IPFIX Interoperability Test Event (28th-30th July 2005, Paris, France)

Specification for IPFIX Test Scenarios

Scope of IPFIX Tests Specification

version: 10, 21.07.2005

Editor: Carsten Schmoll (Carsten.schmoll@fokus.fraunhofer.de)
Contributors: Lutz Mark, Antal Bulanza, Benoit Claise, Falko Dressler, Paul Aitken, Carsten Schmoll, Jeroen Massar, Thomas Dietz

IPFIX Overview

- **IP packet flow** information is often gathered and exported
- **from** IP devices such as routers or measurement stations
- **to** mediation, accounting, and network management systems
- **including** (a) those attributes derived from the IP packet headers and (b) attributes known only to the exporter (e.g. ingress and egress ports, network prefix)
- **for** the purposes of Internet research, measurement, attack and intrusion detection, accounting, and billing.

IPFIX Goals and Targets

- standardizing current practice in flow data export
- for this: develop and standardize a generalized flow export methodology
- define common notion of an “IP flow” and specify a data model and a transport mapping for flow and exporter attributes
- having a standardized IPFIX which is nearly compatible to Cisco NetFlow v9
- select a (congestion-aware) transport protocol by which IP flow information can be transferred in a timely fashion

IPFIX Related Material

RFCs

- RFC 3917: Requirements for IP Flow Information Export
- RFC 3955: Evaluation of Candidate Protocols for IP Flow Information Export (IPFIX)

Drafts

- draft-ietf-ipfix-architecture-08.txt - Architecture for IP Flow Information Export
- draft-ietf-ipfix-info-08.txt - Information Model for IP Flow Information Export
- draft-ietf-ipfix-protocol-16.txt - IPFIX Protocol Specification
- draft-ietf-ipfix-as-06.txt - IPFIX Applicability
- draft-ietf-dressler-ipfix-aggregation-01.txt - IPFIX Aggregation (soon available)

Target of IPFIX Testing Event

The main goal of these tests among participants bringing their own implementations of IPFIX exporter and collector software is to check and potentially prove interoperability among different exporter and collector implementations. Implementations should be based on the RFCs and drafts mentioned above.

If an implementation does not support the latest draft specification or lacks some features (e.g. support for transport protocol) or even requirements, this shall be stated clearly before the tests to adopt or maybe cancel affected tests.

The on-site testing during the MOME Interoperability event will be based on predefined scenarios with fixed specification of execution parameters plus ad-hoc tests among attendees from different organisations.

Test Environment

The IPFIX protocol defines data mapping, templates for data record definition and a transport mapping to export flow information records across the network from an IPFIX exporter to an IPFIX collector.

In different test scenarios we will send predefined sets of data records using different data templates (simple and complex) from one machine to a different machine connected to the same network segment.

Test Setup

The collecting process shall be started anew before a test is executed to eliminate potential side-effects due to state a collector might be in from the previous test. Only in some selected test cases (in the final round) a group of different data sets shall be collected in one run.

A new exporter shall be started for each test to eliminate side-effects due to previous tests.

After each test the results will be compared with the expected results (behaviour and data), and they are documented into prepared IPFIX test results forms.

Test Scenarios

IPFIX base-protocol Tests (template-independent)

The following tests can be performed using an IPFIX exporting process on one host and an IPFIX collecting process on a different host. In our tests exporter and collector will originate from different companies.

connectivity tests

- support of ipv4 and ipv6 network layer protocol
- transport layer connectivity for tcp, udp and sctp
- different combination of exporter and collector implementations

error case tests

- temporary network disconnect (probably not useful for UDP)
- exporter restart during data transmission (simulates software crash + restart)
- collector restart during data transmission (simulates software crash + restart)

IPFIX application-specific Tests (template-dependent)

- check correct transmission of control information (template definition records)
- export of different templates and data (single and multiple elements in one template)
- test transmission of all IPFIX data types
- test transmission of combination of data types (multiple elements)
- test big templates with huge number of elements (memory stress test))
- test big number of records for one template
- test malicious (defective) template and/or data records
- check correct transmission of standard option templates
  (TBD: check if these are mandatory for a conformant implementation)
- check specific well-defined templates, e.g. accounting records
- Test protocol extensions like IPFIX aggregation (probably not test this in detail this time because feature is not yet supported; can check robustness of collector though)
- stress test with multiple exporters active in parallel sending to one collector

The detailed list of tests is specified in a separate document maintained by all test partners (currently: Cisco, IBM Zurich, Fraunhofer Fokus, NEC, University of Erlangen, and University of Tübingen).

Equipment / Software provided by IPFIX Test Attendees

MOME (not IPFIX specific)

- provides infrastructure Ethernet via hubs, plus WLAN, Intranet, FTP server
- unfortunately **no** Internet access
- RFCs and drafts can be stored on-site on local ftp/http server

CISCO

- router for exporting IPFIX/UDP
- collector receiving IPFIX/UDP
- these can also export and collect v9/UDP if needs be

FOKUS

- laptop with installation of our FOKUS' libipfix software library
- software tools using the libipfix for sending and collecting IPFIX data
- export to file or mysql database (probably use file export - output is quicker to check)
- UDP/TCP/SCTP supported
- will bring gcc, gdb/ddd, ethereal (for fixes, just in case that some tests should fail ☺)

IBM

- Laptop with AURORA and HESPERA (http://www.zurich.ibm.com/sys/storage/resource.html)
- AURORA = NetFlow v5/v7/v9/IPFIX over IPv4+IPv6 UDP/SCTP/TCP Collector
- HESPERA = NetFlow v5/v7/v9/IPFIX over IPv4+IPv6 UDP/SCTP/TCP Meter

NEC

- BareBone PC with Linux
- Collector with IPFIX over IPv4/v6 with UDP/SCTP/TCP support
- Complete development Enviroment
- Windows Laptop if needed for presentations etc.

University of Erlangen / University of Tübingen

- PC with installed Vermont (vermont.berlios.de)
- IPFIX/UDP and PSAMP/UDP probe
- probe supports IPFIX aggregation
- Concentrator supports Netflow.v9/UDP, IPFIX/UDP and PSAMP/UDP
- IPFIX exporter and collector library

Additional Notes and Questions

We are only testing interoperability in the MOME Interoperability testing event, right? This would mean that protocol conformance cannot be tested without a fully standardised protocol and an accepted standard conformant implementation. *Carsten*: **True.**

Potentially some more topics to test?

- Variable length information element
- Flow keys option template
- multiple use of one field identifier inside one template (successive or with others in between)
- incorrect set ID's (2 and 3 are valid)
- using any IE's as scope.
- multiple scopes.
- flowsets with and without padding, and with illegal padding.
- paddingOneOctet (IE #210)
    - correctly used, and
    - badly extended (length > 1).
- Enterprise-specific IE's.
- reduced size encoding of IE's.
- variable length IE's.
- template withdrawal message.
- new options: MP stats, MP reliability, EP reliability.
- flow keys option.
- re-using the same template ID inside the template expiry time (without withdrawing the template) for the same or for different data.
- re-using the same template ID after the template expiry time without withdrawing the template.