Internet Engineering Task Force (IETF)
Request for Comments: 7013
BCP: 184
Category: Best Current Practice
ISSN: 2070-1721

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

September 2013

# Guidelines for Authors and Reviewers of IP Flow Information Export (IPFIX) Information Elements

## Abstract

This document provides guidelines for how to write definitions of new Information Elements for the IP Flow Information Export (IPFIX) protocol. It provides instructions on using the proper conventions for Information Elements to be registered in the IANA IPFIX Information Element registry, and provides guidelines for expert reviewers to evaluate new registrations.

## Status of This Memo

This memo documents an Internet Best Current Practice.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on BCPs is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7013.

Table of Contents

# 1. Introduction

This document provides guidelines for the definition of new IPFIX Information Elements beyond those currently in the IANA IPFIX Information Element Registry [IANA-IPFIX]. Given the self-describing nature of the data export format used by IPFIX, the definition of new Information Elements is often sufficient to allow the application of IPFIX to new network measurement and management use cases.

We intend this document to enable the application of IPFIX to new areas by experts in the IETF Working Group or Area Directorate, the IETF community, or organization external to the IETF, concerned with the technical details of the protocol or application to be measured or managed using IPFIX. This expansion occurs with the consultation of IPFIX experts informally called IE-DOCTORS. It provides guidelines both for those defining new Information Elements as well as the IE-DOCTORS reviewing them.

This document essentially codifies two meta-guidelines: (1) "define new Information Elements that look like existing Information Elements" and (2) "don’t define Information Elements unless you need to".

## 1.1. Intended Audience and Usage

This document is meant for two separate audiences. For those defining new Information Elements, it provides specifications and best practices to be used in deciding which Information Elements are necessary for a given existing or new application, instructions for writing the definitions for these Information Elements, and information on the supporting documentation required for the new application (up to and including the publication of one or more RFCs describing it). For the IPFIX experts appointed as IE-DOCTORS, and for IANA personnel changing the IANA IPFIX Information Element Registry [IANA-IPFIX], it defines a set of acceptance criteria against which these proposed Information Elements should be evaluated.

This document is not intended to guide the extension of the IPFIX protocol itself, e.g., through new export mechanisms, data types, or the like; these activities should be pursued through the publication of Standards Track RFCs within the IPFIX Working Group.

This document, together with [RFC7012], defines the procedures for management of the IANA IPFIX Information Element Registry [IANA-IPFIX]. The practices outlined in this document are intended

to guide experts when reviewing additions or changes to the Information Elements in the registry under Expert Review (as defined in [RFC5226]).

### 1.2. Overview of Relevant IPFIX Documents

[RFC7011] defines the IPFIX protocol, the IPFIX-specific terminology used by this document, and the data type encodings for each of the data types supported by IPFIX.

[RFC7012] defines the basis of the IPFIX Information Model, referring to [IANA-IPFIX] for the specific Information Element definitions. It states that new Information Elements may be added to the Information Model on the basis of Expert Review, delegates the appointment of experts to an IESG Area Director, and refers to this document for details on the extension process. This document is intended to further codify the best practices to be followed by these experts, in order to improve the efficiency of this process.

[RFC5103] defines a method for exporting bidirectional Flow information using IPFIX; this document should be followed when extending IPFIX to represent information about bidirectional network interactions in general. Additionally, new Information Elements should be annotated for their reversibility or lack thereof as per this document.

[RFC5610] defines a method for exporting information about Information Elements inline within IPFIX. In doing so, it explicitly defines a set of restrictions, implied in [RFC7011] and [RFC7012], on the use of data types and semantic; these restrictions must be observed in the definition of new Information Elements, as in Section 4.4.

## 2. Terminology

Capitalized terms used in this document that are defined in the Terminology section of [RFC7011] are to be interpreted as defined there.

An "application", as used in this document, refers to a candidate protocol, task, or domain to which IPFIX export, collection, and/or storage is applied. By this definition, the IPFIX applicability statement [RFC5472] defined the initial applications of IPFIX, and Packet Sampling (PSAMP) [RFC5476] was the first new IPFIX application after the publication of the IPFIX protocol itself.

"IANA IE registry", as used in this document, unless otherwise noted, refers to the IANA IPFIX Information Element Registry [IANA-IPFIX].

## 3. How to Apply IPFIX

Though originally specified for the export of IP Flow information, the message format, template mechanism, and data model specified by IPFIX led to it being applicable to a wide variety of network management situations. In addition to Flow information export, for which it was designed, and packet information export as specified by PSAMP [RFC5476], any application with the following characteristics is a good candidate for an IPFIX application:

- The application's data Flow is fundamentally unidirectional. IPFIX is a "push" protocol, supporting only the export of information from a sender (an Exporting Process) to a receiver (a Collecting Process). Request-response interactions are not supported by IPFIX.

- The application handles discrete event information, or information to be periodically reported. IPFIX is particularly well suited to representing events, which can be scoped in time.

- The application handles information about network entities. IPFIX's information model is network-oriented, so network management applications have many opportunities for information model reuse.

- The application requires a small number of arrangements of data structures relative to the number of records it handles. The template-driven self-description mechanism used by IPFIX excels at handling large volumes of identically structured data, compared to representations that define structure inline with data (such as XML).

Most applications meeting these criteria can be supported over IPFIX. Once it has been determined that IPFIX is a good fit, the next step is determining which Information Elements are necessary to represent the information required by the application. Especially for network-centric applications, the IANA IE registry may already contain all the necessary Information Elements (see Section 6.1 for guidelines on maximizing Information Element reuse). In this case, no work within the IETF is necessary: simply define Templates and start exporting.

It is expected, however, that most applications will be able to reuse some existing Information Elements, but may need to define some additional Information Elements to support all their requirements. In this case, see Section 4 for best practices to be followed in defining Information Elements.

Optionally, a Working Group or individual contributor may choose to write an Internet-Draft for publication as an RFC, detailing the new IPFIX application. Such an RFC should contain discussion of the new application, the Information Element definitions as in Section 4, as well as suggested Templates and examples of the use of those Templates within the new application as in Section 9.2. Section 10 defines a compact textual Information Element notation to be used in describing these suggested Templates and/or the use of IPFIX Structured Data [RFC6313] within the new application.

## 4. Defining New Information Elements

In many cases, a new application will require nothing more than a new Information Element or set of Information Elements to be exportable using IPFIX. An Information Element meeting the following criteria, as evaluated by the IE-DOCTORS, is eligible for inclusion in the IANA IE registry:

- The Information Element must be unique within the registry, and its description must represent a substantially different meaning from that of any existing Information Element. An existing Information Element that can be reused for a given purpose should be reused.

- The Information Element should contain as little internal structure as possible. Instead of representing complex information by overlaying internal structure on a simple data type such as octetArray, such information should be represented with multiple simple Information Elements to be exported in parallel or using IPFIX Structured Data [RFC6313], as in Section 4.5. The internal structure of a proposed IE may be evaluated by the IE-DOCTORS with an eye toward interoperability and/or backward compatibility with existing methods of exporting similar data on a case-by-case basis.

- Information Elements representing information about proprietary or nonstandard applications should not be registered in the IANA IE registry. These can be represented using enterprise-specific Information Elements as detailed in Section 3.2 of [RFC7011], instead.

The definition of new Information Elements requires a descriptive name, a specification of the data type from the IPFIX Data Type subregistry in the IANA IE registry (defined in [RFC7012] as itself extensible via Standards Action as per [RFC5226]), and a human-readable description written in English. This section provides

guidelines on each of these components of an Information Element definition, referring to existing documentation such as [RFC7012] as appropriate.

### 4.1. Information Element Naming

As the name of an Information Element is the first thing a potential implementor will use when determining whether it is suitable for a given application, it is important to be as precise and descriptive as possible. Names of Information Elements:

- must be chosen carefully to describe the use of the Information Element within the context in which it will be used.
- must be unique within the IANA IE registry.
- start with lowercase letters.
- use capital letters for the first letter of each component except for the first one (aka "camel case"). All other letters are lowercase, even for acronyms. Exceptions are made for acronyms containing a mixture of lowercase and capital letters, such as 'IPv4' and 'IPv6'. Examples are "sourceMacAddress" and "destinationIPv4Address".

In addition, new Information Elements pertaining to a specific protocol should name the protocol in the first word in order to ease searching by name (e.g., "sipMethod" for a SIP method, as would be used in a logging format for SIP based on IPFIX). Similarly, new Information Elements pertaining to a specific application should name the application in the first word.

### 4.2. Information Element Data Types

IPFIX provides a set of data types covering most primitives used in network measurement and management applications. The most appropriate data type should be chosen for the Information Element type, IPFIX informationElementDataTypes subregistry at [IANA-IPFIX]. This subregistry may be extended from time to time by a Standards Action [RFC5226], as defined in [RFC5610].

Information Elements representing an integral value with a natural width should be defined with the appropriate integral data type. This applies especially to values taken directly from fixed-width fields in a measured protocol. For example, tcpControlBits, the TCP flags byte, is an unsigned8, and tcpSequenceNumber is an unsigned32.

Information Elements representing counters or identifiers should be defined as signed64 or unsigned64, as appropriate, to maximize the range of values available; applications can use reduced-size encoding as defined in Section 6.2 of [RFC7011] in cases where fewer than 2^64 values are necessary.

Information Elements representing time values must be defined with appropriate precision. For example, an Information Element for a time measured at second-level precision should be defined as having a dateTimeSeconds data type, instead of dateTimeMilliseconds.

Information Elements of type string or octetArray that have length constraints (fixed length, minimum and/or maximum length) must note these constraints in their description.

The type of an Information Element must match the type of the data it represents. More specifically, information that could be represented as a string but that better matches one of the other data types (e.g., an integral type for a number or enumerated type, an address type for an address) must be represented by the best-matching type, even if the data was represented using a different type in the source. For example, an IPFIX application that exports Options Template Records mapping IP addresses to additional information about each host from an external database must use Information Elements of an address type to represent the addresses, even if the source database represented these as strings.

Strings and octetArrays must not be used to encode data that would be more properly represented using multiple Information Elements and/or IPFIX Structured Data [RFC6313]; see Section 4.5 for more.

This document does not cover the addition of new Data Types or Data Type Semantics to the IPFIX protocol. As such changes have important interoperability considerations and require implementation on both Collecting and Exporting Processes, they require a Standards Action as per [RFC5610]. However, note that the set of primitive types provided by IPFIX are applicable to almost any appropriate application, so extending the type system is generally not necessary.

### 4.3. Information Element Numbering

Each Information Element has a unique identifier in the IANA registry.

When adding newly registered Information Elements to the IANA IE registry, IANA should assign the lowest available Information Element identifier (the value column in [IANA-IPFIX]) in the range 128-32767.

Information Elements with identifiers in the range 1-127 are reserved for compatibility with corresponding fields in NetFlow version 9, as described in [RFC3954].

### 4.4. Ancillary Information Element Properties

Information Elements to which special semantics apply should refer to one of the values in the Information Element Semantics subregistry of the IANA IE registry, as described in Section 3.2 of [RFC7012], subject to the restrictions given in Section 3.10 of [RFC5610]; in other words, the semantics and the type must be consistent.

When defining Information Elements representing a dimensioned quantity or entity count, the units of that quantity should be defined in the units field. This field takes its values from the IANA Information Element Units subregistry of the IANA IE registry. If an Information Element expresses a quantity in units not yet in this subregistry, then the unit must be added to the Units subregistry at the same time the Information Element is added to the IANA IE registry. Note that the Units subregistry as defined in [RFC5610] is maintained on an Expert Review basis.

Additionally, when the range of values an Information Element can take is smaller than the range implied by its data type, the range should be defined within the Information Element’s entry in the IANA IE registry.

### 4.5. Internal Structure in Information Elements

The definition of Information Elements with an internal structure that is defined in the Description field is not recommended, except in the following cases:

1. The Information Element is a direct copy of a structured entity in a measured protocol (e.g., the tcpControlBits Information Element for the flags byte from the TCP header).

2. The Information Element represents a section of a packet of protocol entity, in raw form as captured from the wire (e.g., the mplsLabelStackSection Information Element for the MPLS label stack).

3. The Information Element represents a set of flags that are tightly semantically related, where representing the flags as separate one-byte booleans would be inefficient, and that should always appear together in a data record (e.g., the anonymizationFlags Information Element for specifying optional features of anonymization techniques).

4. The Information Element contains internal structure by reference to an external data type or specification containing internal structure (e.g., a MIME type or URL), for interoperability and backward-compatibility purposes.

Additional exceptions to the above list should be made through publication of an RFC.

In other cases, candidate Information Elements with internal structure should be decomposed into multiple primitive Information Elements to be used in parallel. For more complicated semantics, where the structure is not identical from Data Record to Data Record, or where there is semantic dependency between multiple decomposed primitive Information Elements, use the IPFIX Structured Data [RFC6313] extension instead.

As an example of Information Element decomposition, consider an application-level identifier called an "endpoint", which represents a {host, port, protocol} tuple. Instead of allocating an opaque, structured "source endpoint" Information Element, the source endpoint should be represented by three separate Information Elements: "source address", "source port", "transport protocol". In this example, the required Information Elements already exist in the IANA IE registry: sourceIPv4Address or sourceIPv6Address, sourceTransportPort, protocolIdentifier. Indeed, as well as being good practice, this normalization down to non-structured Information Elements also increases opportunities for reuse as in Section 6.1.

The decomposition of data with internal structure should avoid the definition of Information Elements that have a meaning too specific to be generally useful or that would result in a multitude of templates to handle different multiplicities. More information on multiplicities is given in the following section.

### 4.6. Information Element Multiplicity

Some Information Elements may represent information with a multiplicity other than one, i.e., items that may occur multiple times within the data to be represented in a single IPFIX record. In this case, there are several options, depending on the circumstances:

1. As specified in Section 8 of [RFC7011]: "if an Information Element is required more than once in a Template, the different occurrences of this Information Element should follow the logical order of their treatments by the Metering Process." In other words, in cases where the items have a natural order (e.g., the order in which they occur in the packet), and the multiplicity is the same for each record, the information can be modeled by

containing multiple instances of the Information Element representing a single item within the Template Record describing the Data Records.

2. In cases where the items have a variable multiplicity, a basicList of the Information Element representing a single item can be used as in the IPFIX Structured Data [RFC6313] extension.

3. If the multiple-item structure is taken directly from bytes observed on the wire by the Metering Process or otherwise taken from the application being measured (e.g., a TCP options stack), the multiple-item structure can be exported as a variable-length octetArray Information Element holding the raw content.

Specifically, a new Information Element should not encode any multiplicity or ordinality information into the definition of the Information Element itself.

### 4.7. Enumerated Values and Subregistries

When defining an Information Element that takes an enumerated value from a set of values that may change in the future, this enumeration must be defined by an IANA IE registry or subregistry. For situations where an existing registry defines the enumeration (e.g., the IANA Protocol Numbers registry for the protocolIdentifier Information Element), that registry must be used. Otherwise, a new subregistry of the IANA IPFIX registry must be defined for the enumerated value, to be modified subject to Expert Review [RFC5226].

### 4.8. Reversibility as per RFC 5103

[RFC5103] defines a method for exporting bidirectional Flows using a special Private Enterprise Number to define reverse-direction variants of IANA Information Elements, and a set of criteria for determining whether an Information Element may be reversed using this method. Since almost all Information Elements are reversible, [RFC5103] enumerates those Information Elements that were defined at the time of its publication that are NOT reversible.

New non-reversible Information Elements must contain a note in the description stating that they are not reversible.

### 4.9. Avoiding Bad Ideas in Information Element Design

In general, the existence of a similarly defined Information Element in the IANA IE registry sets a precedent that may be followed to determine whether a given proposed Information Element "fits" within the registry. Indeed, the rules specified by this document could be

interpreted to mean "make new Information Elements that look like existing Information Elements". However, for reasons of history, there are several Information Elements within the IANA IE registry that do not follow best practices in Information Element design.

These Information Elements are not necessarily so flawed so as to require deprecation, but they should be explicitly ignored when looking for guidance as to whether a new Information Element should be added. Here we provide a set of representative examples taken from the IANA IE registry; in general, entries in the IANA IE registry that do not follow the guidelines in this document should not be used as examples for new Information Element definitions.

Before registering a new Information Element, it must be determined that it would be sufficiently unique within the IANA IE registry. This evaluation has not always been done in the past, and the existence of the Information Elements defined without this evaluation should not be taken as an example that such Information Element definition practices should be followed in the future. Specific examples of such Information Elements include initiatorOctets and responderOctets (which duplicate octetDeltaCount and its reverse per [RFC5103]) and initiatorPackets and responderPackets (the same, for packetDeltaCount).

As mentioned in Section 4.2, the type of an Information Element should match the type of data the Information Element represents. An example of how not to do this is presented by the p2pTechnology, tunnelTechnology, and encryptedTechnology Information Elements: these represent a three-state enumeration using a String. The example set by these Information Elements should not be followed in the definition of new Information Elements.

As mentioned in Section 4.6, an Information Element definition should not include any ordinality or multiplicity information. The only example of this within the IANA IE registry the following list of assigned IPFIX Information Elements: mplsTopLabelStackSection, mplsLabelStackSection2, mplsLabelStackSection3, mplsLabelStackSection4, mplsLabelStackSection5, mplsLabelStackSection6 mplsLabelStackSection7, mplsLabelStackSection8, mplsLabelStackSection9, and mplsLabelStackSection10. The only distinction between those almost-identical Information Elements is the position within the MPLS stack. This Information Element design pattern met an early requirement of the definition of IPFIX that was not carried forward into the final specification -- namely, that no semantic dependency was allowed between Information Elements in the same Record -- and as such should not be followed in the definition of new Information Elements. In this case, since the size of the MPLS stack will vary from Flow to

Flow, it should be exported using IPFIX Structured Data [RFC6313] where supported, as a basicList of MPLS label entries, or as a raw MPLS label stack using the variable-length mplsLabelStackSection Information Element.

## 5. The Information Element Life Cycle

Once an Information Element or set of Information Elements has been identified for a given application, Information Element specifications in accordance with Section 4 are submitted to IANA to follow the process for review by the IE-DOCTORS, as defined below. This process is also used for other changes to the IANA IE registry, such as deprecation or revision, as described later in this section.

### 5.1. The Process for Review by the IE-DOCTORS

Requests to change the IANA IE registry or a linked subregistry are submitted to IANA, which forwards the request to a designated group of experts (IE-DOCTORS) appointed by the IESG; these are the reviewers called for by the Expert Review [RFC5226] policy defined for the IANA IE registry by [RFC7012]. The IE-DOCTORS review the request for such things as compliance with this document, compliance with other applicable IPFIX-related RFCs, and consistency with the currently defined set of Information Elements.

Authors are expected to review compliance with the specifications in this document to check their submissions before sending them to IANA.

The IE-DOCTORS should endeavor to complete referred reviews in a timely manner. If the request is acceptable, the IE-DOCTORS signify their approval to IANA, which changes the IANA IE registry. If the request is not acceptable, the IE-DOCTORS can coordinate with the requestor to change the request to be compliant. The IE-DOCTORS may also choose in exceptional circumstances to reject clearly frivolous or inappropriate change requests outright.

This process should not in any way be construed as allowing the IE-DOCTORS to overrule IETF consensus. Specifically, Information Elements in the IANA IE registry that were added with IETF consensus require IETF consensus for revision or deprecation.

Decisions by the IE-DOCTORS may be appealed as in Section 7 of [RFC5226].

### 5.2. Revising Information Elements

The Information Element status field in the IANA IE registry is defined in [RFC7012] to allow Information Elements to be 'current' or 'deprecated'. No Information Elements are as of this writing deprecated. [RFC5102] additionally specified an 'obsolete' status; however, this has been removed on revision as it served no operational purpose.

In addition, no policy is defined for revising IANA IE registry entries or addressing errors therein. To be certain, changes and deprecations within the IANA IE registry are not encouraged, and should be avoided to the extent possible. However, in recognition that change is inevitable, this section is intended to remedy this situation.

Changes are initiated by sending a new Information Element definition to IANA, as in Section 5.1, for an already-existing Information Element.

The primary requirement in the definition of a policy for managing changes to existing Information Elements is avoidance of interoperability problems; IE-DOCTORS must work to maintain interoperability above all else. Changes to Information Elements already in use may only be done in an interoperable way; necessary changes that cannot be done in a way to allow interoperability with unchanged implementations must result in deprecation.

A change to an Information Element is held to be interoperable only when:

1. it involves the correction of an error that is obviously only editorial; or

2. it corrects an ambiguity in the Information Element's definition, which itself leads to non-interoperability severe enough to prevent the Information Element's usage as originally defined (e.g., a prior change to ipv6ExtensionHeaders); or

3. it expands the Information Element's data type without changing how it is represented (e.g., changing unsigned32 to unsigned64, as with a prior change to selectorId); or

4. it corrects missing information in the Information Element's definition without changing its meaning (e.g., the explicit definition of 'quantity' semantics for numeric Information Elements without a Data Type Semantics value); or

5. it defines a previously undefined or reserved enumerated value, or one or more previously reserved bits in an Information Element with flag semantics; or

6. it expands the set of permissible values in the Information Element's range; or

7. it harmonizes with an external reference that was itself corrected.

If a change is deemed permissible by the IE-DOCTORS, IANA makes the change in the IANA IE registry. The requestor of the change is appended to the requestor in the registry.

Each Information Element in the IANA IE registry has a revision number, starting at zero. Each change to an Information Element following this process increments the revision number by one. Since any revision must be interoperable according to the criteria above, there is no need for the IANA IE registry to store information about old revisions.

When a revised Information Element is accepted into the registry, the date of acceptance of the most recent revision is placed into the revision Date column of the registry for that Information Element.

### 5.3. Deprecating Information Elements

Changes that are not permissible by these criteria may only be handled by deprecation. An Information Element MAY be deprecated and replaced when:

1. the Information Element definition has an error or shortcoming that cannot be permissibly changed as in Section 5.2; or

2. the deprecation harmonizes with an external reference that was itself deprecated through that reference's accepted deprecation method; or

3. changes in the IPFIX protocol or its extensions, or in community understanding thereof, allow the information represented by the Information Element to be represented in a more efficient or convenient way. Deprecation in this circumstance requires a Standards Action.

A request for deprecation is sent to IANA, which passes it to the IE-DOCTORS for review, as in Section 5.1. When deprecating an Information Element, the Information Element description in the IANA

IE registry must be updated to explain the deprecation, as well as to refer to any new Information Elements created to replace the deprecated Information Element.

The revision number of an Information Element is incremented upon deprecation, and the revision Date updated, as with any revision.

Deprecated Information Elements should continue to be supported by Collecting Processes, but should not be exported by Exporting Processes. The use of deprecated Information Elements should result in a log entry or human-readable warning at the Exporting and Collecting Processes.

Names and elementIDs of deprecated Information Elements must not be reused.

## 6. When Not to Define New Information Elements

Due to the relatively limited number space of Information Elements in the IANA IE registry, and the fact that the difficulty of managing and understanding the registry increases with its size, avoiding redundancy and clutter in the registry is important in defining new applications. New Information Elements should not be added to the IANA IE registry unless there is an intent to implement and deploy applications using them; research or experimental applications should use enterprise-specific Information Elements as in Section 6.2 instead.

The subsections below provide guidelines for reuse of existing Information Elements, as well as guidelines on using enterprise-specific Information Elements instead of adding Information Elements in the IANA IE registry.

### 6.1. Maximizing Reuse of Existing Information Elements

Whenever possible, new applications should prefer usage of existing IPFIX Information Elements to the creation of new Information Elements. IPFIX already provides Information Elements for every common Layer 4 and Layer 3 packet header field in the IETF protocol suite, basic Layer 2 information, basic counters, timestamps and time ranges, and so on. When defining a new Information Element similar to an existing one, reviewers should ensure that the existing one is not applicable.

Note that this guideline to maximize reuse does not imply that an Information Element that represents the same information from a packet as an existing Information Element should not be added to the IANA IE registry. For example, consider the ipClassOfService

(Element ID 5), ipDiffServCodePoint (Element ID 98), and ipPrecedence (Element ID 196) Information Elements. These all represent subsets of the same field in an IP version 4 packet header, but different uses of these bits. The representation in one or another of these Information Elements contains information in itself as to how the bits were interpreted by the Metering Process.

On the other hand, simply changing the context in which an Information Element will be used is insufficient reason for the definition of a new Information Element. For example, an extension of IPFIX to log detailed information about HTTP transactions alongside network-level information should not define httpClientAddress and httpServerAddress Information Elements, preferring instead the use of sourceIPv[46]Address and destinationIPv[46]Address.

Applications dealing with bidirectional interactions should use Bidirectional Flow Support for IPFIX [RFC5103] to represent these interactions.

Existing timestamp and time range Information Elements should be reused for any situation requiring simple time stamping of an event: for single observations, the observationTime* Information Elements from PSAMP are provided, and for events with a duration, the flowStart* and flowEnd* Information Elements suffice. This arrangement allows minimal generic time handling by existing Collecting Processes and analysis workflows. New timestamp Information Elements should ONLY be defined for semantically distinct timing information (e.g., an IPFIX-exported record containing information about an event to be scheduled in the future).

In all cases, the use of absolute timestamp Information Elements (e.g., flowStartMilliseconds) is recommended, as these Information Elements allow for maximum flexibility in processing with minimal overhead. Timestamps based on the Export Time header in the enclosing IPFIX Message (e.g., flowStartTimeDeltaMicroseconds) MAY be used if high-precision timing is important, export bandwidth or storage space is limited, timestamps comprise a relatively large fraction of record size, and the application naturally groups records into IPFIX Messages. Timestamps based on information that must be exported in a separate Data Record defined by an Options Template (e.g., flowStartSysUpTime) MAY be used only in the context of an existing practice of using runtime-defined epochs for the given application. New applications should avoid these structures when possible.

### 6.2. Applying Enterprise-Specific Information Elements

IPFIX provides a mechanism for defining enterprise-specific Information Elements, as in Section 3.2 of [RFC7011]. These are scoped to a vendor's or organization's Structure of Management Information (SMI) Private Enterprise Number, and are under complete control of the organization assigning them.

For situations in which interoperability is unimportant, new information should be exported using enterprise-specific Information Elements instead of adding new Information Elements to the IANA IE registry. These situations include:

- export of implementation-specific information, or
- export of information supporting research or experiments within a single organization or closed community, or
- export of information derived in a commercially sensitive or proprietary method, or
- export of information or meta-information specific to a commercially sensitive or proprietary application.

While work within the IETF generally does not fall into these categories, enterprise-specific Information Elements are also useful for pre-standardization testing of a new IPFIX application. While performing initial development and interoperability testing of a new application, the Information Elements used by the application should not be submitted to IANA for inclusion in the IANA IE registry. Instead, these experimental Information Elements should be represented as enterprise-specific until their definitions are finalized.

As this document contains best practices for defining new Information Elements, organizations using enterprise-specific Information Elements are advised to follow the guidelines set forth here even if not submitting Information Elements for inclusion in the IANA IE registry.

## 7. Information Element Definition Checklist

The following three checklists, condensed from the rest of this document, can be used when defining and reviewing Information Elements; they refer back to the section of this document from which they are taken. These checklists are intended for the definition of

new Information Elements; revision should follow the process defined in Section 5.2, and deprecation should follow the process defined in Section 5.3.

Though many of the considerations in this document require the subjective judgement of Information Element authors, reviewers, and IANA, certain parts of the process may be made simpler through tool support. Items on these checklists that could be easily automated or assisted by tools are annotated with "(tool support)". Other items on these checklists require some level of subjective judgement; checks for semantic uniqueness may additionally be supported by textual analysis of descriptions in the future.

Checklist 1 contains conditions that must be met by all proposed Information Elements:

1. The name must be unique within the IANA IE registry, and the name of any current or deprecated Information Element must not be reused. (Section 4.1) (tool support)

2. The description must be sufficiently semantically unique within the IANA IE registry, representing a substantially different meaning from any current or deprecated Information Element. (Section 4)

3. The name must start with a lowercase letter. (Section 4.1) (tool support)

4. Names composed of more than one word must use capital letters for the first letter of each component except for the first one; all other letters are lowercase, even for acronyms. Exceptions are made for acronyms containing a mixture of lowercase and capital letters, such as 'IPv4' and 'IPv6'. (Section 4.1) (tool support)

5. The data type must match the type of the data being represented. (Section 4.2)

6. Data type semantics must be appropriate for the data type. (Section 4.4) (tool support)

7. The Information Element identifier assigned by IANA must be unique. (Section 4.3) (tool support)

8. The Information Element must be reviewed for the potential of information leakage or other misuse that could reduce the security of the measured system; security considerations specific to the Information Element must be discussed in the description or in a supporting RFC. (Section 11)

Checklist 2 contains conditions that must be met by proposed Information Elements with certain properties, as noted:

1. Time values must be defined with appropriate precision. (Section 4.2)

2. Strings and octet arrays with length restrictions must note those length restrictions in their descriptions. (Section 4.2)

3. Enumerations must refer to an IANA IE registry or subregistry, or a registry maintained by an external standards organization. If no suitable registry or subregistry exists, a new subregistry of the IPFIX Information Element registry must be created for the enumeration, to be modified subject to Expert Review [RFC5226]. (Section 4.7)

Checklist 3 contains conditions that should be met by proposed Information Elements:

1. The name of an Information Element pertaining to a specific protocol or application should contain the name of the protocol or application as the first word. (Section 4.1)

2. Information Elements representing integral values should use a data type for the appropriate width for the value. (Section 4.2)

3. Information Elements representing counters or identifiers should be represented as signed64 or unsigned64, unless they are naturally represented with narrower integral types, as appropriate. (Section 4.2)

4. An Information Element should not contain internal structure, subject to the exceptions in Section 4.5; candidate Information Elements with internal structure should be decomposed into multiple Information Elements. (Section 4.5)

5. An Information Element should not contain multiplicity or ordinality information within the definition of the Information Element itself. (Section 4.6)

6. Data type semantics should be defined, if appropriate. (Section 4.4) (tool support)

7. Units should be defined, if appropriate, with new units added to the Information Element Units subregistry if necessary. (Section 4.4) (tool support)

8. Ranges should be defined, if appropriate. (Section 4.4) (tool support)

9. Non-reversible Information Elements (see [RFC5103]) should note non-reversibility in the description. (Section 4.8)

10. Information Elements to be registered with IANA should be intended for implementation and deployment on production networks.

## 8. Applying IPFIX to Non-Flow Applications

At the core of IPFIX is its definition of a Flow, a set of packets sharing some common properties crossing an Observation Point within a certain time window. However, the reliance on this definition does not preclude the application of IPFIX to domains that are not obviously handling Flow data according to this definition. Most network management data collection tasks, those to which IPFIX is most applicable, have at their core the movement of packets from one place to another; by a liberal interpretation of the common properties defining the Flow, then, almost any event handled by these can be held to concern data records conforming to the IPFIX definition of a Flow.

Non-Flow information defining associations or key-value pairs, on the other hand, are defined by IPFIX Options Templates. Here, the Information Elements within an Options Template Record are divided into Scope Information Elements that define the key and non-scope Information Elements that define the values associated with that key. Unlike Flows, Data Records defined by Options Templates are not necessarily scoped in time; these Data Records are generally held to be in effect until a new set of values for a specific set of keys is exported. While this mechanism is often used by IPFIX to export metadata about the collection infrastructure, it is applicable to any association information.

An IPFIX application can mix Data Records described either type of template in an IPFIX Message or Message stream, and exploit relationships among the Flow Keys, values, and Scopes to create interrelated data structures. See [RFC5473] for an example application of this.

## 9. Writing Internet-Drafts for IPFIX Applications

When a new application is complex enough to require additional clarification or specification as to the use of the defined Information Elements, this may be given in an Internet-Draft.

Internet-Drafts for new IPFIX applications are best submitted to a Working Group with expertise in the area of the new application, or to the Independent Submission stream.

When defining new Information Elements in an Internet-Draft, the Internet-Draft should contain a section (or subsection) for each Information Element, which contains the attributes in Section 4 in human-readable form. An example subsection is given below. These Information Element descriptions should not assign Information Element numbers, instead using placeholder identifiers for these numbers (e.g., "TBD1", "TBD2", "TBD3") and a note to IANA in the IANA Considerations section to replace those placeholders in the document with Information Element numbers when the numbers are assigned. The use of these placeholder definitions allows references to the numbers in, e.g., box-and-line diagrams or template definitions as in Section 10.

### 9.1. Example Information Element Definition

This is an example of an Information Element definition that would appear in an Internet-Draft. The name appears in the section title.

Description: Description goes here.; obligatory

Data Type: Data type goes here; obligatory

Data Type Semantics: Data type semantics, if any, go here; optional

Units: Units, if any, go here; optional

Range: Range, if not implied by the data type, goes here; optional

References: References to other RFCs or documents outside the IETF, in which additional information is given, or which are referenced by the description, go here; optional

ElementId: ElementId, if known, or "TBD" if it will be assigned by IANA and filled in at publication time.

### 9.2. Defining Recommended Templates

New IPFIX applications should not, in the general case, define fixed templates for export, as this throws away much of the flexibility afforded by IPFIX. However, fixed template export is permissible in the case that the export implementation must operate in a resource-constrained environment, and/or that the application is replacing an existing fixed-format binary export format in a maximally compatible

way. In any case, Collecting Processes for such applications should support the collection Templates with Information Elements in any order, or Templates with additional Information Elements.

An Internet-Draft clarifying the use of new Information Elements should include any recommended Template or Options Template Records necessary for supporting the application, as well as examples of records exported using these Template Records. In defining these Template Records, such Internet-Drafts should mention, subject to rare exceptions:

1. that the order of different Information Elements within a Template is not significant;

2. that Templates on the wire for the application may also contain additional Information Elements beyond those specified in the recommended Template;

3. that a stream of IPFIX Messages supporting the application may also contain Data Records not described by the recommended Templates; and

4. that any reader of IPFIX Messages supporting the application must accept these conditions.

Definitions of recommended Template Records for Flow-like information, where the Flow Key is well-defined, should indicate which of the Information Elements in the recommended Template are Flow Keys.

Recommended Templates are defined, for example, in [RFC5476] for PSAMP packet reports (Section 6.4.1) and extended packet reports (Section 6.4.2). Recommended Options Templates are defined extensively throughout the IPFIX documents, including in the protocol document itself [RFC7011] for exporting export statistics; in the file format [RFC5655] for exporting file metadata; and in intermediate process definitions such as [RFC6235] for intermediate process metadata. The discussion in these examples is a good model for recommended template definitions.

## 10. A Textual Format for Specifying Information Elements and Templates

Example Templates given in existing IPFIX documents are generally expressed using bitmap diagrams of the respective Templates. These are illustrative of the wire representation of simple Templates, but not particularly readable for more complicated recommended Templates, provide no support for rapid implementation of new Templates, and do not adequately convey the optional nature of ordering and additional

Information Elements. Therefore, we define a recommended textual format for specifying Information Elements and Templates in Internet-Drafts in this section.

Here we define a simple textual syntax for describing IPFIX Information Elements and IPFIX Templates, with human readability, human writability, compactness, and ease of parser/generator implementation without requiring external XML support as design goals. It is intended for use both in human communication (e.g., in new Internet-Drafts containing higher-level descriptions of IPFIX Templates, or describing sets of new IPFIX Information Elements for supporting new applications of the protocol) as well as at runtime by IPFIX implementations.

## 10.1. Information Element Specifiers

The basis of this format is the textual Information Element Specifier, or IESpec. An IESpec contains each of the four important aspects of an Information Element: its name, its number, its type, and its size, separated by simple markup based on various types of brackets. Fully qualified IESpecs may be used to specify existing or new Information Elements within an Information Model, while either fully qualified or partial IESpecs may be used to define fields in a Template.

Bare words are used for Information Element names, and each aspect of information associated with an Information Element is associated with a type of brackets:

- () parentheses for Information Element numbers,
- <> angle brackets for Information Element data types, and
- [] square brackets for Information Element sizes.
- {} curly braces contain an optional space-separated list of context identifiers to be associated with an Information Element, as described in more detail in Section 10.2

The symbol + is reserved for Information Elements nesting within structured data elements; these are described in Section 10.3.

Whitespace in IESpecs is insignificant; spaces can be added after each element in order, e.g., to align columns for better readability.

The basic form of a fully qualified IESpec for an IANA-registered Information Element is as follows:

name(number)<type>[size]

where 'name' is the name of the Information Element in UTF-8, 'number' is the Information Element as a decimal integer, 'type' is the name of the data type as in the IANA informationElementDataTypes registry, and 'size' is the length of the Information Element in octets as a decimal integer, where 65535 or the string 'v' signifies a variable-length Information Element. [size] may be omitted. In this case, the data type's native or default size is assumed.

The basic form of a fully qualified IESpec for an enterprise-specific Information Element is as follows:

name(pen/number)<type>[size]

where 'pen' is the Private Enterprise Number as a decimal integer.

A fully qualified IESpec is intended to express enough information about an Information Element to decode and display Data Records defined by Templates containing that Information Element. Range, unit, semantic, and description information, as in [RFC5610], is not supported by this syntax.

Example fully qualified IESpecs follow:

octetDeltaCount(1)<unsigned64>[8]

octetDeltaCount(1)<unsigned64> (unsigned64 is natively 8 octets long)

sourceIPv4Address(8)<ipv4Address>

wlanSSID(146)<string>[v]

sipRequestURI(35566/403)<string>[65535]

A partial IESpec is any IESpec that is not fully qualified; these are useful when defining templates. A partial IESpec is assumed to take missing values from its canonical definition in the IANA IE registry. At minimum, a partial IESpec must contain a name, or a number. Any name, number, or type information given with a partial IESpec must match the values given in the Information Model; however, size information in a partial IESpec overrides size information in the Information Model; in this way, IESpecs can be used to express reduced-size encoding for Information Elements.

Example partial IESpecs follow:

- octetDeltaCount
- octetDeltaCount[4] (reduced-size encoding)
- (1)
- (1)[4] (reduced-size encoding; note that this is exactly equivalent to an Information Element specifier in a Template)

## 10.2. Specifying Templates

A Template can then be defined simply as an ordered, newline-separated sequence of IESpecs. IESpecs in example Templates illustrating a new application of IPFIX should be fully qualified. Flow Keys may be optionally annotated by appending the {key} context to the end of each Flow Key specifier. A template counting packets and octets per 5-tuple with millisecond precision in IESpec syntax is shown in Figure 1.

```
flowStartMilliseconds(152)<dateTimeMilliseconds>[8]
flowEndMilliseconds(153)<dateTimeMilliseconds>[8]
octetDeltaCount(1)<unsigned64>[8]
packetDeltaCount(2)<unsigned64>[8]
sourceIPv4Address(8)<ipv4Address>[4]{key}
destinationIPv4Address(12)<ipv4Address>[4]{key}
sourceTransportPort(7)<unsigned16>[2]{key}
destinationTransportPort(11)<unsigned16>[2]{key}
protocolIdentifier(4)<unsigned8>[1]{key}
```

Figure 1: Sample Flow Template in IESpec Syntax

An Options Template is specified similarly. Scope is specified appending the {scope} context to the end of each IESpec for a Scope IE. Due to the way Information Elements are represented in Options Templates, all {scope} IESpecs must appear before any non-scope IESpec. The Flow Key Options Template defined in Section 4.4 of [RFC7011] in IESpec syntax is shown in Figure 2.

```
templateId(145)<unsigned16>[2]{scope}
flowKeyIndicator(173)<unsigned64>[8]
```

Figure 2: Flow Key Options Template in IESpec Syntax

### 10.3. Specifying IPFIX Structured Data

IESpecs can also be used to illustrate the structure of the information exported using the IPFIX Structured Data extension [RFC6313]. Here, the semantics of the structured data elements are specified using contexts, and the Information Elements within each structured data element follow the structured data element, prefixed with + to show they are contained therein. Arbitrary nesting of structured data elements is possible by using multiple + signs in the prefix. For example, a basic list of IP addresses with "one or more" semantics would be expressed using partially qualified IESpecs as shown in Figure 3.

```
basicList{oneOrMoreOf}
+sourceIPv4Address(8)[4]
```

Figure 3: Sample basicList in IESpec Syntax

And an example subTemplateList itself containing a basicList is shown in Figure 4.

```
subTemplateList{allOf}
+basicList{oneOrMoreOf}
++sourceIPv4Address(8)[4]
+destinationIPv4Address(12)[4]
```

Figure 4: Sample subTemplateList in IESpec Syntax

This describes a subTemplateMultilist containing all of the expressed set of source-destination pairs, where the source address itself could be one of any number in a basicList (e.g., in the case of SCTP multihoming).

The contexts associable with structured data Information Elements are the semantics, as defined in Section 4.4 of [RFC6313]; a structured data Information Element without any context is taken to have undefined semantics. More information on the application of structured data is available in [RFC6313].

## 11. Security Considerations

The IE-DOCTORS must evaluate the security aspects of new Information Elements in light of the information they could provide to support potential attacks against the measured network or entities about which information is exported. Specific security aspects to evaluate include whether the exported information contains personally identifiable information, or information that should be kept confidential about the described entities (e.g., partial payload, or

configuration information that could be exploited). This is not to say that such Information Elements should not be defined, but there must be an evaluation of the security risk versus the utility of the exported information for the intended application. For example, "A Framework for Packet Selection and Reporting" [RFC5474] concluded in Section 12.3.2 that the hash function's private parameters should not be exported within IPFIX.

Security considerations specific to an Information Element must be addressed in the Security Considerations section of the Internet-Draft describing the Information Element, or in the Information Element description itself in case the Information Element is not defined in an Internet-Draft. Information Elements with specific security considerations should be described in an Internet-Draft.

For example, the ipHeaderPacketSection in the IPFIX IE registry mentions: "This Information Element, which may have a variable length, carries a series of octets from the start of the IP header of a sampled packet. With sufficient length, this element also reports octets from the IP payload, subject to [RFC2804]. See the Security Considerations section". Another example can be seen in the "Packet Sampling (PSAMP) Protocols Specification" [RFC5476]: "In the basic Packet Report, a PSAMP Device exports some number of contiguous bytes from the start of the packet, including the packet header (which includes link layer, network layer, and other encapsulation headers) and some subsequent bytes of the packet payload. The PSAMP Device SHOULD NOT export the full payload of conversations, as this would mean wiretapping [RFC2804]. The PSAMP Device MUST respect local privacy laws."

## 12. Acknowledgments

Thanks to Paul Aitken, Andrew Feren, Dan Romascanu, and David Harrington for their reviews and feedback. Thanks as well to Roni Even and Yoav Nir for their area reviews; and to Pete Resnick, Adrian Farrel, Stephen Farrell, Stewart Bryant, and Barry Leiba for their contributions during IESG discussions. This work is materially supported by the European Union Seventh Framework Programme under grant agreement 257315 (DEMONS).

# 13. References

## 13.1. Normative References

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

[RFC5610] Boschi, E., Trammell, B., Mark, L., and T. Zseby, "Exporting Type Information for IP Flow Information Export (IPFIX) Information Elements", RFC 5610, July 2009.

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

[RFC6313] Claise, B., Dhandapani, G., Aitken, P., and S. Yates, "Export of Structured Data in IP Flow Information Export (IPFIX)", RFC 6313, July 2011.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013.

[RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, September 2013.

## 13.2. Informative References

[RFC2804] IAB IESG, "IETF Policy on Wiretapping", RFC 2804, May 2000.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

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

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

[RFC5472] Zseby, T., Boschi, E., Brownlee, N., and B. Claise, "IP Flow Information Export (IPFIX) Applicability", RFC 5472, March 2009.

[RFC5473] Boschi, E., Mark, L., and B. Claise, "Reducing Redundancy in IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Reports", RFC 5473, March 2009.

[RFC5474] Duffield, N., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, March 2009.

[RFC5476] Claise, B., Johnson, A., and J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, March 2009.

[RFC5560] Uijterwaal, H., "A One-Way Packet Duplication Metric", RFC 5560, May 2009.

[RFC5655] Trammell, B., Boschi, E., Mark, L., Zseby, T., and A. Wagner, "Specification of the IP Flow Information Export (IPFIX) File Format", RFC 5655, October 2009.

[RFC6235] Boschi, E. and B. Trammell, "IP Flow Anonymization Support", RFC 6235, May 2011.

[IANA-IPFIX] IANA, "IP Flow Information Export (IPFIX) Entities", <http://www.iana.org/assignments/ipfix>.

## Appendix A. Example Information Element Definitions

This section contains a few example Information Element definitions as they would appear in an Internet-Draft. Note the conformance of these examples to the guidelines in Section 4.

The sipResponseStatus Information Element (Appendix A.1) illustrates the addition of an Information Element representing Layer 7 application information, with a reference to the registry containing the allowable values. The duplicatePacketDeltaCount Information Element (Appendix A.2) illustrates the addition of a new metric, with a reference to the RFC defining the metric. The ambientTemperature Information Element (Appendix A.3) illustrates the addition of a new measured value outside the area of traditional networking applications.

### A.1. sipResponseStatus

Description: The SIP Response code as an integer, as in the Response Codes registry at http://www.iana.org/assignments/sip-parameters defined in [RFC3261] and amended in subsequent RFCs. The presence of this Information Element in a SIP Message record marks it as describing a SIP response; if absent, the record describes a SIP request.

Data Type: unsigned16

Data Type Semantics: identifier

References: [RFC3261]

ElementId: TBD1

Replaces Enterprise-Specific Element: 35566 / 412

### A.2. duplicatePacketDeltaCount

Description: The number of uncorrupted and identical additional copies of each individual packet in the Flow arriving at the destination since the previous Data Record for this Flow (if any), as measured at the Observation Point. This is measured as the Type-P-one-way-packet-duplication metric defined in Section 3 of [RFC5560].

Data Type: unsigned64

Data Type Semantics: deltaCounter

Units: packets

References: [RFC5560]

ElementId: TBD2

### A.3. ambientTemperature

Description: An ambient temperature observed by measurement equipment at an Observation Point, positioned such that it measures the temperature of the surroundings (i.e., not including any heat generated by the measuring or measured equipment), expressed in degrees Celsius.

Data Type: float

Units: degrees Celsius

Range: -273.15 - +inf

ElementId: TBD3

## Authors' Addresses

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

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

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