

IPFIX Working Group
Internet-Draft
Expires: August 17, 2006

C. Schmoll
Fraunhofer Institute Fokus
P. Aitken
Cisco Systems
February 13, 2006

IP Flow Information Exchange (IPFIX) Testing
draft-schmoll-ipfix-testing-00.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/1id-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 August 17, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document presents a list of tests which implementers of IP Flow Information Export (IPFIX) compliant systems are encouraged to perform on their IPFIX system. This document has been created to help implementers test the functionality of their IPFIX exporter and/or collector component. The goal of these tests is to ensure that all important functions are covered by tests and thereby to gain a level of confidence in the system which allows the implementer to

perform interoperability or plug tests with other IPFIX systems.

Table of Contents

1.	Introduction	4
1.1.	Motivation	4
1.2.	Document Scope	4
1.3.	Related Documents	4
2.	Terminology	6
3.	Test Specifications	7
3.1.	Exporter/Collector Connectivity Tests	7
3.1.1.	Connectivity Tests between Exporter and Collector	7
3.2.	Data Template and Data Transmission Tests	7
3.2.1.	Transmission of Simple Data Template and Data	7
3.2.2.	Transmission of Data Template with variable-length IEs and Data	7
3.2.3.	Flowsets with Padding	8
3.3.	IE Tests	8
3.3.1.	Enterprise-specific IEs	8
3.3.2.	Reduced-size Encoding of IEs	8
3.3.3.	Multiple use of the same IE in one Template	8
3.4.	Options Templates	9
3.4.1.	Using any IEs as Scope	9
3.4.2.	Using multiple Scopes	9
3.4.3.	Metering Process (MP) Statistics Option Template	9
3.4.4.	Metering Process (MP) Reliability Statistics Option Template	9
3.4.5.	Exporting Process (EP) Reliability Statistics Option Template	10
3.4.6.	Flow Keys Option Template	10
3.4.7.	Template Withdrawal Message	10
3.5.	Stress/Load Tests	10
3.5.1.	Large Number of Records for one Template	11
3.5.2.	High Rate of incoming Data Records	11
3.5.3.	Large Templates with high Number of IEs	11
3.5.4.	Many new Templates within Data Template timeout interval	11
3.5.5.	Multiple Exporters sending to one Collector	11
3.5.6.	Export from one Exporter to multiple Collectors	11
3.6.	Error Handling	12
3.6.1.	Temporary Network Disconnect	12
3.6.2.	Exporter Termination and Restart during Data Transmission	12
3.6.3.	Collector Termination and Restart during Data Transmission	12
3.6.4.	Incorrect Template Records	13
3.6.5.	Export of defective IPFIX Data record	13

3.6.6.	Export of non-matching Template and Data	13
3.6.7.	Incorrect Set IDs	13
3.6.8.	Flowsets with Invalid Padding	13
3.6.9.	Re-using the same Template ID inside the Template Expiry Time	14
3.6.10.	Re-using the same Template ID after the Template Expiry Time	14
3.6.11.	Re-sending an existing template ID without withdrawal	14
4.	Security Considerations	15
5.	References	15
	Authors' Addresses	16
	Intellectual Property and Copyright Statements	17

1. Introduction

The IPFIX protocol has been developed for the purpose of exporting IP flow information from devices such as routers or measurement stations to mediation, accounting, and network management systems. It is intended for the purposes of Internet research, QoS and traffic measurement, attack and intrusion detection reporting, accounting, and billing.

The IPFIX architecture [[I-D.ietf-ipfix-architecture](#)] defines the different components which are involved in this data export process. For a testable IPFIX software toolkit one needs at least the IPFIX exporter and the IPFIX collector component. The exporter component communicates information regarding flows from a location close to the point of measurement in the network to the collector via SCTP, TCP, or UDP. The collector may then e.g., store this data into a database or transfer it directly to an application for further processing.

An implementation of these IPFIX components in software, firmware, or hardware needs to be tested thoroughly in order to check its robustness and the conformity to the IPFIX drafts it is based on. This document suggests tests which should be run in order to check the system and to gain a high confidence in the conformity, robustness, and correct behavior of such implementation.

1.1. Motivation

The main driving force for preparing this document is the observation that protocols for data exchange often fail to work properly when implementations from different companies or organizations are in use together. In many cases this even holds true when tests had previously been performed successfully using an exporting and collecting process from a single implementer. 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 exporter and collector first.

1.2. Document Scope

This document lists tests intended to be performed between an implementation of an IPFIX exporter and an IPFIX collector. For some tests multiple instances of each of those components are involved. The tests cover basic application connectivity, export of template and data records, high load, and error condition situations.

1.3. Related Documents

This draft refers to the following draft documents: "Information

Model for IP Flow Information Export" [[I-D.ietf-ipfix-info](#)] and
"IPFIX Protocol Specification" [[I-D.ietf-ipfix-protocol](#)].

2. Terminology

The terminology used in this document is fully aligned with the terminology defined in [[I-D.ietf-ipfix-architecture](#)] and [I-D.ietf-ipfix-protocol].

In the remainder of this document IE means Information Element.

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](#)].

3. Test Specifications

The following tests SHOULD be performed using an IPFIX exporting process on one host and an IPFIX collecting process on a different host. The network configuration and software component setup SHOULD be recorded. The test results SHOULD be recorded per test performed.

3.1. Exporter/Collector Connectivity Tests

This section lists the basic tests which must succeed as a precondition for the more complex tests in later sections.

3.1.1. Connectivity Tests between Exporter and Collector

Setup one exporting and one collecting process. Configure the exporting process to send to the collecting process. Configure a minimal data set so that the exporter will initiate a connection. Detect whether a connection was established (in case of SCTP and TCP) and whether data was exchanged. The transmitted data might be observed on-line with an appropriate tool such as Ethereal (www.ethereal.com).

Perform the test for all the supported combinations of IPv4 and IPv6 and UDP, SCTP, and TCP as transmission protocol.

3.2. Data Template and Data Transmission Tests

This section lists the important tests for checking the correct transmission of IPFIX templates and data sets.

3.2.1. Transmission of Simple Data Template and Data

Create and export an IPFIX data template and data record for a few fixed-size IEs over the transports in [Section 3.1](#). Verify the correct reception and decoding of the template and data. Use various IEs so that each data type (octet, unsigned16, unsigned32 ...) is used in at least one test.

3.2.2. Transmission of Data Template with variable-length IEs and Data

Create and export data templates and data records for a mixture of fixed-sized and variable-length IEs over the transports in [Section 3.1](#). The various templates should contain:

- o a single variable-length IE
- o a single variable length IE followed by a fixed length IE

- o a fixed length IE followed by a variable length IE
- o multiple variable-length IEs

Verify the correct reception and decoding of all templates and data.

3.2.3. Flowsets with Padding

Create and send data records which contain padding (i.e. which use the PaddingOctets IE). Test with legal and illegal padding sizes (i.e. padding to boundaries other than 4 or 8 octets). Make sure the implementation captures the (illegal) 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). 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 data element to a 4 byte boundary).

3.3. IE Tests

This section lists the tests which cover the use of IEs and the correct transfer of IPFIX Options Templates.

3.3.1. Enterprise-specific IEs

Export a template and data set which makes use of Enterprise-specific IEs as defined in [[I-D.ietf-ipfix-info](#)] and check correct reception and decoding. Verify correct reception of IEs which are unknown to the collector. Ensure that such IEs are not silently discarded.

3.3.2. Reduced-size Encoding of IEs

Generate export and test reception of IEs which have been transmitted using a reduced-size encoding as defined in [section 6.2](#) of [[I-D.ietf-ipfix-protocol](#)]. Make sure that the collector is aware of the real size of each IE and not only the length used for its transmission.

3.3.3. Multiple use of the same IE in one Template

Create and export a data template containing multiple instances of the same IE, either consecutively or with other IEs in between. Verify that the collector is able to parse the message contents and stores all values received for all the IEs which appeared multiple times in the template definition.

3.4. Options Templates

This section lists the tests which cover the use of IEs and the correct transfer of IPFIX Options Templates.

3.4.1. Using any IEs as Scope

Options Templates contain a scope field which gives the context of the reported IEs in the corresponding Data Records. The scope is an IE specified in [[I-D.ietf-ipfix-info](#)].

Export Options Template Records containing various different IEs in their scope fields, and export a data record using each template. Verify the correct reception of the templates and data records at the collector. Verify whether the collector accepts an unknown IE in the scope field. Verify whether the collector accepts an Enterprise specific IE in the scope field.

The Scope Field Count MAY NOT be zero. Verify that the collector does not accept an Options Template with no scope fields.

3.4.2. Using multiple Scopes

Multiple scope fields MAY be present in the Options Template Record. If the order of the scope fields is relevant, the order of the scope fields MUST be used.

Export an Options Template Record containing multiple scope fields, and a data record using that template. Verify the correct reception of the template and data record at the collector.

Note that the Scope Field Count MAY NOT be zero. Verify that the collector does not accept an Options Template with no scope fields.

3.4.3. Metering Process (MP) Statistics Option Template

Check that the IPFIX collector can handle the reception and decoding of options template records in general and that it is able to receive and decode MP statistic option data records as defined in section 4.1 of [[I-D.ietf-ipfix-protocol](#)]. Note that not all fields listed there might be present in a received MP statistic option data record. Also check that the optional additional Scope Field is supported by the implementation.

3.4.4. Metering Process (MP) Reliability Statistics Option Template

Check that the IPFIX collector can handle the reception and decoding of MP reliability option data records as defined in [section 4.2](#) of

[[I-D.ietf-ipfix-protocol](#)]. Note that not all fields listed there might be present in a received MP reliability option data record. Also check that the optional additional Scope Field is supported by the implementation.

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

Check that the IPFIX collector can handle the reception and decoding of EP reliability option data records as defined in section 4.3 of [[I-D.ietf-ipfix-protocol](#)]. Note that not all fields listed there might be present in a received EP reliability option data record.

[3.4.6.](#) Flow Keys Option Template

Check that the IPFIX collector can handle the reception and decoding of flow key option template data records as defined in section 4.4 of [[I-D.ietf-ipfix-protocol](#)]. Note that not all fields listed there might be present in a received EP reliability option data record. Make sure that the implementation also properly handles the case where the transmitted templateId incorrectly refers to a non-existing template.

[3.4.7.](#) Template Withdrawal Message

Send a template withdrawal message for (a) a template which had been sent before, (b) for a template which has never been sent, and (c) for a template which was previously sent and already withdrawn. Check correct behavior of the collector when receiving data records before and after the template withdrawal. IPFIX template management is defined in chapter 8 of [[I-D.ietf-ipfix-protocol](#)].

[3.5.](#) Stress/Load Tests

Stress tests are used to check correct behavior and robustness of an IPFIX collector 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 collector must not block the IPFIX exporter(s) from sending, since UDP is not congestion aware. Such stress tests may not be applicable to the devices being tested. The tests may be dependant 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 SHOULD verify that his implementation is stable under excessive traffic conditions, for whatever definition of "excessive" applies at their intended installation.

The implementer MUST verify the correct operation of his exporter and/or collector when the collector is incapable of processing

records at the rate which they are received.

3.5.1. Large Number of Records for one Template

Export many records to the collecting process. Depending on what that process does (save to file, store to database, analyze the data) the collector may use up a lot of memory. Verify that if it runs out of memory, it terminates the connection gracefully but remains available to receive data exported on other connections.

3.5.2. High Rate of incoming Data Records

If possible, export to the collector with an increasing records per second export rate. For TCP or SCTP export this should stall the exporter once the collector becomes fully loaded. For UDP export, the collector should drop records gracefully as it becomes overloaded.

3.5.3. Large Templates with high Number of IEs

Create and export templates with the maximum possible number of IEs. Create and export matching data records. Note that, for the implementation, these data records might be smaller or larger than the template records depending on the type of IEs inside and the presence of variable-length IEs.

3.5.4. Many new Templates within Data Template timeout interval

Create and export a large number of data templates using different template IDs, to stress test the collector's memory consumption. Ensure that the collector gracefully discards data templates (i.e. logs warnings) if it's running in a system with insufficient memory resources.

3.5.5. Multiple Exporters sending to one Collector

Setup multiple exporting processes to export data templates and data to the same collecting process at the same time. Observe correct reception and decoding of all the information at the collector. Check that no exporter stalls or disconnects completely.

3.5.6. Export from one Exporter to multiple Collectors

If possible, configure the exporter to export data records in parallel to different IPFIX collectors. Use simple and complex templates and/or a mixture of them and check for correct reception.

3.6. Error Handling

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

3.6.1. Temporary Network Disconnect

Due to network failures (either physical or logical, e.g. defective routing) the connectivity between an IPFIX exporter and collector might be disrupted. The IPFIX system **MUST** be able to handle such events in a deterministic and graceful way if they should occur during an IPFIX export. When connection oriented transmission protocols (TCP/SCTP) are in use, such a failure may or may not be signaled to the exporter and collector by the operating system depending on the type of network adapter, driver software and operating system in use. The effect might be the direct signaling of an error when IP packet read/write system functions are invoked (signaling connection reset by peer) or there might be an OS-dependant connection timeout. An implementer should check the behavior of his/her IPFIX system upon such interruptions of data transmission. For TCP- and SCTP-based connections, short disconnects and long disconnects should be tested. For UDP-based data export there is no noticeable connection loss, but data received with non-consecutive sequence numbers indicates data loss and should be recognized and reported by the collector per section 3.1 of [[I-D.ietf-ipfix-protocol](#)].

3.6.2. Exporter Termination and Restart during Data Transmission

An IPFIX collecting process might be confronted with a faulty exporter implementation which suddenly crashes, dropping any open connections. The exporter may be restarted again soon after the crash. Kill a running and exporting exporter process. Check that the associated collector gracefully closes all connections associated to that exporter. Start the exporting process again. The collector must be able to correctly receive from the new exporter instance at the same source host.

3.6.3. Collector Termination and Restart during Data Transmission

An IPFIX exporting process might be confronted with a faulty collector implementation which suddenly crashes, dropping any open connections. That collector may be restarted again soon after the crash. Kill a running collector while collecting. Check that the exporter gracefully closes all connections associated with that

collector. Restart the collector. Check that the exporter is able to export correctly to the new collector instance.

3.6.4. Incorrect Template Records

IPFIX Options Templates contain an overall Field Count and a Scope Field Count. The Field Count is the number of all fields in the Option Template Record, including the Scope Fields. The Scope Field Count MAY NOT be zero.

Verify the collector's operation when it receives an options template where the Field Count is less than the Scope Field Count.

Verify the collector's operation when it receives an options template where the Scope Field Count is zero.

3.6.5. Export of defective IPFIX Data record

Check that the collector successfully drops all those data records which are not correct IPFIX messages. Potential errors include but are not limited to:

- o IPFIX message too short
- o illegal use of reduced size encoding
- o invalid length specification in case of variable length IEs

3.6.6. Export of non-matching Template and Data

Check that the collector successfully drops all those data records which do not match with their corresponding template. Potential errors include but are not limited to:

- o too few IEs in data record
- o too many IEs in data record

3.6.7. Incorrect Set IDs

Check that Template Sets, Options Template Sets, and Data Sets with an incorrect Set ID are discarded by the IPFIX collector. As of [[I-D.ietf-ipfix-protocol](#)] version 19 only the Set ID values 2 and 3 denote valid sets.

3.6.8. Flowsets with Invalid Padding

Check that the IPFIX collector gracefully handles flowsets which have

invalid padding, i.e. when the number of padding bytes is incorrect, or when the padding is not composed of NUL character(s). The collector MAY accept the data records only for the latter case.

3.6.9. Re-using the same Template ID inside the Template Expiry Time

Check how the collector handles the case where a template definition is received via UDP export with a template ID which is still in use, i.e. not yet timed out. If the template is the same as the previous this is a valid behavior. Sending a different template with the same ID within the template expiry time however is not allowed and should be reported by the collector.

3.6.10. Re-using the same Template ID after the Template Expiry Time

Check that the collector successfully handles the case where a template definition is received via UDP with a template ID that was in use but has expired.

Also check and ensure that the collector drops data records which refer to a template after its expiry (or withdrawal in the case of SCTP).

3.6.11. Re-sending an existing template ID without withdrawal

[I-D.ietf-ipfix-protocol] states in [section 8](#) that a template MUST NOT be sent more than once during the lifetime of an SCTP association. Create and export a template multiple times using SCTP based data transmission. Ensure that the collector gracefully discards any but the first template record. The collector should log a warning about such error observed from an exporter, and MUST shut down the SCTP association (if any).

4. Security Considerations

This memo raises no security issues.

5. References

[I-D.ietf-ipfix-architecture]

Sadasivan, G., "Architecture for IP Flow Information Export", [draft-ietf-ipfix-architecture-09](#) (work in progress), August 2005.

[I-D.ietf-ipfix-info]

Quittek, J., "Information Model for IP Flow Information Export", [draft-ietf-ipfix-info-11](#) (work in progress), September 2005.

[I-D.ietf-ipfix-protocol]

Claise, B., "IPFIX Protocol Specification", [draft-ietf-ipfix-protocol-19](#) (work in progress), September 2005.

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

Authors' Addresses

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

Phone: +49 30 3463 7136
Email: 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>

Intellectual Property Statement

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.

Disclaimer of Validity

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 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.

Copyright Statement

Copyright (C) The Internet Society (2006). 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.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

