-	0-10	0-10	-
TPRS	1	1	2-10	-	0-10	-	-	-	-
DPRS	1	1	0-14	0	0-14	0-5	-	0-14***	0-6**
SRE	1	1	-	-	-	-	-	-	-
VRSP	1	1	-	-	-	0-1	-	-	-
EVER	1	1	-	-	-	-	-	-	-
SRT	1	1	0-14	-	-	-	-	-	-
ERRT	1	1	-	-	-	-	-	-	-
LRE	1	1	-	-	-	-	-	-	-
ERRL	1	1	1-n	-	-	-	-	-	-
IRQ	1	1	-	-	-	-	-	-	-
CPR	1	1	-	-	-	-	-	-	-
PRR	1	1	-	-	-	1-5	-	-	-
IRR	1	1	1-4	-	-	0-1	-	-	-
ISR	1	1	-	-	-	-	-	-	-
ERRI	1	1	-	-	-	-	-	-	-

* - Less than 14 images are acceptable in cases of amputation, deformity, or bandage. This should be indicated in the Type-2 record.

** - This field shall contain IRIS images.

*** - This field shall contain 1,000-ppi fingerprint images if available.

****- For each finger submitted as minutiae (Type-9), the corresponding image must be submitted as a Type-4, 7, 13, or 14 record.

RISKS AND RECOMMENDATIONS:

Data Hiding and Data Attack: An EBTS transaction that does not adhere to the structure defined in this construct could be an attempt at data hiding. If unnecessary or unneeded records are implemented, then they may be ignored on the receiving end (some applications may leverage optional records). Ignored binary records could be an attempt at a data attack by hiding malicious shellcode or executable content.

1. Validate: Check for each EBTS transaction, based upon the ToT field, that the remainder of the EBTS data adheres to its structural requirements.

2. Remove: Remove any record types that do not adhere to the structure per each transaction type. This will require rebuilding the Type-1 record (File Content field – CNT).

3. Reject: Reject any EBTS message or transaction that does not adhere to the structure per each transaction.

PRODUCT: DOD EBTS 1.2

LOCATIONS:

The ToT field is in a Type-1 record (1.004), but this construct applies to the entire block ofdata.

4.31.2 DoD EBTS Version 2.0

OVERVIEW:

The more recent version of the DoD EBTS standard identifies a slightly different requirement for EBTS transactions. The following data in Table 4-22 describes each valid EBTS transaction and its record composition. A smaller subset of transactions are present in EBTS Version 2.0.

Table 4-22. EBTS Structure per Transaction (EBTS Version 2.0)

ToT	Type-1	Type-2	Type-10	Type-13	Type-14	Type-15	Type-17
CAR	1	1	0-n	-	0-14[*]	0-8	0-6[**]
MAP	1	1	0-n	-	0-14[*]	0-8	0-6[**]
LFS	1	1	-	0-10	0-10	-	-
SRE	1	1	-	-	-	-	-
LSR	1	1	-	-	0-10	-	-
ERRT	1	1	-	-	-	-	-
ERRL	1	1	-	-	-	-	-

*- If less than 14 images are present, a reason shall be provided in a Type-2 record.

**- Record type contains IRIS images.

RISKS AND RECOMMENDATIONS:

Data Hiding and Data Attack: An EBTS transaction that does not adhere to the structure defined in this construct could be an attempt at data hiding. If unnecessary or unneeded records are implemented then they may be ignored on the receiving end (some applications may leverage optional records). Ignored binary records could be an attempt for data attack by hiding malicious shellcode or executable content.

1. Validate: Check for each EBTS transaction, based upon the ToT field, that the remainder of the EBTS data adheres to its structural requirements.

2. Remove: Remove any record types that do not adhere to the structure per each transaction type. This will require rebuilding the Type-1 record (File Content field – CNT).

3. Reject: Reject any EBTS message or transaction that does not adhere to the structure per each transaction.

PRODUCT: DOD EBTS 2.0

LOCATIONS:

The ToT field is in a Type-1 record (1.004), but this construct applies to the entire block ofdata.

5. SUMMARY OF RISKS

Table 5-1 Summary of Risks

Construct Section	Data Attack	Data Hiding	Data Disclosure
4.1 Record/Field Numbers and Data Types	X	X	
4.2 Information Separators	X	X	
4.3 Mandatory and Optional Fields	X	X	
4.4 Record Field Size	X		
4.5. Field Occurrence		X	
4.6 Record Order	X	X	
4.7 Trailing Data	X	X	
4.8 Record Lengths	X	X	
4.9 Generic Image Records	X	X	X
4.10 Type-1 Record	X	X	
4.10.1 Type-1 Record – File Content Field	X	X	
4.10.2 Type-1 Record – Character Encoding	X	X	
4.11 Type-2 Record	X	X	
4.11.1.1 FBI EBTS 7.0-10.0 Domain Type-2 Fields		X	X
4.11.1.2 DoD EBTS 1.2 Domain Type-2 Fields		X	X
4.11.1.3 DoD EBTS 2.0 Type-2 Fields		X	X
4.12.1 Image Designation Characters		X	
4.12.2 Agency and Source Information			X
4.12.3 Record Hash Field	X	X	
4.12.4 Geographic Sample Acquisition Location	X	X	X
4.12.5 Annotation Information			X
4.12.6 Comments		X	X
4.12.7 User Defined Fields	X	X	
4.12.8 Reserved Fields	X	X	
4.13 Type-3-6 Records	X	X	
4.14 Type-7 Record	X	X	X
4.15 Type-8 Record	X	X	X
4.16 Type-9 Record		X	
4.16.1 Type-9 Records – General Minutiae Data	X	X	
4.16.2 Type-9 Records – DoD EBTS v2.0 Minutiae Data	X	X	
4.17 Type-10 Record	X	X	X
4.18 Type-11 and 12 Records		X	

4.19 Type-13 Variable-Resolution Latent Image Records	X	X	X
4.20 Type-14 Variable-Resolution Fingerprint Image Records	X	X	X
4.21 Type-15 Variable-Resolution Palmprint Image Records	X	X	X
4.22 Type-16 User-Defined Testing Image Records	X	X	X
4.23 Type-17 Iris Image Records	X	X	X
4.24 Type-18 Records		X	X
4.25 Type-19 Records	X	X	X
4.26 Type-20 Records	X	X	X
4.27 Type-21 Records	X	X	X
4.28 Type-22 Records	X	X	
4.29 Type-98 Records	X	X	
4.30 Type-99 CBEFF Biometric Data Records	X	X	
4.31.1 DoD EBTS Version 1.2	X	X	
4.31.2 DoD EBTS Version 2.0	X	X	
Sum of each risk category:	**35**	**43**	**20**
Total Risks:			**98**

6. ACRONYMS

Table 6-1. Acronyms

Acronym	Denotation
ABIS	Automated Biometric Identification System
ANSI	American National Standards Institute
APS	Application Profile Specification
ASCII	American Standard Code for Information Interchange
BDB	Biometric Data Block
CBEFF	Common Biometric Exchange Formats Framework
CNT	File (or Transaction) Content Field
CT	Computerized Tomography
DNA	Deoxyribonucleic Acid
DoD	Department of Defense
EBTS	Electronic Biometric Transmission Specification
EFTS	Electronic Fingerprint Transmission Specification
EOI	End of Image
ETX	End of Text
FBI	Federal Bureau of Investigation
FS	File Separator
GS	Group Separator
HLL	Horizontal Line Length
IAFIS	Integrated Automated Fingerprint Identification System
IDC	Image Designation Character
JPEG	Joint Photography Experts Group
MRI	Magnetic Resonance Image
NIST	National Institute of Standards and Technology
PNG	Portable Network Graphics
RS	Record Separator
SMT	Scar, Mark, and Tattoo
SOI	Start of Image
SRT	Signature Representation Type
STX	Start of Text

Acronym	Denotation
ToT	Type of Transaction
URI	Uniform Resource Identifier
US	Unit Separator
UTF	Universal Coded Character Set Transformation Format
VLL	Vertical Line Length
WSQ	Wavelet Scalar Quantization
XML	Extensible Markup Language

7. REFERENCED DOCUMENTS

[1] NIST Special Publication 500-271. Information Technology: American National Standard for Information Systems - Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information – Part 1. Available online at: http://www.nist.gov/itl/iad/ig/ansi_standard.cfm (ANSI/NIST ITL1-2007-Approved-Std-20070427.pdf). April 27, 2007.

[2] NIST Special Publication 500-290. Information Technology: American National Standard for Information Systems Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information. Available online at: http://www.nist.gov/itl/iad/ig/ansi_standard.cfm (ANSI/NIST 2011 ITL1.pdf). November, 2011.

[3] NIST Special Publication 500-290 Rev1 (2013). Information Technology: American National Standard for Information Systems – Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information. ANSI/NIST ITL1-2011 – Updated 2013 version. Available online at: http://www.nist.gov/itl/iad/ig/ansi_standard.cfm. December, 2013

[4] Department of Defense: Electronic Biometric Transmission Specification. Version 1.2. DIN: DOD_BTF_TS_EBTS_Nov06_01.02.00. November 8, 2006.

[5] Department of Defense: Electronic Biometric Transmission Specification. Version 2.0. DIN: DOD_BTF_TS_EBTS_Mar09_02.00.00. March 27, 2009.

[6] ANSI/NIST ITL 2011 Type-98 Best Practice Implementation Guidance. For the Assurance of Biometric Data Integrity, Authenticity, and Auditable Chain of Custod. Version 1.3. Available at: http://biometrics.nist.gov/cs_links/standard/Type_98_Best_Practice_Guidance_v1.3.pdf. June 24, 2011.

[7] NIST Resources: Glossary. Online reference available at: http://www.nist.gov/forensics/EFSTrainingTool/ResourcesTab/Glossary.html