

Guidelines for Authors and Reviewers of IPFIX Information Elements

draft-trammell-ipfix-ie-doctors-00

Brian Trammell Benoit Claise

IETF 79, November 6-13, 2010 Beijing, China

Outline

Introduction

The Draft

Table of Contents

Specifications and Guidelines

Templates and Applications

Next Steps

The problem

- ▶ Expansion of IPFIX into new application areas
 - ▶ Application-layer logging (e.g. draft-nicolini-sipclf-ipfix)
 - ▶ New layers (e.g. draft-kashima-ipfix-data-link-layer-monitoring)
 - ▶ New metrics (e.g. draft-akhter-ipfix-perfmon)
- ▶ Most new applications just need new Information Elements
- ▶ Present process is “bring a draft to IPFIX, add IEs with expert review”
- ▶ This needs to be streamlined with diversification of IPFIX
 - ▶ Enable domain experts to specify IPFIX applications with IPFIX expert assistance outside the WG.
 - ▶ Don't write a draft unless one is absolutely necessary.
 - ▶ Improve scalability of expert review.

The solution

- ▶ Appoint IE-DOCTORS, taking inspiration from MIB-DOCTORS
 - ▶ Hi, Nevil!
 - ▶ A longer list of experts scales better.
- ▶ Provide guidelines to three audiences in a BCP
 - ▶ Subject matter experts and authors (e.g., SIPCLF)
 - ▶ IPFIX experts and reviewers (IE-DOCTORS)
 - ▶ IANA
- ▶ Define processes left undefined by 5102 for management of IE registry

Draft contents (1)

1. Introduction

1.1 Intended Audience and Usage

1.2 Overview of relevant IPFIX documents

2. Terminology

- ▶ Defines “application”: “a candidate protocol, task, or domain to which IPFIX export, collection, and/or storage is applied, beyond those within the IPFIX Applicability statement [RFC5472]”
- ▶ By this definition, PSAMP [RFC5476] was the first new IPFIX application after the publication of the IPFIX protocol [RFC5101].

Draft contents (2)

3. How to apply IPFIX
 - ▶ Guidelines on how to determine whether IPFIX fits for an application
4. Defining new Information Elements
 - 4.1 Information Element naming
 - 4.2 Information Element data types
 - 4.3 Ancillary Information Element properties
 - 4.4 Internal structure in Information Elements
 - 4.5 Enumerated Values and Subregistries
 - 4.6 Reversibility as per RFC 5103
5. The Information Element Lifecycle
 - ▶ Defines processes for revising and deprecating Information Elements, left undefined in 5102

Draft contents (3)

6. When not to define new Information Elements
 - 6.1 Maximizing reuse of existing Information Elements
 - 6.2 Applying enterprise-specific Information Elements
7. Applying IPFIX to non-Flow Applications
8. Defining Recommended Templates
 - ▶ Guidelines for defining templates in drafts describing new applications
9. A Textual Format for Specifying Information Elements and Templates
 - ▶ the section formerly known as draft-trammell-ipfix-text-iespec

Information Element guidelines

- ▶ “Make Information Elements that look like those in 5102”
- ▶ Many of these taken direct from 5102 or 5153: this is a superset
- ▶ Descriptive interCapped English names, naming related protocol
- ▶ Use unsigned64/signed64 and reduced size encoding for maximum flexibility with integers, unless there's a native width
- ▶ Data type semantics and units should be defined when appropriate
- ▶ Information elements should have no internal structure
 - ▶ Use Structured Data when necessary
- ▶ Use subregistries when appropriate
- ▶ Non-reversible Information Elements should be noted

Don't make Information Elements you don't need

- ▶ Use existing Information Elements whenever possible:
 - ▶ Simply changing the context in which an Information Element will be used is insufficient reason for the definition of a new Information Element.
 - ▶ Use RFC5103 for reversible Information Elements
 - ▶ Reuse observationTime* timestamps for events, and flow(Start,End) for events with duration.
 - ▶ Use absolute timestamps whenever possible
- ▶ Use enterprise-specific Information Elements when appropriate:
 - ▶ Implementation-specific information
 - ▶ Information derived in a commercially-sensitive or proprietary way
 - ▶ Pre-standardization or experimental testing.

Information Element Lifecycle: Modification

- ▶ Interoperable changes to Information Elements may be made
 - ▶ to correct obviously editorial errors
 - ▶ to correct ambiguities which lead to interoperability problems
 - ▶ to expand the IE's data type without changing representation (e.g. unsigned32 -> unsigned64)
 - ▶ to define a previously undefined enumerated value
 - ▶ to expand the set of permissible values
 - ▶ to harmonize with an external reference
- ▶ Non-interoperable changes may be made if the Information Element has no widespread implementation, as determined by experts and community
- ▶ Changes reviewed by experts.

Information Element Lifecycle: Deprecation

- ▶ Information Elements may be deprecated (and optionally replaced)
 - ▶ when the Information Element definition has an error and cannot be modified
 - ▶ when the deprecation harmonizes with an external reference
 - ▶ when the protocol changes to make the information represented by an Information Element more efficiently exportable: deprecation should be specified in the Internet Draft(s) defining the protocol change.
- ▶ Deprecations reviewed by experts.
- ▶ Deprecated Information Elements become Obsolete after some time.

Information Element Lifecycle: Open Issues

- ▶ New specification, left uncovered in 5102
- ▶ This is a proposal, requires WG input to finalize
- ▶ How do we address versioning?
- ▶ How to handle “community consent” for exceptional changes
- ▶ How long to delay obsolescence of Information Elements?

Specifying Recommended Templates

- ▶ Some applications will require more explanation → Internet-Draft
- ▶ These drafts can specify recommended (*not mandatory*) templates for illustration.
- ▶ Recommended templates:
 - ▶ are order-independent
 - ▶ are extensible
 - ▶ coexist with other templates in a stream
 - ▶ indicate flow keys as appropriate
- ▶ Textual IE Specification provided for simple definition of recommended templates

Textual IE Specification (IESpec)

- ▶ Adapted from draft-trammell-ipfix-text-iespec
- ▶ Information Elements expressed as delimited tuples of *name(number)<type>[length]*
- ▶ Redundant fields can be omitted
- ▶ Templates expressed as simple lists of Information Elements
- ▶ Structured data expressed as nested prefixed lists of Information Elements
- ▶ Easy to write, easy to read
- ▶ Easy to parse in rapid prototyping of new IPFIX applications

WG Adoption

- ▶ This draft specifies procedures for applying IPFIX in the wider community
 - ▶ Including new procedures left undefined in earlier RFCs
- ▶ These rules aren't for implementations, rather for *us* and for external authors
 - ▶ Needs input from WG
 - ▶ Needs input from future IE-DOCTORS and IANA experts
 - ▶ Needs input from external WGs not familiar with IPFIX, and other stakeholders
- ▶ WG adoption useful earlier, rather than later: Treat this draft as a starting point to build upon.