Network Working Group   C. Schmoll
Request for Comments: 5471   Fraunhofer FOKUS
Category: Informational   P. Aitken
                                                            B. Claise
                                                  Cisco Systems, Inc.
                                                          March 2009

           Guidelines for IP Flow Information Export (IPFIX) Testing

Status of This Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

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.

Table of Contents

1. Introduction

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

   For a testable IPFIX software tool kit, 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 third-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.

   The tests can be executed in a testbed environment or on a live
   network.

   However, care should be taken regarding the "stress/load test" and
   the "temporary network disconnect", as they might impact other
   systems in the network. We recommend that these specific tests
   should be executed only in a testbed environment.

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
   other. 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 on, e.g., 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 [RFC5470], per the requirements
   specified in [RFC3917].

   [RFC5470] specifies how IPFIX Data Records and Templates are carried
   via a congestion-aware transport protocol from IPFIX Exporting
   Processes to IPFIX Collecting Processes. IPFIX has a formal
   description of IPFIX Information Elements, their name, type, and
   additional semantic information, as specified in [RFC5102]. Finally,
   [RFC5472] describes what type of applications can use the IPFIX
   protocol and how they can use the information provided. It
   furthermore shows how the IPFIX framework relates to other
   architectures and frameworks.

2. Terminology

   IPFIX-specific terminology used in this document is defined in
   Section 2 of [RFC5101]. In this document, as in [RFC5101], the first
   letter of each IPFIX-specific term is capitalized.

2.1. 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].

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 the 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 a Stream Control Transmission Protocol (SCTP) transport, the
   tester must ensure that an SCTP association is established.

   When the Exporting Process and Collecting Process are to be connected
   by a 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 online with an appropriate
   packet sniffing tool. Such a 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.

```
+--------------------------------------------------+
| Set Header #1                                    |
+--------------------------------------------------+
| Data Record                                      |
+--------------------------------------------------+
| Data Record                                      |
+--------------------------------------------------+
   ...
+--------------------------------------------------+
| Data Record                                      |
+--------------------------------------------------+
| Padding with 0 valued octets                     |
+--------------------------------------------------+
| Set Header #2                                    |
+--------------------------------------------------+
| Data Record                                      |
+--------------------------------------------------+
```
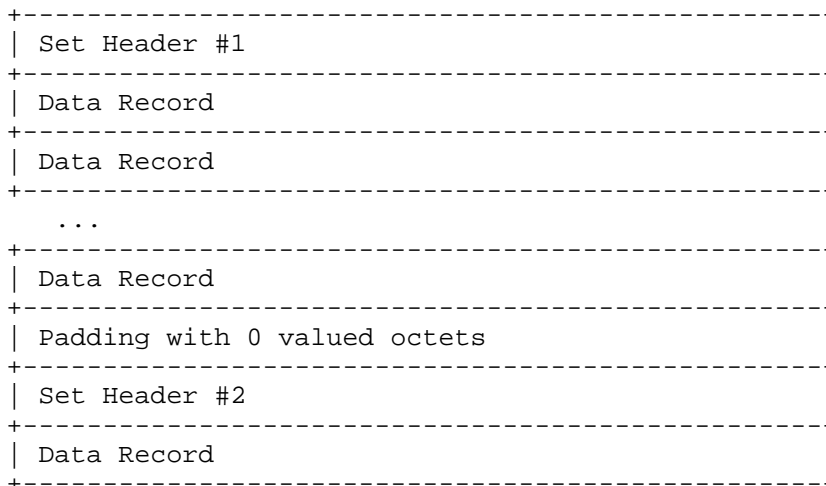
                             Figure 1

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
   cases 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 byte 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.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Set ID = 2           |        Length = 16 octets     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Template ID 256        |        Field Count = 2        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|  ipDiffServCodePoint = 195  |        Field Length = 1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|    paddingOctets = 210      |        Field Length = 3       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                               Figure 2

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Set ID = 256         |          Length = 64          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        1        |                    0                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
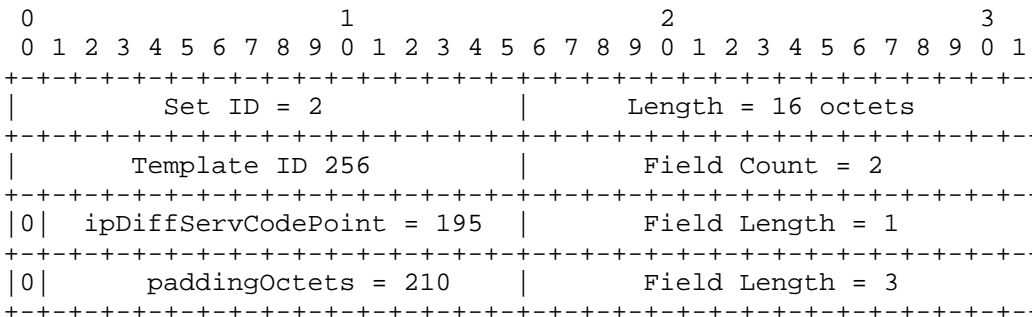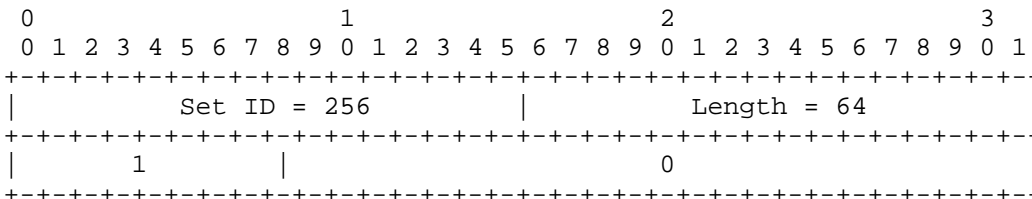
                               Figure 3

   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
   Section 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 that 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 it
   must ensure 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 that it 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 that it 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 Element 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 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 that 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 Template,
   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 it
   must ensure 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 because UDP is not
   congestion aware.

   The tests may be dependent upon the hardware and transport 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 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 the
   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 logged 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 in one application-level datagram. This limits
the number of Information Elements one can specify in an IPFIX
Template when using UDP export. SCTP and TCP are streaming
protocols, so they do not impose much restriction on the packet
level. UDP requires 20 octets for a minimal IPv4 header, 8 octets
for the UDP 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 - 8 - 16 - 4 - 4) = 65483 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 65483 / 4 = 16370. With this many
Information Elements, the Template will be 65480 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 the 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 it
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.

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 that 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 it 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 it 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 it 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.

```
 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 = 32      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Export Time = 1155202151                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Sequence Number = 0x12345678                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Observation Domain ID = 0x33334444             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Set ID = 2           |        Set Length = 12        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Template ID = 257      |        Field Count = 2        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0| Info Element Identifier = 8 |      Field Length = 4         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
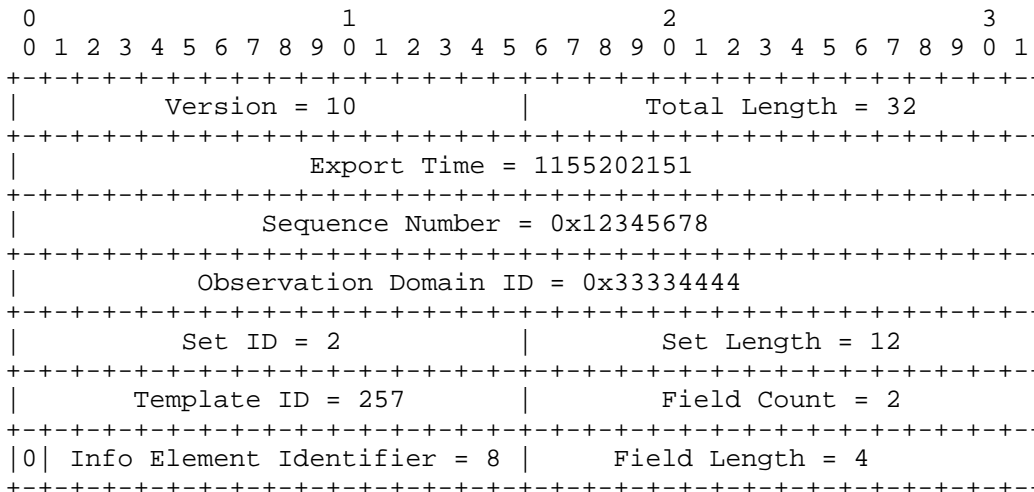
Figure 4

The tester must create and cause the Exporting Process to export the following IPFIX Templates, and must ensure the correct Collecting Process behavior for each of the transports and protocols in Section 3.1.

The tester must do the following:

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

o For TCP transport, ensure that the Collecting Process discards the IPFIX Message, closes the TCP connection, and 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 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 behavior 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 the 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 behavior 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 behavior 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 the 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 behavior 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      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
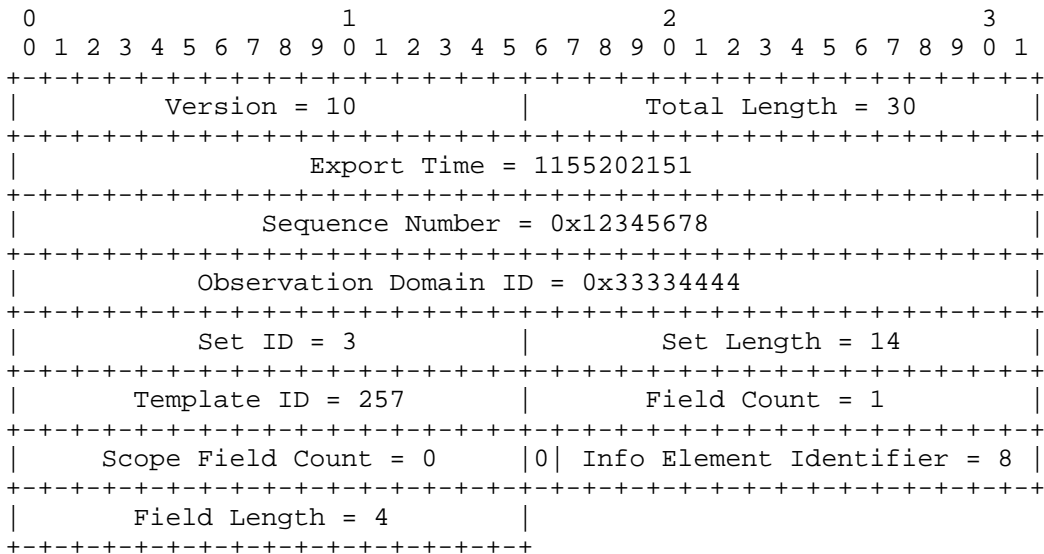
                              Figure 6

   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. Unknown Set IDs

   The tester must create Template Sets, Option Template Sets, and
   associated Data Sets using Set IDs which are unknown to the
   Collecting Process, and cause these to be exported to the Collecting
   Process over all applicable combinations of transports and protocols
   in Section 3.1.

   Per Section 3.3.2 of [RFC5101], only the Set ID values 2 and 3 denote
   valid Sets.

   The tester must ensure that the Collecting Process ignores the
   unknown Sets, logs a warning, and processes the remainder of the
   IPFIX Message.

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
   Datagram Transport Layer Security (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) DNS records
   for each host in the test on a DNS server used by the hosts for name
   resolution. Note that there is no need to configure reverse (PTR)
   DNS records for the hosts, as no part of the protocol uses reverse
   lookups.

   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 the Partially Reliable-Stream Control
   Transmission Protocol (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 Sections 3.7.2,
   3.7.3, and 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 Sections 3.7.2,
   3.7.3, and 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 Sections 3.7.2,
   3.7.3, and 3.7.4.

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

4. Security Considerations

   This memo raises no security issues.

5. Acknowledgments

   The authors wish to thank Brian Trammell for contributing the initial
   text for Section 3.7.

6. Normative References

   [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., Ed., "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.

   [RFC5470]   Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek,
               "Architecture for IP Flow Information Export", RFC 5470,
               March 2009.

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

Authors' Addresses

   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, Inc.
   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, Inc.
   De Kleetlaan 6a b1
   1831 Diegem
   Belgium

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