TOC

Network Working Group	C. Sommer
Internet-Draft	F. Dressler
Intended status: Standards Track	Univ. Erlangen
Expires: January 15, 2009	G. Muenz
	Univ. Tuebingen
	July 14, 2008

Mediator-Specific Extensions to IPFIX Protocol and Information Model <draft-sommer-ipfix-mediator-ext-01.txt>

Status of this Memo

Abstract

IPFIX supports the concept of a Mediator, a device that receives, transforms, and exports data streams using IPFIX. One of the most important requirements is the reduction of the volume of IPFIX traffic by aggregating and discarding received information. This document introduces a number of extensions to the IPFIX Information Model that support the export of aggregated IPFIX data, introducing abstract data types and Information Elements that optimize the transport of descriptive information in terms of flow records' amount and size. All extensions are directly applicable to the IPFIX Mediator but can be used in many different applications as well.

Table of Contents

1. Introduction

TOC

The IPFIX Mediator is intended to provide techniques and features to process IPFIX data in a Mediation Process. This process receives data streams using IPFIX. It can apply transformations or aggregation techniques and forward the resulting Flow information to an Exporting Process and, thus, to another IPFIX collector. Flow aggregation is one of the key operations in high-bandwidth networks. The main idea is to reduce both the number and the size of IPFIX messages.

This document introduces extensions to the IPFIX Information Model that support the export of aggregated IPFIX data. These extensions allow and optimize the transport of descriptive information on aggregated IPFIX data. Thus, more information can be preserved in the transmission while further reducing both the number and the size of IPFIX messages. All the proposed extensions are directly applicable to the IPFIX Mediator but can be used in many different applications as well.

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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.). Illustrations of abstract data types are written in Augmented Backus-Naur Form (ABNF), as specified in [RFC4234] (Crocker, D., Ed. and P. Overell, “Augmented BNF for Syntax Specifications: ABNF,” October 2005.), extending the abstract data types defined in [RFC5102] (Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, “Information Model for IP Flow Information Export,” January 2008.). Apart from the basic terms as defined in [RFC5101] (Claise, B., “Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information,” January 2008.), this document uses terminology first introduced in

[I-D.dressler-ipfix-aggregation] (Dressler, F., Sommer, C., Muenz, G., and A. Kobayashi, “IPFIX Flow Aggregation,” July 2008.).

2. Abstract data type orderedList

TOC

IPFIX allows the transport of an ordered list of values by including in a Template several Information Elements of the same type more than once. This approach requires one Template for each possible length of the list. In the context of flow mediation, however, the number of entries in such lists typically changes with each exported compound flow, leading to a dramatic increase of Templates and associated housekeeping overhead. Therefore, a new abstract data type, orderedList, is defined in this section.
The abstract data type orderedList defines an ordered list of Information Elements, each being of the same type (referred to as elementType) and the same, pre-defined length. An orderedList can transport any finite number of Information Elements. The length of an orderedList thus varies and is an integer multiple of the contained Information Elements' length. If more than one contained Information Element is transmitted in the form of an orderedList, reduced size encoding of elementType MUST NOT be used. If only one contained Information Element is transmitted, reduced size encoding of elementType MAY be used. In ABNF-style notation, the syntax can be summed up as follows:

```
orderedList     =  *( elementType )
```

The number of Information Elements contained in an orderedList can be determined by dividing the length of the orderedList by the length of elementType. An Information Element basing on orderedList MAY also be used as a variable-length Information Element by prefixing it with a one-octet or three-octet length specifier as defined in [RFC5101] (Claise, B., “Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information,” January 2008.).
Table 1 (orderedList Examples) shows some encoding examples if unsigned16 is used as the elementType.

Human-Readable	Octets	Hexadecimal	Remarks
80	1	50	Reduced size encoding
80	2	0050	
80, 443	4	0050 01BB	

80, 443, 8080	6	0050 01BB 1F90	

Table 1: orderedList Examples

3. Abstract data type orderedPair

TOC

The abstract data type orderedPair defines a 2-tuple of Information Elements, each being of the same type (referred to as elementType) and the same, pre-defined length. The length of an orderedPair is thus defined as twice the length of its elementType. If more than one contained Information Element is transmitted in the form of an orderedPair, reduced size encoding of elementType MUST NOT be used. If only one contained Information Element is transmitted, reduced size encoding of orderedPair MAY be used if both contained Information Elements are of the same value. The reduced size representation of the orderedPair is in this case identical with the (full or reduced size representation) of elementType. In ABNF-style notation, the syntax can be summed up as follows:

```
orderedPair     =  elementType elementType
orderedPair     =/ elementType
```

Table 2 (orderedPair Examples) shows some encoding examples if unsigned16 is used as the elementType.

Human-Readable	Octets	Hexadecimal	Remarks
80, 80	1	50	Reduced size encoding
80, 80	2	0050	Reduced size encoding
80, 80	4	0050 0050	
80, 443	4	0050 01BB	

Table 2: orderedPair Examples

TOC

4. Abstract data type portRanges

For some applications it might be useful to restrict the applicability of an Aggregation Rule to Flows with source or destination port being of a specific set of port numbers. In an Aggregation Rule, such a set of port numbers can be specified as a pattern. However, the current IPFIX Information Model does not define any data type that allows transmitting a set of port numbers, which is necessary in order to export the pattern as a Common Property of the resulting Compound Flows. Therefore, the new abstract data type portRanges for a list of port ranges is defined in this section.
The abstract data type portRanges is an orderedList of orderedPair Information Elements, each pair consisting of two unsigned16 Information Elements representing the port range's first and last port number.
Data types basing on portRanges MAY thus be cast down to unsigned16 using reduced size encoding to represent a single Port and, hence, the transportSourcePort and transportDestinationPort data types, currently based on the unsigned16 abstract data type, can also be parsed as portRanges-based data types. As specified for data types basing on orderedList, an Information Element basing on portRanges MAY also be used as a variable-length Information Elements by prefixing it with a one-octet or three-octet length specifier as defined in [RFC5101] (Claise, B., “Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information,” January 2008.).
Table 3 (PortRanges Examples) shows some encoding examples with portRanges.

Port Ranges	Octets	Hexadecimal Representation
80	2	0050
1:7	4	0001 0007
80, 443	8	0050 0050 01BB 01BB
1:7, 256:1024	8	0001 0007 0100 0400
20, 80, 443	12	0014 0014 0050 0050 01BB 01BB
1:7, 80, 443	12	0001 0007 0050 0050 01BB 01BB

Table 3: PortRanges Examples

TOC

5. Abstract data type ipv4Network

Currently, the transport of IP network information as specified by IPFIX is done using two separate fields for the network address and the corresponding mask. We propose a new abstract data type ipv4Network that represents the common notation of IP networks: address/mask.
The ipv4Network abstract data type extends the abstract data type ipv4Address to allow a concatenated unsigned8 specifying the prefix length. Alternatively, Information Elements based on the ipv4Network abstract data type MAY be transmitted using reduced size encoding to transmit only the network part of an IPv4 address. In ABNF-style notation, the syntax can be summed up as follows:

```
ipv4Network    =  ipv4Address unsigned8
ipv4Network    =/ *4( unsigned8 )
```

Although using an ipv4Network field instead of two separate fields for prefix and mask will not reduce the length of resulting Flow Records, it eases the work of the aggregator: With ipv4Network, the comparison of subnet addresses requires only one field lookup per Flow Record instead of two. Furthermore, using the abstract data type ipv4Network reduces the Template size by one field equaling four octets.
Applications such as IPFIX Aggregation benefit from ipv4Network if network addresses are frequently exported.

6. excludedPropertiesId Information Element

TOC

The IPFIX Information Model [RFC5102] (Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, “Information Model for IP Flow Information Export,” January 2008.) defines the commonPropertiesId Information Element, which can be used to link to information which several Flows have in common.
Similarly, the excludedPropertiesId shall be defined to link to a set of Common Properties which a Flow does explicitly not exhibit. An Element Id of 239 is proposed for this Information Element.
The excludedPropertiesId works like a Boolean "and not" operation on the linked properties. This means that, if an excludedPropertiesId refers to a set of Common Properties which in turn specifies excluded properties, these transitively referenced properties are to be treated as if directly referenced via a commonPropertiesId element and, hence, as being present in the Flow in question.
Multiple excludedPropertiesId and commonPropertiesId specified for an IPFIX Record must never contradict each other. If an IPFIX Collector is able to detect that contradicting IEs were received, it SHOULD proceed as if it received bad or nonsensical data.
The excludedPropertiesId can, for example, be used when a hierarchy of Aggregation Rules with a "preceding rule" semantic, as introduced in

[I-D.dressler-ipfix-aggregation] (Dressler, F., Sommer, C., Muenz, G., and A. Kobayashi, “IPFIX Flow Aggregation,” July 2008.), is configured in an IPFIX Aggregator.

Figure 1 (Using excludedPropertiesId to communicate a rule hierarchy) illustrates the use of Common Property definitions and the linking to these definitions with Information Elements of types commonPropertiesId (CP) and excludedPropertiesId (EP). In this example, two rules are defined in the aggregator: Rule 1 matches Flows with a sourceIPv4Address of 192.0.2.1, Rule 2 matches Flows with a destinationIPv4Address of 192.0.2.2. Furthermore, Rule 1 is configured to precede Rule 2 in a hierarchy of rules, i.e. Flows that matched Rule 1 will never match Rule 2.

In order to communicate this fact to a receiver, each Aggregation Rule is transmitted as two sets of Common Properties. One set of properties (shown on the right hand side of Figure 1 (Using excludedPropertiesId to communicate a rule hierarchy)) directly transmits a rule's filtering criteria. The other set of properties (shown on the left hand side) refers via a commonPropertiesId to all properties that a Compound Flow exhibits, as well as via an excludedPropertiesId to all that the Compound Flow does not exhibit.

The Flow depicted at the bottom of Figure 1 (Using excludedPropertiesId to communicate a rule hierarchy) thus communicates a source port of 80, a destination port of 65432, a destination IP of 192.0.2.2 and a source IP of "not 192.0.2.1". However, besides the transmission of this Flow in one Data Record, previous transmissions (and the successful reception) of four Option Templates, four Option Data Records and one Template are required to communicate this information.

```
Rule 1:
+########+------+             +######+---------------+
# CP=101 # CP=1 |<----------->#  CP=1 # SRC=192.0.2.1 |
+########+------+             +######+---------------+
                                 ^
            .--------------------'
Rule 2:     v
+########+------+------+      +######+---------------+
# CP=102 # EP=1 | CP=2 |<---->#  CP=2 # DST=192.0.2.2 |
+########+------+------+      +######+---------------+
     ^
     '-------------------.
Flow:                    v
+--------+-----------+--------+
| SPT=80 | DPT=65432 | CP=102 |
+--------+-----------+--------+
```

Figure 1: Using excludedPropertiesId to communicate a rule hierarchy

7. Security considerations

TOC

As all methods described in this document are merely variations on methods already introduced in [RFC5101] (Claise, B., “Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information,” January 2008.), the same rules regarding exchange of Flow information apply.

8. IANA Considerations

TOC

Use of excludedPropertiesId, as well as use of a data type basing on ipv4Network or on portRanges requires one new IPFIX Information Element identifier each to be assigned.

9. Normative References

TOC

[I-D.dressler-ipfix-aggregation]	Dressler, F., Sommer, C., Muenz, G., and A. Kobayashi, “IPFIX Flow Aggregation,” draft-dressler-ipfix-aggregation-05 (work in progress), July 2008 (TXT).
[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC4234]	Crocker, D., Ed. and P. Overell, “Augmented BNF for Syntax Specifications: ABNF,” RFC 4234, October 2005 (TXT, HTML, XML).
[RFC5101]	Claise, B., “Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information,” RFC 5101, January 2008 (TXT).
[RFC5102]	Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, “Information Model for IP Flow Information Export,” RFC 5102, January 2008 (TXT).

Authors' Addresses

TOC

Christoph Sommer
University of Erlangen-Nuremberg
Department of Computer Science 7
Martensstr. 3
Erlangen 91058
Germany
Phone: +49 9131 85-27993
Email: christoph.sommer@informatik.uni-erlangen.de
URI: http://www7.informatik.uni-erlangen.de/~sommer/

Falko Dressler
University of Erlangen-Nuremberg
Department of Computer Science 7
Martensstr. 3
Erlangen 91058
Germany
Phone: +49 9131 85-27914
Email: dressler@informatik.uni-erlangen.de
URI: http://www7.informatik.uni-erlangen.de/

Gerhard Muenz
University of Tuebingen
Computer Networks and Internet
Sand 13
Tuebingen 72076
Germany
Phone: +49 7071 29-70534
Email: muenz@informatik.uni-tuebingen.de
URI: http://net.informatik.uni-tuebingen.de/

Full Copyright Statement

TOC

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.