

IPFIX Working Group
Internet-Draft
Intended status: Informational
Expires: October 16, 2008

C. Schmoll
Fraunhofer FOKUS
P. Aitken
B. Claise
Cisco Systems
April 14, 2008

Guidelines for IP Flow Information eXport (IPFIX) Testing
draft-ietf-ipfix-testing-05.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on October 16, 2008.

Internet-Draft

Guidelines for IPFIX Testing

April 2008

Abstract

This document presents a list of tests for implementers of IP Flow Information Export (IPFIX) compliant Exporting Processes and Collecting Processes. This document specifies guidelines for a series of tests that can be run on the IPFIX Exporting Process and Collecting Process in order to probe the conformity and robustness of the IPFIX protocol implementations. These tests cover all important functions, in order to gain a level of confidence in the IPFIX implementation. Therefore they allow the implementer to perform interoperability or plug tests with other IPFIX Exporting Processes and Collecting Processes.

Conventions used in this document

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](#)].

Table of Contents

| | | |
|------------------------|---|--------------------|
| 1. | Introduction | 5 |
| 1.1. | Document Scope | 5 |
| 1.2. | IPFIX Documents Overview | 5 |
| 2. | Terminology | 7 |
| 3. | Test Specifications | 8 |
| 3.1. | Exporting Process / Collecting Process connectivity tests | 8 |
| 3.1.1. | Connectivity tests between Exporting Process and Collecting Process | 8 |
| 3.2. | Template and Data Record tests | 9 |
| 3.2.1. | Transmission of Template with fixed size Information Elements | 9 |
| 3.2.2. | Transmission of Template with variable length Information Elements | 9 |
| 3.2.3. | Set Padding | 10 |
| 3.2.4. | Record Padding | 10 |
| 3.2.5. | Template Withdrawal Message | 11 |
| 3.3. | Information Element tests | 13 |
| 3.3.1. | Enterprise-specific Information Elements | 13 |
| 3.3.2. | Reduced Size Encoding of Information Elements | 13 |
| 3.3.3. | Multiple instances of the same Information Element in one Template | 13 |
| 3.4. | Options Template tests | 14 |
| 3.4.1. | Using any Information Elements as Scope | 14 |
| 3.4.2. | Using multiple Scopes | 15 |
| 3.4.3. | Metering Process Statistics Option Template | 15 |
| 3.4.4. | Metering Process Reliability Statistics Option | |

| | |
|--|--------------------|
| Template | 15 |
| 3.4.5. Exporting Process Reliability Statistics Option Template | 16 |
| 3.4.6. Flow Keys Option Template | 16 |
| 3.5. Stress/Load tests | 17 |
| 3.5.1. Large number of Records for one Template | 17 |
| 3.5.2. Excessive rate of incoming Data Records | 17 |
| 3.5.3. Large Templates | 18 |
| 3.5.4. Many new Templates within Data Template timeout interval | 18 |
| 3.5.5. Multiple Exporting Processes exporting to one Collecting Process | 18 |

| | |
|---|--------------------|
| 3.5.6. Export from one Exporting Process to multiple Collecting Processes | 19 |
| 3.6. Error handling | 19 |
| 3.6.1. Temporary network disconnect | 19 |
| 3.6.2. Exporting Process termination and restart during data transmission | 20 |
| 3.6.3. Collecting Process termination and restart during data transmission | 20 |
| 3.6.4. Incorrect Template Records and Options Template Records | 21 |
| 3.6.5. Incorrect Data Record | 24 |
| 3.6.6. Export of non-matching Template and Data Records . . . | 24 |
| 3.6.7. Incorrect Set IDs | 25 |
| 3.6.8. Re-using Template IDs | 25 |
| 3.7. TLS connectivity and policy selection | 29 |
| 3.7.1. TLS test setup | 29 |
| 3.7.2. TLS over TCP connectivity test | 30 |
| 3.7.3. DTLS over UDP connectivity test | 30 |
| 3.7.4. DTLS over PR-SCTP connectivity test | 30 |
| 3.7.5. TLS bidirectional authentication policy test | 31 |
| 3.7.6. Exporting Process Identity Mismatch TLS Policy test . | 31 |
| 3.7.7. Collecting Process Identity Mismatch TLS Policy test | 31 |
| 4. Security considerations | 33 |
| 5. IANA considerations | 34 |
| 6. Acknowledgements | 35 |
| 7. Normative references | 36 |
| Authors' Addresses | 37 |
| Intellectual Property and Copyright Statements | 38 |

1. Introduction

An IPFIX implementation, whether in software, firmware, or hardware, needs to be tested thoroughly in order to check its robustness, gain confidence in the conformity to the IPFIX documents it is based on.

For a testable IPFIX software toolkit one needs at least one IPFIX Exporting Process and one IPFIX Collecting Process. However when one has for example only implemented a collector then it can be complemented with a 3rd party exporter for these tests.

This document specifies guidelines for a series of tests that can be run on the IPFIX Exporting Process and Collecting Process in order to probe the conformity and robustness of the IPFIX protocol implementations.

The tests listed here can form a valuable common basis for implementers involved in interoperability testing when all of them use these tests to check their own Exporting Process and Collecting Process implementation first.

1.1. Document Scope

This document lists tests intended to be performed between an implementation of an IPFIX Exporting Process and an IPFIX Collecting Process. For some tests multiple instances of each of those components (Observation Points, Metering Process, Exporting Process, Collecting Process) are involved. The testing of those different IPFIX components complicates the testing as usually one tests his software against an existing implementation, which is proven to be compliant. In some cases, two unproven implementations of the Exporting Process and Collecting Process must be tested against each others. The tests range from basic transport connectivity to export of Template and associated Data Records, high load on the Collecting Process, and error condition situations. This document is not intended as a replacement for formal testing software procedures based e.g. on TTCN3 (<http://www.ttcn-3.org/>) but as a best-practices approach to an informal testing of a developer's IPFIX implementation.

1.2. IPFIX Documents Overview

The IPFIX protocol [[RFC5101](#)] provides network administrators with access to IP Flow information. The architecture for the export of measured IP Flow information out of an IPFIX Exporting Process to a Collecting Process is defined in [[I-D.ietf-ipfix-architecture](#)], per the requirements specified in [[RFC3917](#)]. [[I-D.ietf-ipfix-architecture](#)] specifies how IPFIX data records and

Templates are carried via a congestion-aware transport protocol from IPFIX Exporting Processes to IPFIX Collecting Process. IPFIX has a formal description of IPFIX Information Elements, their name, type and additional semantic information, as specified in [[RFC5102](#)]. Finally [[I-D.ietf-ipfix-as](#)] 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.](#) Terminology

IPFIX-specific terminology used in this document is defined in [Section 2 of \[RFC5101\]](#). As in [\[RFC5101\]](#), these IPFIX-specific terms have the first letter of a word capitalized when used in this document.

[3.](#) Test Specifications

The tests described in this section MAY be performed using an IPFIX Exporting Process on one host and an IPFIX Collecting Process on a different host. The configuration of the Observation Point, Metering Process, Exporting Process, and Collection Process SHOULD be recorded for every test along with the test results.

The successful execution of all tests described in this section will give the tester a high confidence that the tested implementation is conformant with the IPFIX architecture and protocol. It does however not provide a 100% comprehensive coverage or formal proof of conformance.

3.1. Exporting Process / Collecting Process connectivity tests

This section lists the basic tests that are preconditions for the more complex tests specified in later sections of this document.

3.1.1. Connectivity tests between Exporting Process and Collecting Process

The tester must create one Exporting Process and one Collecting Process, must configure the Exporting Process to export at least one Template Set and associated Data Records to the Collecting Process, and must cause the Exporting Process to initiate the export.

When the Exporting Process and Collecting Process are to be connected by an SCTP transport, the tester must ensure that an SCTP association is established.

When the Exporting Process and Collecting Process are to be connected by an TCP transport, the tester must ensure that a TCP connection is established.

The tester must ensure that the Transport Session parameters (IP addresses and ports) are correct.

Note that specifying instructions and tools on how to ensure that a Transport Session is correctly established and that the parameters are correct is out of the scope of this document.

The tester must ensure that the Data Records are actually exported. The transmitted data might be observed on-line with an appropriate packet sniffing tool, such as Wireshark (www.wireshark.org). Such tool is also a viable help to check if the initial connection (SCTP, TCP) has been successfully established.

The tester must record which combinations of IPv4 and IPv6 transports, and UDP, SCTP, and TCP transmission protocols are supported, and should perform the test for all the supported combinations.

[3.2.](#) Template and Data Record tests

This section lists tests for checking the correct transmission of IPFIX Template Sets and associated Data Sets.

[3.2.1.](#) Transmission of Template with fixed size Information Elements

The tester must create a Template with a few fixed-size Information Elements where each data type specified in [section 6.1 of \[RFC5101\]](#) (octet, unsigned16, unsigned32 ...) is used at least once, and cause the Template and associated Data Records to be exported over all applicable combinations of transports and protocols in [Section 3.1](#).

The tester must ensure that the Template and associated Data Records were correctly received and decoded by the Collecting Process. For this process the use of verbose debugging output is suggested in order to allow a detailed comparison with the sent (and therefore expected) data.

[3.2.2.](#) Transmission of Template with variable length Information Elements

The tester must create a Template with a mixture of fixed-sized and variable length Information Elements, as specified in [section 7 of \[RFC5101\]](#) and cause the Template and associated Data Records to be exported over all applicable combinations of transports and protocols in [Section 3.1](#).

The tester must ensure that the Template contains at least:

- o a single variable length Information Element
- o a single variable length Information Element followed by a fixed length Information Element
- o a fixed length Information Element followed by a variable length Information Element
- o multiple variable length Information Elements

The tester must ensure that the Template and associated Data Records

were correctly received and decoded by the Collecting Process.

[3.2.3.](#) Set Padding

[Section 3.3.1 of \[RFC5101\]](#) specifies IPFIX Set alignment using padding.

The tester must configure a packet generator to generate two Data Sets with padding in between consisting of zero valued octets, as shown in Figure 1. They must be exported to the Collecting Process, which must correctly decode the Data Sets and all the Data Records.

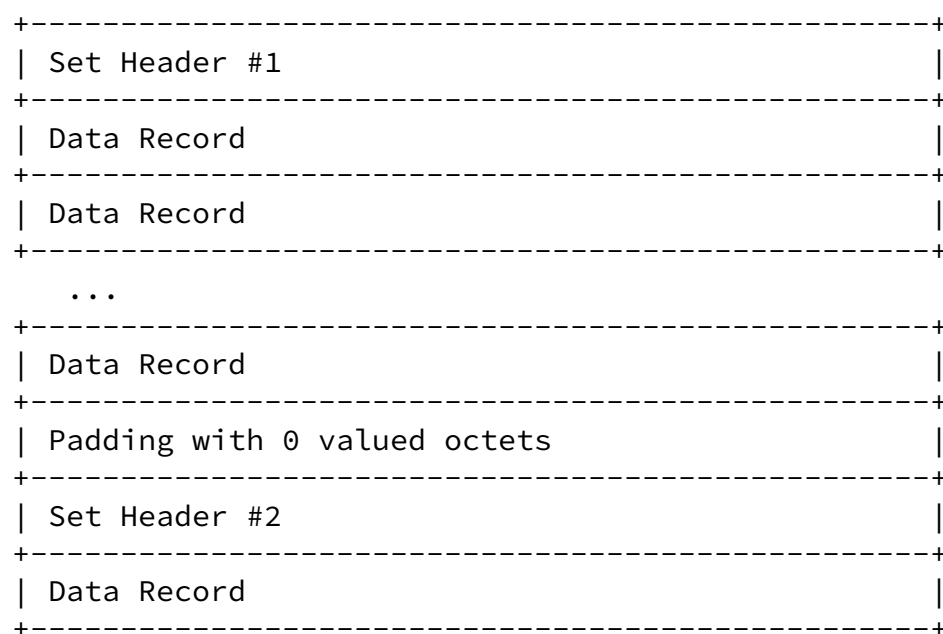


Figure 1

[3.2.4.](#) Record Padding

The tester must configure a packet generator to generate a Template that contains the padding Information Element (i.e. paddingOctets). The Template and associated Data Records must be exported to the Collecting Process, over all applicable combinations of transports and protocols in [Section 3.1](#).

The tester must repeat the test with various padding sizes, including

padding to boundaries other than 4 or 8 octets.

The tester must ensure the Collecting Process correctly interprets case where the Data Records are so short that the padding is equal to or longer than the length of the record, so the padding might otherwise be interpreted as another record (e.g. 1 bytes TOS plus 3 bytes of padding). Refer to the specifications in [section 3.3.1 of \[RFC5101\]](#). Figure 2 depicts such a Template, while Figure 3 depicts a Data Record conforming to that Template.

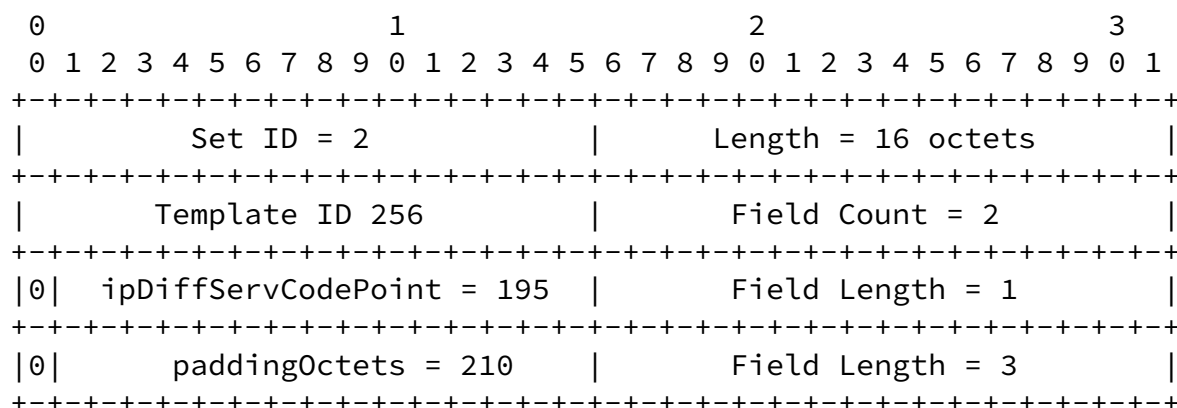


Figure 2

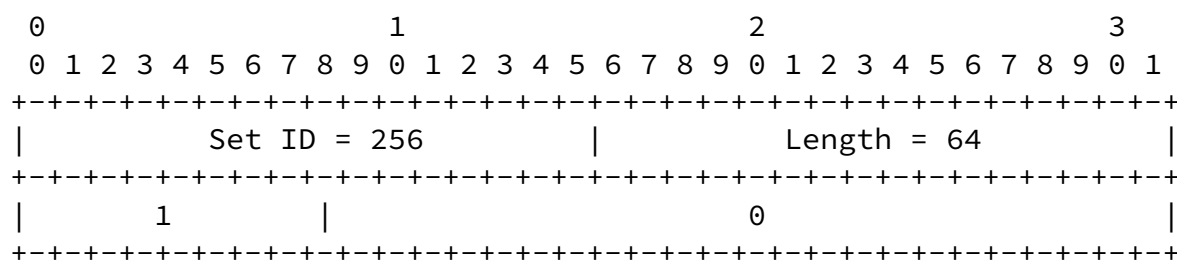


Figure 3

The tester must test fixed-size padding (e.g. 12 bytes of data plus 2 bytes of padding) and variable length padding (e.g. export a string and a variable number of padding bytes afterwards to align the next Information Element to a 4 byte boundary).

3.2.5. Template Withdrawal Message

IPFIX Template management and Template Withdrawal are specified in chapter 8 of [[RFC5101](#)].

[3.2.5.1](#). Withdrawal of a previously sent Template

The tester must create an IPFIX Template and cause that Template to be exported to an IPFIX Collector over a reliable transport.

The tester must check that the Template will be correctly received and decoded by the Collecting Process.

The tester must cause the Exporting Process to send an IPFIX Template Withdrawal Message to the Collector in respect of the Template. The Template Withdrawal Message must be sent over the same Transport Session as the Template.

The tester must ensure that the Template Withdrawal Message was correctly received and decoded by the Collecting Process, and that the previously sent Template was discarded by the Collecting Process.

[3.2.5.2](#). Withdrawal of a previously withdrawn Template

The tester must create, export and withdraw an IPFIX Template as described in [Section 3.2.5.1](#).

The tester must cause the Exporting Process to send a second IPFIX Template Withdrawal Message to the Collector in respect of the same Template. The Template Withdrawal Message must be sent over the same Transport Session as the Template.

The tester must ensure that the Collecting Process discards the IPFIX Message and shuts down the SCTP association or closes the TCP connection. The tester must check that the Collecting Process logged the error.

[3.2.5.3](#). Withdrawal of a previously unsent Template

The tester must cause the Exporting Process to send an IPFIX Template Withdrawal Message to the Collector in respect of a Template which has not yet been exported. The Template Withdrawal Message must be sent over a reliable transport.

The tester must ensure that the Collecting Process discards the IPFIX Message and shuts down the SCTP association or closes the TCP connection. The tester must check that the Collecting Process logged the error.

[3.2.5.4.](#) Withdrawing all Data Templates

The tester must create several IPFIX Templates and cause them to be exported to an IPFIX Collector over a reliable transport.

The tester must ensure that the Templates were correctly received and decoded by the Collecting Process.

The tester must cause the Exporting Process to send an IPFIX All Data Templates Withdrawal Message to the Collector over the same Transport Session as the Templates.

The tester must ensure that the All Data Templates Withdrawal Message was correctly received and decoded by the Collecting Process, and that all the previously sent Templates were discarded by the Collecting Process.

[3.2.5.5.](#) Withdrawing all Option Templates

The tester must create several IPFIX Option Templates and cause them to be exported to an IPFIX Collector over a reliable transport.

The tester must ensure that the Option Templates were correctly received and decoded by the Collecting Process.

The tester must cause the Exporting Process to send an IPFIX All Option Templates Withdrawal Message to the Collector over the same Transport Session as the Templates.

The tester must ensure that the All Option Templates Withdrawal Message was correctly received and decoded by the Collecting Process, and that all the previously sent Option Templates were discarded by the Collecting Process.

[3.3.](#) Information Element tests

This section lists the tests that cover the use of Information Elements.

[3.3.1.](#) Enterprise-specific Information Elements

The tester must cause the export of a Template and associated Data Record that makes use of Enterprise-specific Information Elements as specified in [section 3.2 of \[RFC5101\]](#).

The tester must ensure that the Template and associated Data Record are correctly received and decoded by the Collecting Process, and that Information Elements that are unknown to the Collecting Process are not silently discarded.

[3.3.2.](#) Reduced Size Encoding of Information Elements

The tester must cause the export of a Template and associated Data Record containing Information Elements using reduced-size encoding as specified in [section 6.2 of \[RFC5101\]](#).

The tester must ensure that in the case of Information Elements transmitted using Reduced Size Encoding, the Collecting Process is aware of the real size of each Information Element and not only the reduced size used for its transmission.

[3.3.3.](#) Multiple instances of the same Information Element in one Template

The tester must cause the export of a Template and associated Data

Record containing multiple instances of the same Information Element consecutively.

The tester must ensure that the Collecting Process is able to parse the IPFIX Message, and stores all values received for all the Information Elements that appeared multiple times in the Template definition.

The tester must ensure that the Collecting Process reports the Information Elements in the same order as they were specified in the Template Record, as specified in [section 8 of \[RFC5101\]](#).

The tester must cause the export of another Template and associated Data Record containing multiple instances of the same Information Element with other Information Elements in between.

The tester must ensure that the Collecting Process is able to parse the IPFIX Message, and stores all values received for all the Information Elements that appeared multiple times in the Template definition.

The tester must ensure that the Collecting Process reports the Information Elements in the same order as they were specified in the Template Record, as specified in [section 8 of \[RFC5101\]](#).

[3.4.](#) Options Template tests

This section lists the tests that cover the correct transfer of IPFIX Options Templates.

[3.4.1.](#) Using any Information Elements as Scope

Options Templates contain scope fields that give the context of the reported Information Elements in the corresponding Data Records. Scope fields are an Information Elements specified in [\[RFC5102\]](#).

The tester SHOULD perform the export of Options Template Records containing various different Information Elements of each of the abstract data types specified in [section 6.1 of \[RFC5101\]](#) (octet, unsigned16, unsigned32 ...) in their scope fields, and must export a Data Record using each Template.

The tester must check and if necessary improve the software so that the Templates and the associated Data Records are correctly received and decoded by the Collecting Process.

The tester must ensure that the Collecting Process accepts Information Elements in the scope field other than IPFIX Information

Elements which have been recorded by IANA.

The tester must ensure that the Collecting Process accepts an Enterprise specific Information Element in the scope field.

As specified in [section 3.4.2.1 of \[RFC5101\]](#), the Scope Field Count must NOT be zero. The tester must cause the export of an Options Template Record containing a Scope Field Count of zero.

The tester must ensure that the Collecting Process shuts down the SCTP association and discards the IPFIX Message. The tester should check that the Collecting Process logged the error.

[3.4.2.](#) Using multiple Scopes

The tester must cause the export of an Options Template Record containing multiple scope fields, and a Data Record conforming to that Template.

The tester must ensure that the Collecting Process reports the Information Elements in the same order as they were specified in the Options Template Record, as specified in [section 3.4.2.1 of \[RFC5101\]](#).

[3.4.3.](#) Metering Process Statistics Option Template

The tester must create a Metering Process Statistics Option Templates as specified in [section 4.1 of \[RFC5101\]](#), and cause the Option Template and an associated Data Record to be exported.

The tester must ensure that the Collecting Process correctly receives and decodes the Option Template and associated Data Record.

The tester must also check that the optional meteringProcessId Scope Field is supported by the Collecting Process implementation.

If several Metering Processes are available on the Exporter Observation Domain, the tester must create a Metering Process Statistics Option Template containing multiple scopes and an associated Data Record, must cause the Option Template and associated Data Record to be exported, and must ensure that the Collecting Process correctly receives and decodes the Option Template and associated Data Record.

[3.4.4.](#) Metering Process Reliability Statistics Option Template

The tester must create a Metering Process Reliability Statistics Option Template as specified in [section 4.2 of \[RFC5101\]](#), and must

cause the Option Template and an associated Data Record to be exported.

The tester must ensure that the Collecting Process correctly receives and decodes the Option Template and associated Data Record.

The tester must also check that the optional meteringProcessId Scope Field is supported by the Collecting Process implementation.

[3.4.5.](#) Exporting Process Reliability Statistics Option Template

The tester must create an Exporting Process Reliability Statistics Option Template as specified in [section 4.3 of \[RFC5101\]](#), and must cause the Option Template and an associated Data Record to be exported.

The tester must ensure that the Collecting Process correctly receives and decodes the Option Template and associated Data Record.

[3.4.6.](#) Flow Keys Option Template

The tester must create a Flow Keys Option Template, as specified in [section 4.4 of \[RFC5101\]](#), where the templateId refers to an existing Template, and must cause the Option Template and an associated Data Record to be exported.

The tester must ensure that the Collecting Process correctly receives and decodes the Option Template and associated Data Record, and that the Collecting Process associates the Flow Keys with the right Data Record Information Elements.

The tester must create another Flow Keys Data Record to be exported where the associated templateId has insufficient fields to satisfy the flowKeyIndicator.

The tester must ensure that the Collecting Process discards the IPFIX Message and shuts down the SCTP association or closes the TCP connection. The tester must check that the Collecting Process logged the error.

The tester must create another Flow Keys Option Template, where the templateId refers to a non-existing Template, and must cause the Option Template and an associated Data Record to be exported.

The tester must ensure that the Collecting Process shuts down the SCTP association and discards the IPFIX Message. The tester should check that the Collecting Process logged the error.

[3.5.](#) Stress/Load tests

Stress tests are used to check correct behavior and robustness of an IPFIX Collecting Process implementation when a number of Data Records arrive very quickly. This is especially important when IPFIX over UDP is used, since in that case a slow Collecting Process cannot block the IPFIX Exporting Processes from exporting, since UDP is not congestion aware.

The tests may be dependent upon the hardware and transports technology in use. Therefore the tests may need to be scaled up or down to meet the needs of the particular implementation. However, the implementer must ensure that the implementation is stable under excessive traffic conditions, for whatever definition of "excessive" applies at their intended installation.

The implementer must ensure the correct operation of the Exporting Process and/or Collecting Process when the Collecting Process is incapable of processing records at the rate that they are received.

[3.5.1.](#) Large number of Records for one Template

The tester should export many Data Records to the Collecting Process, all conforming to the same Template, in order to put the Collecting Process under stress.

Depending on what that the Collecting Process does (save to file, store to database, analyze the data) the Collecting Process may use up a lot of memory.

The tester must ensure that, if the Collecting Process runs out of memory, it shuts down the specific SCTP association or closes the TCP connection but remains available to receive data on other open Transport Sessions and also stays available for future Transport Sessions.

[3.5.2.](#) Excessive rate of incoming Data Records

The tester should perform a test where Data Records are exported to the Collecting Process with an increasing export rate.

For TCP or SCTP in reliable mode, the tester must ensure that export stalls the Exporting Process once the Collecting Process becomes fully loaded.

For UDP export, the tester must ensure that the Collecting Process drops records as it becomes overloaded, and must check that the Collecting Process logs a warning.

[3.5.3.](#) Large Templates

The tester must create Templates with the maximum possible number of Information Elements, and cause these to be exported to the Collecting Process.

The total length field in the IP header is 16 bits, allowing a length up to 65535 octets. 20 octets are required for a minimal IPv4 header, 16 octets for the IPFIX header, 4 octets for the Set header and 4 octets for the Template header, so the Template definition may be up to $(65535 - 20 - 16 - 4 - 4) = 65491$ octets long. The minimum IPFIX Information Element specification requires 4 octets: two for the Information Element ID and two for the field length. Therefore, the maximum number of IPFIX Information Elements in a single Template is $65491 / 4 = 16372$. With this many Information Elements, the Template will be 65488 octets long while the entire packet will be 65532 octets long.

The tester must create Data Records conforming to this Template, and cause them to be exported. Note that, for the implementation, the associated Data Records might be smaller or larger than the Template Records depending on the length of the Information Elements defined by the Template and on the presence of variable length Information Elements.

The tester must ensure that the Collecting Process correctly receives and decodes the Template and associated Data Records.

[3.5.4.](#) Many new Templates within Data Template timeout interval

The tester should create a large number of different Templates and cause them to be exported to the Collecting Process to stress test the Collecting Process's memory consumption.

The tester must ensure that the Collecting Process gracefully discards Templates if it's running out of memory resources, and should check that warnings are logged.

[3.5.5.](#) Multiple Exporting Processes exporting to one Collecting Process

The tester must configure multiple Exporting Processes to export Templates and associated Data Records to the same Collecting Process at the same time.

The tester must ensure that all the Templates and associated Data Records are correctly received and decoded at the Collecting Process, and that no Exporting Process stalls or disconnects completely unless the Collecting Process runs out of memory.

Schmoll, et al.

Expires October 16, 2008

[Page 18]

Internet-Draft

Guidelines for IPFIX Testing

April 2008

[3.5.6.](#) Export from one Exporting Process to multiple Collecting Processes

If the Exporting Process supports multiple simultaneous export destinations, the tester must configure the Exporting Process to export Data Records in parallel to different Collecting Processes.

The tester must configure the use of a mixture of simple and complex Templates and ensure they are all correctly received and decoded by all the Collecting Processes.

[3.6.](#) Error handling

This section lists and describes a number of problems that might occur in either the network or data transmission or related to wrong information encoding, and which the IPFIX Exporting Process and Collecting Process must be capable of handling in a graceful way. It is intended to test the robustness and fault tolerance of the IPFIX Processes.

[3.6.1.](#) Temporary network disconnect

The IPFIX Exporting Process and Collecting Process behavior must be checked upon interruptions of data transmission due to network failures (whether physical or logical, e.g. defective routing).

The tester must configure continuous export over all applicable combinations of transports and protocols in [Section 3.1](#) in turn.

For SCTP-based associations and TCP-based connections, the tester should create a short disconnect between the Exporting Process and the Collecting Process (e.g. by momentarily interrupting the network connection) and must ensure that export continues after the connection is repaired. The tester must then create a longer disconnection between the Exporting Process and Collecting Process, and must ensure that export continues after the connection is repaired.

For UDP-based data export there is no noticeable connection loss, but data received with non-consecutive sequence numbers indicates data loss. Refer to the sequence number specifications in [section 3.1 of \[RFC5101\]](#). The tester should create a short disconnect between the Exporting Process and Collecting Process, and must ensure that this is recognized and reported by the Collecting Process per [section 3.1 of \[RFC5101\]](#).

[3.6.2.](#) Exporting Process termination and restart during data transmission

An IPFIX Collecting Process might be confronted with a faulty Exporting Process implementation that suddenly crashes, dropping any open connections. The Exporting Process may be restarted again soon after the crash.

Such an event will only be visible to the Collecting Process when the IPFIX Messages (Templates and associated Data Records) are carried over TCP or SCTP. For export via UDP no such test is available due to the connection-less nature of the transport.

The tester must configure continuous export over all applicable combinations of SCTP and TCP transports and protocols in [Section 3.1](#) in turn. For each combination, the tester must establish export, then kill the active Exporting Process.

The tester must ensure that the associated Collecting Process shuts

down SCTP associations and closes TCP connections associated with that export after a suitable timeout period.

The tester must Ensure that the Collecting Process discards the Template(s) received on the killed transport session.

The tester must restart the Exporting Process again, and must ensure that the Exporting Process exports the Templates again.

The tester must ensure that the Collecting Process receives and accepts both Templates and associated Data Records from the new Exporting Process running at the same source host.

3.6.3. Collecting Process termination and restart during data transmission

An IPFIX Exporting Process might be confronted with a faulty Collecting Process implementation that suddenly crashes, dropping any open Transport Sessions. The Collecting Process may be restarted again soon after the crash.

The tester must set up an Exporting Process and Collecting Process and cause IPFIX Templates and associated Data Records to be exported over all applicable combinations of SCTP and TCP transports and protocols in [Section 3.1](#) in turn. Via UDP the restart of the Collecting Process will be invisible to the Exporting Process and have no effect.

The tester must terminate the Collecting Process while the export is

in progress, and must ensure that the Exporting Process shuts down SCTP associations and closes TCP connections associated with that Collecting Process.

The tester must restart the Collecting Process and ensure that the Exporting Process connects to the Collecting Process again and that it exports the IPFIX Templates again.

The tester must ensure that the new Collecting Process correctly receives and decodes the IPFIX Data Records again.

3.6.4. Incorrect Template Records and Options Template Records

These tests verify the Collecting Process's operation when it receives a Template Record or Options Template Record with an invalid message length. Refer to the specifications in [section 3.4.1](#) and [3.4.2](#) of [RFC5101], respectively.

Consider the example Template Record shown in Figure 4. This Template record is missing one Information Element ID and one Information Element length field. There is insufficient data in the Set for the specified Set length, and the overall record is four octets too short for the specified total length.

[illegible]

Figure 4

The tester must create and cause the Exporting Process to export the following IPFIX Templates, and must ensure the correct Collecting Process behaviour for each of the transports and protocols in [Section 3.1](#).

- o For SCTP transport, ensure that the Collecting Process discards the IPFIX Message and shuts down the SCTP association, and check that it logs the error.
- o For TCP transport, ensure that the Collecting Process discards the

IPFIX Message, closes the TCP connection, and check that it logs the error. Note that since TCP is a streaming (rather than record-based) protocol, template length errors cannot be detected and may cause framing to be lost, potentially rendering the remainder of the IPFIX stream unintelligible. Therefore some of these tests are not applicable for TCP transport, as indicated.

- o For UDP transport, ensure that the Collecting Process discards the IPFIX Message, and check that it logs the error.

(a) The tester must create the IPFIX Template shown in Figure 4 and cause the Exporting Process to export it. The tester must ensure that the Collecting Process's behaviour is as specified above for each transport type except for TCP, for which this test is not applicable.

(b) Consider the IPFIX Template shown in Figure 4, modified with total length = 28. In this case the IPFIX Message has to be rejected because field count = 2 and there is no second Information Element record present in the Set. The available data is exhausted after reading the first Information Element record.

The tester must create the modified Template and cause the Exporting Process to export it. The tester must ensure that the Collecting Process's behaviour is as specified above for each transport type.

(c) Consider the IPFIX Template shown in Figure 4, modified with total length = 26. In this case the IPFIX Message has to be rejected because the IPFIX Message length is too short. After the first Information Element the IPFIX Message data is exhausted according to the total length information.

The tester must create the modified Template and cause the Exporting Process to export it. The tester must ensure that the Collecting Process's behaviour is as specified above for each transport type. The TCP connection used for this test must be manually reset after the test.

(d) Consider the IPFIX Template shown in Figure 4, modified with field count = 1. In this case the IPFIX Message must be rejected because total length is too large and does not match the amount of data available.

The tester must create the modified Template and cause the Exporting Process to export it. The tester must ensure that the Collecting Process's behaviour is as specified above for each transport type. This test is not applicable for TCP transport.

(e) Finally when the IPFIX Template shown in Figure 4 is extended with the data shown in Figure 5, it becomes a correct IPFIX Template.

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0| Info Element Identifier = 12|           Field Length = 4           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 5

The tester must create the modified Template and cause the Exporting Process to export it. The tester must ensure that the Template is accepted by the Collecting Process for each transport type.

The example Template record shown in Figure 6 must be dropped because the scope field count = 0.

```

      0              1              2              3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Version = 10           |           Total Length = 30           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Export Time = 1155202151           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Sequence Number = 0x12345678           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Observation Domain ID = 0x33334444           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Set ID = 3           |           Set Length = 14           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Template ID = 257           |           Field Count = 1           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Scope Field Count = 0           |0| Info Element Identifier = 8 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Field Length = 4           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 6

The tester must create the Template shown in Figure 6 and cause the Exporting Process to export it. The tester must ensure that the IPFIX Message is discarded by the Collecting Process for each transport type, and must check that the Collecting Process logs an error. The tester must ensure that the Collecting Process also shuts

down the SCTP association or closes the TCP connection.

The tester must create an IPFIX Options Template where the field count is less than the scope field count, and cause the Exporting Process to export it. Use the above IPFIX Options Template with scope field count = 2. The tester must ensure that the Template is discarded by the Collecting Process for each transport type, and must check that the Collecting Process logs an error. The tester must ensure that the Collecting Process shuts down the SCTP association or closes the TCP connection.

[3.6.5.](#) Incorrect Data Record

The tester must create the following invalid Data Records and cause them to be exported to the Collecting Process over all applicable combinations of transports and protocols in [Section 3.1](#).

- o IPFIX Message too short.
- o Illegal use of reduced size encoding.
- o Invalid length specification in case of variable length Information Elements.

The tester must ensure that the Collecting Process discards the IPFIX Message and shuts down the SCTP association or closes the TCP connection.

[3.6.6.](#) Export of non-matching Template and Data Records

The tester must create Templates and associated Data Records that fail to conform to those Templates in the following ways:

- o too few Information Elements in Data Record
- o too many Information Elements in Data Record

The tester must cause the Templates and associated Data Records to be exported to the Collecting Process over all applicable combinations of transports and protocols in [Section 3.1](#).

The tester must ensure that the Collecting Process discards the IPFIX

Message and shuts down the SCTP association or closes the TCP connection.

[3.6.7.](#) Incorrect Set IDs

The tester must create Template Sets, Option Template Sets, and associated Data Sets with an incorrect Set ID, and cause these to be exported to the Collecting Process over all applicable combinations of transports and protocols in [Section 3.1](#).

Per [\[RFC5101\] section 3.3.2](#), only the Set ID values 2 and 3 denote valid Sets.

The tester must ensure that the Collecting Process discards the IPFIX Message and shuts down the SCTP association or closes the TCP connection.

[3.6.8.](#) Re-using Template IDs

[3.6.8.1.](#) Using SCTP transport.

Refer to [section 9 of \[RFC5101\]](#) for the Collecting Process's SCTP Template management specifications.

The tester must create an IPFIX Template and cause it to be exported to a Collecting Process over SCTP transport.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

The tester must cause the same Template to be exported to the same Collecting Process over the same SCTP association, and must ensure that the Collecting Process resets the SCTP association and discards the IPFIX Message.

The tester must create another IPFIX template and cause it to be exported to the Collecting Process over SCTP transport.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

The tester must modify the Template contents while retaining the same Template ID.

The tester must cause the modified Template to be exported to the same Collecting Process over the SCTP same association, and must ensure that the Collecting Process resets the SCTP association and discards the IPFIX Message.

The tester must check that an error was logged.

The tester must create another IPFIX Template and cause it to be exported to the Collecting Process over SCTP transport.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

The tester must cause a Template Withdrawal Message for the Template to be sent to the Collecting Process over the same SCTP association, and must ensure that the Template has been discarded by the Collecting Process.

The tester must create Data Records conforming to the Template and cause them to be exported to the Collecting Process over the same SCTP association.

The tester must ensure that the Collecting Process discards the Data Records and logs a warning.

The tester must cause the same Template to be exported to the same Collecting Process over the same SCTP association.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

The tester must create Data Records conforming to the Template and cause them to be exported to the Collecting Process over the same SCTP association.

The tester must ensure that the Collecting Process correctly receives and decodes the Data Records.

3.6.8.2. Using TCP transport.

Refer to [section 10.4.3 of \[RFC5101\]](#) for the Collecting Process's TCP Template management specifications.

The tester must create an IPFIX Template and cause it to be exported to a Collecting Process over TCP transport.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

The tester must cause the same Template to be exported to the same Collecting Process over the same TCP connection, and must ensure that the Collecting Process resets the TCP connection and discards the IPFIX Message.

The tester must create an IPFIX Template and cause it to be exported

to the Collecting Process over TCP transport.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

The tester must modify the Template contents while retaining the same Template ID.

The tester must cause the modified Template to be exported to the same Collecting Process over the same TCP connection, and must ensure that the Collecting Process resets the TCP connection and discards the IPFIX Message.

The tester must check that an error was logged.

The tester must create another IPFIX Template and cause it to be exported to the Collecting Process over TCP transport.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

The tester must cause a Template Withdrawal Message for the Template to be sent to the Collecting Process over the same TCP connection, and must ensure that the Template has been discarded by the Collecting Process.

The tester must create Data Records conforming to the same Template and cause them to be exported to the same Collecting Process over the same TCP connection.

The tester must ensure that the Collecting Process discards the Data Records and logs a warning.

The tester must cause the same Template to be exported to the same Collecting Process over the same TCP connection.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

The tester must create Data Records conforming to the Template and cause them to be exported to the same Collecting Process over the same TCP connection.

The tester must ensure that the Collecting Process correctly receives and decodes the Data Records.

[3.6.8.3](#). Using UDP transport.

Refer to sections [10.3.6](#) and [10.3.7](#) of [[RFC5101](#)] for the UDP Template management specifications.

[3.6.8.3.1](#). Re-using the same Template ID inside the Template lifetime

The tester must create an IPFIX Template and cause it to be exported to a Collecting Process over UDP transport.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

Before the Template lifetime expires on the Collecting Process, the

tester must cause the same Template to be exported over the same UDP connection to the same Collecting Process and must ensure that the Collecting Process accepts the Template.

The tester must create a different Template with the same ID and must cause this to be exported to the same Collecting Process over the same UDP connection before the original Template lifetime expires.

The tester must ensure that the Collecting Process does not reject the new Template.

The tester must create Data Records conforming to the new Template and cause them to be exported to the same Collecting Process over the same UDP connection.

The tester must ensure that the Collecting Process correctly receives and decodes the Data Records.

3.6.8.3.2. Re-using the same Template ID after the Template lifetime

The tester must create an IPFIX Template and cause it to be exported to a Collecting Process over UDP transport.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

The tester must allow the received Template lifetime to expire on the Collecting Process.

The tester must create Data Records conforming to the Template and cause them to be exported to the same Collecting Process over the same UDP connection.

The tester must ensure that the Collecting Process discards the Data

Records.

The tester must check that the Collecting Process logs a warning.

The tester must cause the same Template to be exported to the same Collecting Process over the same UDP connection.

The tester must ensure that the Template was correctly received and decoded by the Collecting Process.

The tester must create Data Records conforming to the Template and cause them to be exported to the same Collecting Process over the same UDP connection.

The tester must ensure that the Collecting Process correctly receives and decodes the Data Records.

[3.7.](#) TLS connectivity and policy selection

This section lists tests that verify connectivity over TLS and DTLS, and proper selection of TLS policies as specified in the IPFIX Protocol. It specifically does NOT purport to test the security of IPFIX Message transport over TLS or DTLS, as evaluating the security of a transport session is really a test of the TLS or DTLS implementation over which a given IPFIX implementation runs, and as such is out of scope for this document.

Refer to [section 11 of \[RFC5101\]](#) for the security specifications.

[3.7.1.](#) TLS test setup

Setting up for TLS connectivity and policy testing requires the creation of appropriate X.509 certificates and private keys for a test environment, and the configuration of a DNS server to answer with consistent information for the hosts used in the test.

The tester must configure the following certificates:

1. A Certificate Authority (CA) certificate and associated private key for signing the following certificates.
2. One certificate and associated private key, with a CN (Common Name) or subjectAltName extension of type dNSName containing the fully qualified domain name of the host, signed by the CA certificate in 1, for each IPFIX Exporting Process in the test.
3. One certificate and associated private key, with a CN (common name) or subjectAltName extension of type dNSName containing the

fully qualified domain name of the host, signed by the CA certificate in 1, for each IPFIX Collecting Process in the test.

The tester must configure consistent forward (A, AAAA) and reverse (PTR) DNS records for each host in the test on a DNS server used by the hosts for name resolution.

The tester must ensure that the Exporting Process and Collecting Process are on different hosts.

[3.7.2.](#) TLS over TCP connectivity test

The tester must set up certificates and DNS as in [Section 3.7.1](#).

The tester must configure one Exporting Process and one Collecting Process with their appropriate certificates to transfer IPFIX Messages over TLS over TCP.

The tester must create an IPFIX Template and associated Data Record, and cause them to be exported over the TCP connection.

The tester must ensure that a TCP connection and a TLS connection were established, must ensure that data was exchanged, and must ensure that the data received at the Collecting Process is correct.

[3.7.3.](#) DTLS over UDP connectivity test

The tester must set up certificates and DNS as in [Section 3.7.1](#).

The tester must configure one Exporting Process and one Collecting Process with their appropriate certificates to transfer IPFIX Messages over DTLS over UDP.

The tester must create an IPFIX Template and associated Data Record, and cause them to be exported over the UDP connection.

The tester must ensure that UDP packets were sent and a DTLS connection was established, must ensure that data was exchanged, and must ensure that the data received at the Collecting Process is correct.

[3.7.4.](#) DTLS over PR-SCTP connectivity test

The tester must set up certificates and DNS as in [Section 3.7.1](#).

The tester must configure one Exporting Process and one Collecting Process with their appropriate certificates to transfer IPFIX Messages over DTLS over PR-SCTP.

The tester must create an IPFIX Template and associated Data Record, and cause them to be exported over the SCTP association.

The tester must ensure that an SCTP association and a DTLS connection were established, must ensure that data was exchanged, and must ensure that the data received at the Collecting Process is correct.

[3.7.5.](#) TLS bidirectional authentication policy test

This is an optional test for Collecting Processes only; it requires the modification of an Exporting Process to NOT present a certificate.

The tester must modify an Exporting Process to NOT present a certificate.

The tester must perform the connectivity tests in [Section 3.7.2](#), [Section 3.7.3](#), and [Section 3.7.4](#).

The tester must ensure that the Collecting Process rejects the TLS or DTLS connection establishment.

[3.7.6.](#) Exporting Process Identity Mismatch TLS Policy test

The tester must set up certificates and DNS as in [Section 3.7.1](#).

The tester must use a certificate for the Exporting Process that does NOT match the fully qualified domain name of the host on which the Exporting Process runs.

The tester must Perform the connectivity tests in [Section 3.7.2](#), [Section 3.7.3](#), and [Section 3.7.4](#).

The tester must ensure that the Collecting Process rejects the TLS or DTLS connection establishment.

[3.7.7.](#) Collecting Process Identity Mismatch TLS Policy test

The tester must set up certificates and DNS as in [Section 3.7.1](#).

The tester must use a certificate for the Collecting Process that does NOT match the fully qualified domain name of the host on which the Collecting Process runs.

The tester must perform the connectivity tests in [Section 3.7.2](#), [Section 3.7.3](#), and [Section 3.7.4](#).

The tester must ensure that the Exporting Process rejects the TLS or

| | | |
|-----------------|--------------------------|-----------|
| Schmoll, et al. | Expires October 16, 2008 | [Page 31] |
|-----------------|--------------------------|-----------|

| | | |
|----------------|------------------------------|------------|
| Internet-Draft | Guidelines for IPFIX Testing | April 2008 |
|----------------|------------------------------|------------|

DTLS connection establishment.

Schmoll, et al.

Expires October 16, 2008

[Page 32]

Internet-Draft

Guidelines for IPFIX Testing

April 2008

[4.](#) Security considerations

This memo raises no security issues.

Schmoll, et al.

Expires October 16, 2008

[Page 33]

Internet-Draft

Guidelines for IPFIX Testing

April 2008

[5.](#) IANA considerations

This memo raises no IANA considerations.

[6.](#) Acknowledgements

The authors wish to thank Brian Trammell for contributing the initial text for [Section 3.7](#).

7. Normative references

[I-D.ietf-ipfix-architecture]

Sadasivan, G., "Architecture for IP Flow Information Export", [draft-ietf-ipfix-architecture-12](#) (work in progress), September 2006.

[I-D.ietf-ipfix-as]

Zseby, T., "IPFIX Applicability", [draft-ietf-ipfix-as-12](#) (work in progress), July 2007.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3917] Quittek, J., Zseby, T., Claise, B., and S. Zander, "Requirements for IP Flow Information Export (IPFIX)", [RFC 3917](#), October 2004.
- [RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", [RFC 5101](#), January 2008.
- [RFC5102] Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, "Information Model for IP Flow Information Export", [RFC 5102](#), January 2008.

Carsten Schmoll
Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
Berlin D-10589
Germany

Phone: +49 30 3463 7136
Email: carsten.schmoll@fokus.fraunhofer.de
URI: <http://www.fokus.fraunhofer.de>

Paul Aitken
Cisco Systems
96 Commercial Quay
Edinburgh EH6 6LX
Scotland

Phone: +44 131 561 3616
Email: paitken@cisco.com
URI: <http://www.cisco.com>

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

Phone: +32 2 704 5622
Email: bclaise@cisco.com
URI: <http://www.cisco.com>

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

