Network Working Group
Internet Draft
Obsoletes: 5102
Category: Standards Track
Expires: July 11, 2013

B. Claise, Ed.
Cisco Systems, Inc.
B. Trammell, Ed.
ETH Zurich

January 7, 2013

Information Model for IP Flow Information eXport (IPFIX)
draft-ietf-ipfix-information-model-rfc5102bis-09.txt

Abstract

This document provides an overview of the information model for the IP Flow Information eXport (IPFIX) protocol, as defined in the IANA IPFIX Information Element Registry. It is used by the IPFIX Protocol for encoding measured traffic information and information related to the traffic Observation Point, the traffic Metering Process, and the Exporting Process. Although developed for the IPFIX Protocol, the model is defined in an open way that easily allows using it in other protocols, interfaces, and applications. This document obsoletes RFC 5102.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 23, 2012.

Table of Contents

1. Introduction

The IP Flow Information eXport (IPFIX) protocol serves for transmitting information related to network traffic measurement. The protocol specification in [RFC5101bis] defines how Information Elements are transmitted. For Information Elements, it specifies the encoding of a set of basic data types. However, the list of Information Elements that can be transmitted by the protocol, such as Flow attributes (source IP address, number of packets, etc.) and information about the Metering and Exporting Process (packet Observation Point, sampling rate, Flow timeout interval, etc.), is not specified in [RFC5101bis].

The canonical reference for IPFIX Information Elements is the IANA IPFIX Information Element registry [IPFIX-IANA]; the initial values for this registry were provided by [RFC5102].

This document complements the IPFIX protocol specification [RFC5101bis] by providing an overview of the IPFIX information model and specifying data types for it. IPFIX-specific terminology used in this document is defined in Section 2 of [RFC5101bis]. As in [RFC5101bis], these IPFIX-specific terms have the first letter of a word capitalized when used in this document.

The use of the term 'information model' is not fully in line with the definition of this term in [RFC3444], as the IPFIX information model does not specify relationships between Information Elements. Nor does the IPFIX informaiton model specify a concrete encoding of Information Elements; for an encoding suitable for use with the IPFIX protocol, see [RFC5101bis]. Besides the encoding used by the IPFIX protocol, other encodings of IPFIX Information Elements can be

applied, for example, XML-based encodings.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.1. Changes since RFC 5102

This document obsoletes the Proposed Standard revision of the IPFIX Protocol Specification [RFC5102]. The following changes have been made to this document with respect to the previous document:

- All outstanding technical and editorial errata filed on the [RFC5102] as of publication time have been corrected.
- All references into [RFC5101] have been updated to [RFC5101bis], reflecting changes in that document as necessary.
- Information element definitions have been removed, as the reference for these is now [IPFIX-IANA]; a historical note on categorizations of information elements as defined in [RFC5102] has been retained in section 5.
- The process for modifying [IPFIX-IANA] has been improved, and is now described in [IPFIX-IE-DOCTORS]; Section 6 has been updated accordingly, and a new section 7.3 gives IANA considerations for this process.
- Definitions of timestamp data types have been clarified.
- Appendices A and B have been removed

1.2. IPFIX Documents Overview

The IPFIX protocol provides network administrators with access to network flow information. The architecture for the export of measured flow information out of an IPFIX Exporting Process to a Collecting Process is defined in [RFC5470], per the requirements defined in [RFC3917]. The IPFIX Protocol Specification [RFC5101bis] defines how IPFIX data records and templates are carried via a number of transport protocols from IPFIX Exporting Processes to IPFIX Collecting Processes.

Four IPFIX optimizations/extensions are currently specified: a bandwidth saving method for the IPFIX protocol in [RFC5473], an efficient method for exporting bidirectional flows in [RFC5103], a method for the definition and export of complex data structures in [RFC6313], and the specification of the Protocol for IPFIX Mediations [IPFIX-MED-PROTO] based on the IPFIX Mediation Framework [RFC6183].

IPFIX has a formal description of IPFIX Information Elements, their name, type and additional semantic information, as specified in this document, with the export of the Information Element types specified

in [RFC5610].

[RFC6728] specifies a data model for configuring and monitoring IPFIX and PSAMP compliant devices using the NETCONF protocol, while [RFC6615] specifies a MIB module for monitoring.

In terms of development, [RFC5153] provides guidelines for the implementation and use of the IPFIX protocol, while [RFC5471] provides guidelines for testing.

Finally, [RFC5472] describes what type of applications can use the IPFIX protocol and how they can use the information provided. It furthermore shows how the IPFIX framework relates to other architectures and frameworks.

2. Properties of IPFIX Protocol Information Elements

2.1. Information Element Specification Template

Information in messages of the IPFIX protocol is modeled in terms of Information Elements of the IPFIX information model. The IPFIX Information Elements mentioned in Section 5 are specified in [IPFIX-IANA].

All Information Elements specified for the IPFIX protocol MUST have the following properties defined.

name - A unique and meaningful name for the Information Element.

elementId - A numeric identifier of the Information Element. If this identifier is used without an enterprise identifier (see [RFC5101bis] and enterpriseId below), then it is globally unique and the list of allowed values is administered by IANA. It is used for compact identification of an Information Element when encoding Templates in the protocol.

description - The semantics of this Information Element. Describes how this Information Element is derived from the Flow or other information available to the observer. Information Elements of dataType string or octetArray which have length constraints (fixed length, minimum and/or maximum length) MUST note these constraints in their description.

dataType - One of the types listed in Section 3.1 of this document or registered in the IANA IPFIX Information Element Data Types registry. The type space for attributes is constrained to facilitate implementation. The existing type space encompasses most primitive types used in modern programming languages, as well

as some derived types (such as ipv4Address) that are common to this domain.

status - The status of the specification of this Information Element. Allowed values are 'current' and 'deprecated'. All newly-defined Information Elements have 'current' status. The process for moving Information Elements to the 'deprecated' status is defined in Section 5.2 of [IPFIX-IE-DOCTORS].

Enterprise-specific Information Elements MUST have the following property defined:

enterpriseId - Enterprises may wish to define Information Elements without registering them with IANA, for example, for enterprise-internal purposes. For such Information Elements, the Information Element identifier described above is not sufficient when the Information Element is used outside the enterprise. If specifications of enterprise-specific Information Elements are made public and/or if enterprise-specific identifiers are used by the IPFIX protocol outside the enterprise, then the enterprise-specific identifier MUST be made globally unique by combining it with an enterprise identifier. Valid values for the enterpriseId are defined by IANA as Structure of Management Information (SMI) network management private enterprise numbers, defined at [PEN-IANA].

All Information Elements specified for the IPFIX protocol either in this document or by any future extension MAY have the following properties defined:

dataTypeSemantics - The integral types are qualified by additional semantic details. Valid values for the data type semantics are specified in Section 3.2 of this document or in a future extension of the information model.

units - If the Information Element is a measure of some kind, the units identify what the measure is.

range - Some Information Elements may only be able to take on a restricted set of values that can be expressed as a range (e.g., 0 through 511 inclusive). If this is the case, the valid inclusive range should be specified; values for this Information Element outside the range are invalid and MUST NOT be exported.

reference - Identifies additional specifications that more precisely define this item or provide additional context for its use.

The following two Information Element properties are defined to allow the management of an Information Element registry with Information Element definitions that may be updated over time, per the process defined in Section 5.2 of [IPFIX-IE-DOCTORS].

revision - The revision number of an Information Element, starting at 0 for Information Elements at time of definition, and incremented by one for each revision.

date - The date of the entry of this revision of the Information Element into the registry.

A template for specifying Information Elements in Internet-Drafts is given in Section 9.1 of [IPFIX-IE-DOCTORS], and an XML Schema for specifying Information Elements in the IANA IPFIX registry [IPFIX-IANA] at [IPFIX-XML-SCHEMA].

2.2. Scope of Information Elements

By default, most Information Elements have a scope specified in their definitions. Within Data Records defined by Option Templates, the IPFIX protocol allows further limiting of the Information Element scope. The new scope is specified by one or more scope fields and defined as the combination of all specified scope values; see Section 3.4.2.1 on IPFIX scopes in [RFC5101bis].

2.3. Naming Conventions for Information Elements

The following naming conventions were used for naming Information Elements in this document. It is recommended that extensions of the model use the same conventions.

- Names of Information Elements SHOULD be descriptive.

- Names of Information Elements MUST be unique within the IANA IPFIX registry [IPFIX-IANA]. Enterprise-specific Information Elements SHOULD be prefixed with a vendor name.

- Names of Information Elements MUST start with non-capitalized letters.

- Composed names MUST use capital letters for the first letter of each component (except for the first one). All other letters are non-capitalized, even for acronyms. Exceptions are made for acronyms containing non-capitalized letters, such as 'IPv4' and 'IPv6'. Examples are sourceMacAddress and destinationIPv4Address.

- Middleboxes [RFC3234] may change Flow properties, such as the

Differentiated Service Code Point (DSCP) value or the source IP address. If an IPFIX Observation Point is located in the path of a Flow before one or more middleboxes that potentially modify packets of the Flow, then it may be desirable to also report Flow properties after the modification performed by the middleboxes. An example is an Observation Point before a packet marker changing a packet's IPv4 Type of Service (TOS) field that is encoded in Information Element ipClassOfService. Then the value observed and reported by Information Element ipClassOfService is valid at the Observation Point, but not after the packet passed the packet marker. For reporting the change value of the TOS field, the IPFIX information model uses Information Elements that have a name prefix "post", for example, "postIpClassOfService". Information Elements with prefix "post" report on Flow properties that are not necessarily observed at the Observation Point, but which are obtained within the Flow's Observation Domain by other means considered to be sufficiently reliable, for example, by analyzing the packet marker's marking tables.

3. Type Space

This section describes the abstract data types that can be used for the specification of IPFIX Information Elements in Section 4. Section 3.1 describes the set of abstract data types.

Abstract data types unsigned8, unsigned16, unsigned32, unsigned64, signed8, signed16, signed32, and signed64 are integral data types. As described in Section 3.2, their data type semantics can be further specified, for example, by 'totalCounter', 'deltaCounter', 'identifier', or 'flags'.

3.1. Abstract Data Types

This section describes the set of valid abstract data types of the IPFIX information model. Note that further abstract data types may be specified by future updates to this document. Changes to the associated IPFIX Information Element Data Types subregistry [IPFIX-IANA] specified in [RFC5610] require a Standards Action [RFC5226].

3.1.1. unsigned8

The type "unsigned8" represents a non-negative integer value in the range of 0 to 255.

3.1.2. unsigned16

The type "unsigned16" represents a non-negative integer value in the range of 0 to 65535.

3.1.3. unsigned32

The type "unsigned32" represents a non-negative integer value in the range of 0 to 4294967295.

3.1.4. unsigned64

The type "unsigned64" represents a non-negative integer value in the range of 0 to 18446744073709551615.

3.1.5. signed8

The type "signed8" represents an integer value in the range of -128 to 127.

3.1.6. signed16

The type "signed16" represents an integer value in the range of -32768 to 32767.

3.1.7. signed32

The type "signed32" represents an integer value in the range of -2147483648 to 2147483647.

3.1.8. signed64

The type "signed64" represents an integer value in the range of -9223372036854775808 to 9223372036854775807.

3.1.9. float32

The type "float32" corresponds to an IEEE single-precision 32-bit floating point type as defined in [IEEE.754.1985].

3.1.10. float64

The type "float64" corresponds to an IEEE double-precision 64-bit floating point type as defined in [IEEE.754.1985].

3.1.11. boolean

The type "boolean" represents a binary value. The only allowed values are "true" and "false".

3.1.12. macAddress

The type "macAddress" represents a MAC-48 address as in

[IEEE.802-3.2002].

3.1.13. octetArray

The type "octetArray" represents a finite-length string of octets.

3.1.14. string

The type "string" represents a finite-length string of valid characters from the Unicode character encoding set [ISO.10646-1.1993]. Unicode allows for ASCII [ISO.646.1991] and many other international character sets to be used.

3.1.15. dateTimeSeconds

The data type dateTimeSeconds represents the number of seconds since the UNIX epoch, 1 January 1970 at 00:00 UTC, as defined in [POSIX.1].

3.1.16. dateTimeMilliseconds

The data type dateTimeMilliseconds represents the number of milliseconds since the UNIX epoch, 1 January 1970 at 00:00 UTC, as defined in [POSIX.1].

3.1.17. dateTimeMicroseconds

The type "dateTimeMicroseconds" represents a time value with microsecond precision according to the NTP Timestamp format as defined in section 6 of [RFC5905].

3.1.18. dateTimeNanoseconds

The type "dateTimeNanoseconds" represents a time value with nanosecond precision according to the NTP Timestamp format as defined in section 6 of [RFC5905].

3.1.19. ipv4Address

The type "ipv4Address" represents an IPv4 address.

3.1.20. ipv6Address

The type "ipv6Address" represents an IPv6 address.

3.1.21. basicList

The type "basicList" supports structured data export as described in [RFC6313]; see section 4.5.1 of that document for encoding details.

3.1.22. subTemplateList

The type "subTemplateList" supports structured data export as described in [RFC6313]; see section 4.5.2 of that document for encoding details.

3.1.23. subTemplateMultiList

The type "subTemplateMultiList" supports structured data export as described in [RFC6313]; see section 4.5.3 of that document for encoding details.

3.2. Data Type Semantics

This section describes the set of valid data type semantics of the IPFIX information model. A sub-registry of data type semantics [IPFIX-IANA] is established in [RFC5610]; the restrictions on the use of semantics below are compatible with those specified in section 3.10 of that document. These semantics apply only to numeric types, as noted in the description of each semantic below.

Further data type semantics may be specified by future updates to this document. Changes to the associated IPFIX Information Element Semantics sub-registry [IPFIX-IANA] require a Standards Action [RFC5226].

3.2.1. quantity

A numeric (integral or floating point) value representing a measured value pertaining to the record. This is distinguished from counters that represent an ongoing measured value whose "odometer" reading is captured as part of a given record. This is the default semantic type of all numeric data types.

3.2.2. totalCounter

An numeric value reporting the value of a counter. Counters are unsigned and wrap back to zero after reaching the limit of the type. For example, an unsigned64 with counter semantics will continue to increment until reaching the value of 2**64 - 1. At this point, the next increment will wrap its value to zero and continue counting from zero. The semantics of a total counter is similar to the semantics of counters used in SNMP, such as Counter32 defined in [RFC2578]. The only difference between total counters and counters used in SNMP is that the total counters have an initial value of 0. A total counter counts independently of the export of its value.

3.2.3. deltaCounter

An numeric value reporting the value of a counter. Counters are unsigned and wrap back to zero after reaching the limit of the type. For example, an unsigned64 with counter semantics will continue to increment until reaching the value of 2**64 - 1. At this point, the next increment will wrap its value to zero and continue counting from zero. The semantics of a delta counter is similar to the semantics of counters used in SNMP, such as Counter32 defined in RFC 2578 [RFC2578]. The only difference between delta counters and counters used in SNMP is that the delta counters have an initial value of 0. A delta counter is reset to 0 each time it is exported and/or expires without export.

3.2.4. identifier

An integral value that serves as an identifier. Specifically, mathematical operations on two identifiers (aside from the equality operation) are meaningless. For example, Autonomous System ID 1 * Autonomous System ID 2 is meaningless. Identifiers MUST be one of the signed or unsigned data types.

3.2.5. flags

An integral value that represents a set of bit fields. Logical operations are appropriate on such values, but not other mathematical operations. Flags MUST always be of an unsigned data type.

4. Information Element Identifiers

All Information Elements defined in the IANA IPFIX Information Element registry [IPFIX-IANA] have their identifiers assigned by IANA.

The value of these identifiers is in the range of 1-32767. Within this range, Information Element identifier values in the sub-range of 1-127 are compatible with field types used by NetFlow version 9 [RFC3954] for historical reasons.

In general, IANA will add newly registered Information Elements to the registry, assigning the lowest available Information Element identifier in the range 128-32767.

Enterprise-specific Information Element identifiers have the same range of 1-32767, but they are coupled with an additional enterprise identifier. For enterprise-specific Information Elements, Information Element identifier 0 is also reserved. Enterprise-specific Information Element identifiers can be chosen by an enterprise

arbitrarily within the range of 1-32767. The same identifier may be assigned by other enterprises for different purposes; these Information Elements are distinct because the Information Element identifier is coupled with an enterprise identifier.

Enterprise identifiers are be registered as SMI network management private enterprise code numbers with IANA. The registry can be found at [PEN-IANA].

5. Information Elements

[IPFIX-IANA] is now the normative reference for IPFIX Information Elements. At the time of publication of [RFC5102], this section defined the initial contents of that registry.

As a historical note, Information Elements were organized into categories in [RFC5102] according to their semantics and their applicability; these categories were not carried forward into [IPFIX-IANA] as an organizing principle. The categories (with example IEs) were:

1. Identifiers (e.g. ingressInterface)
2. Metering and Exporting Process Configuration (e.g. exporterIPv4Address)
3. Metering and Exporting Process Statistics (e.g. exportedOctetTotalCount)
4. IP Header Fields (e.g. sourceIPv4Address)
5. Transport Header Fields (e.g. sourceTransportPort)
6. Sub-IP Header Fields (e.g. sourceMacAddress)
7. Derived Packet Properties (e.g. bgpSourceAsNumber)
8. Min/Max Flow Properties (e.g. minimumIpTotalLength)
9. Flow Timestamps (e.g. flowStartTimeMilliseconds)
10. Per-Flow Counters (e.g. octetDeltaCount)
11. Miscellaneous Flow Properties (e.g. flowEndReason)
12. Padding (paddingOctets)

Information Elements derived from fields of packets or from packet treatment can typically serve as Flow Keys used for mapping packets to Flows. These Information Elements were placed in categories 4-7 in the original categorization.

Information Elements not serving as Flow Keys may have different values for each packet in a Flow. For Information Elements with values derived from packets fields or packet treatment, and for which the value may change from packet to packet within a single Flow, the exported value of an Information Element is by default determined by the first packet observed for the corresponding Flow; the description of the Information Element may however explicitly specify different

semantics. This simple rule allows writing all Information Elements related to header fields once when the first packet of the Flow is observed. For further observed packets of the same Flow, only Flow properties that depend on more than one packet need to be updated; these Information Elements were placed in categories 8-11 in the original categorization.

Information Elements with a name having the "post" prefix (e.g. postIpClassOfService), do not necessarily report properties that were actually observed at the Observation Point, but may be retrieved by other means within the Observation Domain. These Information Elements can be used if there are middlebox functions within the Observation Domain changing Flow properties after packets passed the Observation Point; they may also be reported directly by the Observation Point if the Observation Point is situated such as to observe packets on both sides of the middlebox.

6. Extending the Information Model

A key requirement for IPFIX is to allow for extension of the Information Model via the IANA IPFIX registry [IPFIX-IANA]. New Information Element definitions can be added to this registry subject to an Expert Review [RFC5226], with additional process considerations decribed in [IPFIX-IE-DOCTORS]; that document also provides guidelines for authors and reviewers of new Information Element definitions.

For new Information Elements, the type space defined in Section 3 can be used. If required, new abstract data types can be added to the data type subregistry [IPFIX-IANA] defined in [RFC5610]. New abstract data types and semantics are subject to Standards Action [RFC5226], and MUST be defined in IETF Standards Track documents updating this document.

Enterprises may wish to define Information Elements without registering them with IANA. IPFIX explicitly supports enterprise-specific Information Elements. Enterprise-specific Information Elements are described in Sections 2.1 and 4; guidelines for using them appear in [IPFIX-IE-DOCTORS].

7. IANA Considerations

7.1. IPFIX Information Elements

This document refers to Information Elements, for which the Internet Assigned Numbers Authority (IANA) has created the IPFIX Information Element Registry [IPFIX-IANA]. The columns of this registry must at minimum be able to store the information defined in the template in Section 2.1; it may contain other information as necessary for the management of the registry.

The process for making additions or other changes to the IPFIX Information Element Registry is given in Section 7.4.

[NOTE to IANA: please update the Reference for the IPFIX Information Element Registry to refer to this document.]

[NOTE to IANA: on publication of this document, please create a new Revision column in the the IPFIX Information Element Registry, and set the Revision of all existing Information Elements to 0.]

[NOTE to IANA: on publication of this document, please create a new Date column in the the IPFIX Information Element Registry, and set the Date of all existing Information Elements to the publication date of this document.]

[NOTE to IANA: on publication of this document, please set the Name of all existing Reserved Information Elements with identifier 127 or less to "Assigned for NetFlow v9 compatibility", and the Reference to [RFC3954].]

7.2. MPLS Label Type Identifier

Information Element #46, named mplsTopLabelType, carries MPLS label types. Values for 5 different types have initially been defined. For ensuring extensibility of this information, IANA has created a new subregistry for MPLS label types and filled it with the initial list from the description Information Element #46, mplsTopLabelType.

New assignments for MPLS label types are administered by IANA through Expert Review [RFC5226], i.e., review by one of a group of experts designated by an IETF Area Director. The group of experts must double check the label type definitions with already defined label types for completeness, accuracy, and redundancy. The specification of new MPLS label types MUST be published using a well-established and persistent publication medium.

[NOTE to IANA: please update the Reference for the IPFIX MPLS Label Type

subregistry to refer to this document.]

7.3. XML Namespace and Schema

[IPFIX-XML-SCHEMA] defines an XML schema for IPFIX Information Element definitions. All Information Elements specified in [IPFIX-IANA] are defined by this schema. This schema may also be used for specifying further Information Elements in future extensions of the IPFIX information model in a machine-readable way.

[IPFIX-XML-SCHEMA] uses URNs to describe an XML namespace and an XML schema for IPFIX Information Elements conforming to a registry mechanism described in [RFC3688]. Two URI assignments have been made.

1. Registration for the IPFIX information model namespace

* URI: urn:ietf:params:xml:ns:ipfix-info
* Registrant Contact: IETF IPFIX Working Group <ipfix@ietf.org>, as designated by the IESG <iesg@ietf.org>.
* XML: None. Namespace URIs do not represent an XML.

2. Registration for the IPFIX information model schema

* URI: urn:ietf:params:xml:schema:ipfix-info
* Registrant Contact: IETF IPFIX Working Group <ipfix@ietf.org>, as designated by the IESG <iesg@ietf.org>.

Using a machine-readable syntax for the information model enables the creation of IPFIX-aware tools that can automatically adapt to extensions to the information model, by simply reading updated information model specifications.

The wide availability of XML-aware tools and libraries for client devices is a primary consideration for this choice. In particular, libraries for parsing XML documents are readily available. Also, mechanisms such as the Extensible Stylesheet Language (XSL) allow for transforming a source XML document into other documents. This document was authored in XML and transformed according to [RFC2629].

It should be noted that the use of XML in Exporters, Collectors, or other tools is not mandatory for the deployment of IPFIX. In particular, Exporting Processes do not produce or consume XML as part of their operation. It is expected that IPFIX Collectors MAY take advantage of the machine readability of the information model vs. hard coding their behavior or inventing proprietary means for accommodating extensions.

[NOTE to IANA: please update the Reference for the the IPFIX information model namespace and schema to refer to this document.]

7.4. Addition, Revision, and Deprecation

New assignments for IPFIX Information Elements are administered by IANA through Expert Review [RFC5226]. These experts are referred to as IE-DOCTORS experts, and are appointed by the IESG. The process they follow is defined in [IPFIX-IE-DOCTORS].

[IANA NOTE: please establish an ie-doctors mailing list for communicating with the IE-DOCTORS experts.]

Information Element identifiers in the range 1-127 are compatible with field types used by NetFlow version 9 [RFC3954] for historical reasons, and must not be assigned unless the Information Element is compatible with the NetFlow version 9 protocol, as determined by an IE-DOCTORS expert designated by the IESG as a Netflow version 9 expert.

Future assignments added to the IPFIX Information Element Registry which require subregistries for enumerated values (e.g. section 7.2, below) must have those subregistries added simultaneously with the new assignment; additions to these subregistries must be subject to Expert Review [RFC5226]. Unless specified at assignment time, the experts for the subregistry will be the same as for the Information Element registry as a whole.

When IANA receives a request to add, revise, or deprecate an Information Element in the IPFIX Information Elements Registry, it forwards the request to the IE-DOCTORS experts for review.

When IANA receives an approval for a request to add an Information Element definition from the IE-DOCTORS experts, it adds that Information Element to the registry. The approved request may include changes made by the requestor and/or reviewers as compared to the original request.

When IANA receives an approval for a request to revise an Information Element definition from the IE-DOCTORS experts, it changes that Information Element's definition in the registry, and updates the Revision and Date columns as appropriate. The approved request may include changes from the original request. If the original Information Element was added to the registry with IETF consensus (i.e., was defined by an RFC), the revision will require IETF consensus as well.

When IANA receives an approval for a request to deprecate an Information Element definition from the IE-DOCTORS experts, it changes that Information Element's definition in the registry, and updates the Revision and Date columns as appropriate. The approved request may include changes from the original request. If the original Information Element was added to the registry with IETF consensus (i.e., was defined by an RFC), the deprecation will require IETF consensus as well.

8. Security Considerations

The IPFIX information model itself does not directly introduce security issues. Rather, it defines a set of attributes that may for privacy or business issues be considered sensitive information.

For example, exporting values of header fields may make attacks possible for the receiver of this information, which would otherwise only be possible for direct observers of the reported Flows along the data path.

The underlying protocol used to exchange the information described here must therefore apply appropriate procedures to guarantee the integrity and confidentiality of the exported information. Such protocols are defined in separate documents, specifically the IPFIX protocol document [RFC5101bis].

This document does not specify any Information Element carrying keying material. If future extensions will do so, then appropriate precautions need to be taken for properly protecting such sensitive information.

9. Acknowledgements

The editors would like to thank the authors of [RFC5102], as this document is directly based upon this original RFC: Juergen Quittek, Stewart Bryant, Paul Aitken, and Jeff Meyer. Thanks to Paul Aitken for his detailed review.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5905] Mills, D., Delaware, U., Martin, J., Burbank, J. and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010

[RFC6313] Claise, B., Dhandapani, G., Aitken, P, and S. Yates, "Export of Structured Data in IP Flow Information Export (IPFIX)", RFC6313, July 2011.

[RFC5101bis] Claise, B., and B. Trammell, Editors, "Specification of the IP Flow Information eXport (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", draft-ietf-ipfix-protocol-rfc5101bis-04, Work in Progress, December 2012.

[IPFIX-IE-DOCTORS]
Trammell, B., and B. Claise, "Guidelines for Authors and Reviewers of IPFIX Information Elements", draft-ietf-ipfix-ie-doctors-07, Work in Progress, October 2012.

10.2. Informative References

[IEEE.802-3.2002]
Insitute of Electrical and Electronics Engineers, "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications", IEEE Standard 802.3, September 2002.

[IEEE.754.1985]
Institute of Electrical and Electronics Engineers, "Standard for Binary Floating-Point Arithmetic", IEEE Standard 754, August 1985.

[ISO.10646-1.1993]
International Organization for Standardization, "Information Technology - Universal Multiple-octet coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane", ISO Standard 10646-1, May 1993.

[ISO.646.1991]
International Organization for Standardization, "Information technology - ISO 7-bit coded character set for information interchange", ISO Standard 646, 1991.

[POSIX.1] IEEE 1003.1-2008 - IEEE Standard for Information Technology - Portable Operating System Interface, IEEE, 2008.

[RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between

Information Models and Data Models", RFC 3444, January 2003.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC3917] Quittek, J., Zseby, T., Claise, B., and S. Zander, "Requirements for IP Flow Information Export (IPFIX)", RFC 3917, October 2004.

[RFC3954] Claise, B., Ed., "Cisco Systems NetFlow Services Export Version 9", RFC 3954, October 2004.

[RFC5101] Claise, B., Bryant, S., Leinen, S., Dietz, T., and Trammell, B., "Specification of the IPFIX Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.

[RFC5102] Quittek, J., Bryant, S. Claise, B., Aitken, P., and Meyer, J., "Information Model for IP Flow Information Export", RFC 5102, January 2008.

[RFC5103] Trammell, B., and E. Boschi, "Bidirectional Flow Export Using IP Flow Information Export (IPFIX)", RFC 5103, January 2008.

[RFC5153] Boschi, E., Mark, L., Quittek J., and P. Aitken, "IP Flow Information Export (IPFIX) Implementation Guidelines", RFC5153, April 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", RFC5470, March 2009.

[RFC5471] Schmoll, C., Aitken, P., and B. Claise, "Guidelines for IP Flow Information Export (IPFIX) Testing", RFC5471, March 2009.

[RFC5472] Zseby, T., Boschi, E., Brownlee, N., and B. Claise, "IP Flow Information Export (IPFIX) Applicability", RFC5472, March 2009.

[RFC5473] Boschi, E., Mark, L., and B. Claise, "Reducing Redundancy in IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Reports", RFC5473, March 2009.

[RFC5610] Boschi, E., Trammell, B., Mark, L., and T. Zseby, "Exporting Type Information for IP Flow Information Export (IPFIX) Information Elements", July 2009.

[RFC6183] Kobayashi, A., Claise, B., Muenz, G, and K. Ishibashi, "IP Flow Information Export (IPFIX) Mediation: Framework", RFC6183, April 2011.

[RFC6615] Dietz, T., Kobayashi, A., Claise, B., and G. Muenz, "Definitions of Managed Objects for IP Flow Information Export", RFC6615, June 2012.

[RFC6728] Muenz, G., Claise, B., and P. Aitken, "Configuration Data Model for IPFIX and PSAMP", RFC 6728, October 2012.

[IPFIX-MED-PROTO] Claise, B., Kobayashi, A., and B. Trammell, "Operation of the IP Flow Information Export (IPFIX) Protocol on IPFIX Mediators", draft-ietf-ipfix-mediation-protocol-02, Work in Progress, July 2012.

[IPFIX-IANA] http://www.iana.org/assignments/ipfix/ipfix.xml

[PEN-IANA] http://www.iana.org/assignments/enterprise-numbers

[IPFIX-XML-SCHEMA] http://www.iana.org/assignments/xml-registry/schema/ipfix.xsd

Authors' Addresses

Benoit Claise
Cisco Systems, Inc.
De Kleetlaan 6a b1
1831 Diegem
Belgium

Phone: +32 2 704 5622
EMail: bclaise@cisco.com

Brian Trammell
Swiss Federal Institute of Technology Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Phone: +41 44 632 70 13
EMail: trammell@tik.ee.ethz.ch