Abstract

This document defines the use of a structured application information in the service function chaining metadata, and specifies a YANG model for the configuration of the application registry.

The consumers of application information are Service Functions that apply policy and provide application statistics based on the metadata contained in the packet.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on February 16, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents
Internet-Draft                SFC Traceroute                August 2016

carefully, as they describe your rights and restrictions with respect to
this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Open Issues .......................... 2
2. Introduction .......................... 2
  2.1. Terminology ........................ 3
  2.2. Tree Diagrams ..................... 4
3. Application Information Structure .......... 4
4. Application Information Yang Model ........ 5
  4.1. Module Structure .................. 5
  4.2. Application Information Configuration Module .... 6
5. Service Function Chaining Metadata .......... 11
6. Relationship to existing YANG Modules .......... 11
7. Expected Usage ........................ 12
8. IANA Considerations ................... 12
9. Security Considerations ................ 12
10. Acknowledgements .................... 12
11. Changes ............................. 12
12. Informative References ................ 12
Authors’ Addresses ........................ 14

1. Open Issues

1. Relationship of this YANG module and draft-penno-sfc-yang

2. Any reasons why those attributes are not modeled as boolean: P2P-
technology, tunnel-technology, encrypted?

3. The connection between the YANG Model in this document and
[I-D.penno-sfc-yang] must be explained

2. Introduction

As described in the Service Function Architecture [RFC7665], Service
Functions are provide specific treatment for packets and are part of
the end-to-end delivery of services. Many of these network services
include application-specific functions, treatments, and
optimizations.

The SFC Encapsulation, Network Service Header (NSH), therefore needs
to provide with dynamic, flexible, and easily methods to bind service
policy to granular traffic information, which includes application
information. This is achieved by the ability to carry metadata along

the service function path, which is derived from various sources. (e.g., orchestration systems, DPI Classification, etc.) The consumers of this application information are Service Functions that apply policy and provide application statistics based on this metadata contained in the packet.

This document concerns itself with defining structured application information in the service function chaining metadata.

The "Cisco Systems Export of Application Information in IP Flow Information Export (IPFIX) [RFC6759] specifies an extension to the IPFIX information model [RFC7012] to export application information. This IPFIX information element is registered as the identifier 95 in the IPFIX registry [IANA-IPFIX]. Applications could be identified at different OSI layers, from layer 2 to layer 7. For example, the Link Layer Distribution Protocol [LLDP] can be identified in layer 2, ICMP can be identified in layer 3 [IANA-PROTO], HTTP can be identified in layer 4 [IANA-PORTS], and Webex can be identified in layer 7. However, the layer 7 application registry values are out of scope of [RFC6759]

This document purposes the use of IPFIX [RFC7011] application information to be carried in the NSH MD-Type 1 context metadata [I-D.ietf-sfc-nsh]. Optionally, encoding for NSH MD-Type 2 is provided with the Application ID TLV [I-D.quinn-sfc-nsh-tlv]. The information in the metadata will be provided by an orchestration system or the result of packet processing done by a firewall, Intrusion Protection Service (IPS), Deep Packet Inspection (DPI), amongst others. These are defined as providers of application information.

2.1. Terminology

The reader should be familiar with the terms contained in the following documents:

- Section 1.4 of the "Service Function Chaining (SFC) Architecture" [RFC7665]
- Section 2.1 of the "Network Service Header" [I-D.ietf-sfc-nsh]
- The "Generic Protocol Extension for VXLAN" [I-D.ietf-nvo3-vxlan-gpe]
- Sections 3 and 3.1 of "Cisco Systems Export of Application Information in IP Flow Information Export (IPFIX)" [RFC6759]
2.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.
- Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- Abbreviations before data node names: "rw" means configuration (read-write), "ro" state data (read-only), "-x" RPC operations, and "-n" notifications.
- Symbols after data node names: "?" means an optional node, "!" a container with presence, and "*" denotes a "list" or "leaf-list".
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (".").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

3. Application Information Structure

The application information data structure can be seen in Figure 1. It was extracted and adapted from [RFC6759].

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Class. Eng. ID| Zero-valued upper-bits ... Selector ID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Application Identification Data Format

Table 1 displays the currently allocated Classification Engine IDs, including their name and value, as well as their corresponding Selector ID default length.
<table>
<thead>
<tr>
<th>Classification Engine ID Value</th>
<th>Classification Engine ID Name</th>
<th>Selector ID default length (in octets)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IANA-L3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>PANA-L3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>IANA-L4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>PANA-L4</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>USER-Defined</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>PANA-L2</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>PANA-L7</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>ETHERTYPE</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>LLC</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>PANA-L7-PEN</td>
<td>3 (*)</td>
</tr>
</tbody>
</table>

Table 1: Existing Classification Engine IDs

Where:

"PANA = Proprietary Assigned Number Authority". In other words, an enterprise specific version of IANA for internal IDs.

PEN = Private Enterprise Number

(*) There are an extra 4 bytes for the PEN. However, the PEN is not considered part of the Selector ID.

Section 6 of [RFC6759] provides various illustrative examples of the encoding for different applications.

4. Application Information Yang Model

4.1. Module Structure
module: ietf-ipfix-application-information
  +--rw class-id-dictionary
    |  +--rw class-id* [name]
    |       |  +--rw id?      uint8
    |       |  +--rw name     string
    |       |  +--rw description? string
    +--rw application-id-dictionary
      +--rw application-id* [application-name]
        |  +--rw class-id -> /class-id-dictionary/class-id/id
        +--rw pen             uint32
        +--rw selector-id    uint32
        +--rw application-name string
        +--rw application-description? string
        +--rw application-category-name? string
        +--rw application-sub-category-name? string
        +--rw application-group-name? string

4.2. Application Information Configuration Module

<CODE BEGINS> file "ietf-ipfix-application-information@2015-04-28.yang"

module ietf-ipfix-application-information {  
  yang-version 1;

  namespace "urn:ietf:params:xml:ns:yang:"  
   + "ietf-ipfix-application-information";

  prefix ipfix-app-info;

  organization  
   "IETF SFC (Service Function Chaining) Working Group";

  contact  
   "Editor: Christophe Fontaine  
    christophe.fontaine@qosmos.com"

   Editor: Reinaldo Penno  
   rapenno@gmail.com";

  description  
   "This module contains a collection of YANG definitions for  
    the configuration of application ids.

  Copyright (c) 2015 IETF Trust and the persons identified as  
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or  
  without modification, is permitted pursuant to, and subject
typedef application-id-ref {
  type leafref {
    path "/ipfix-app-info:application-id-dictionary/"
    + "ipfix-app-info:application-id/ipfix-app-info"
    + ":application-name";
  }
  description "This type is used by data models that need
to reference an application-id";
}

typedef classification-engine-id {
  type enumeration {
    enum "IANA-L3" {
      value 1;
      description "IANA-L3";
    }
    enum "PANA-L3" {
      value 2;
      description "PANA-L3";
    }
    enum "IANA-L4" {
      value 3;
      description "IANA-L4";
    }
    enum "PANA-L4" {
      value 4;
      description "PANA-L4";
    }
  }
}
enum "USER-Defined" {
    value 6;
    description "USER-Defined";
}
enum "PANA-L2" {
    value 12;
    description "PANA-L2";
}
enum "PANA-L7" {
    value 13;
    description "PANA-L7";
}
enum "ETHERTYPE" {
    value 18;
    description "ETHERTYPE";
}
enum "LLC" {
    value 19;
    description "LLC";
}
enum "PANA-L7-PEN" {
    value 20;
    description "PANA-L7-PEN";
}

description "The definitions for Classification engine ID names.";
reference "RFC 6759: Cisco Systems Export of Application Information in IP Flow Information Export (IPFIX)";

/*
 * Configuration data nodes
*/

container class-id-dictionary {
    description "Dictionary for classification ids";
    list class-id {
        key "name";
        unique "id";
        leaf id {
            type uint8;
        }
    }
}
description "Classification identifier";
}
leaf name {
type string;
description "classification Engine name";
}
leaf description {
type string;
description "Description of the class-id";
}
description "A list of all classification ids";
}
}

container application-id-dictionary {
description "Dictionary for application ids";
list application-id {
key "application-name";
unique "class-id pen selector-id";
leaf class-id {
type leafref {
path "/ipfix-app-info:class-id-dictionary/"
+ "ipfix-app-info:class-id/ipfix-app-info:id";
}
mandatory true;
description "Application Name";
}
leaf pen {
type uint32;
mandatory true;
description "Private Enterprise Number, only relevant when used with appropriate class-id. Set to 0 when not used.";
}
leaf selector-id {
type uint32 {
range "0..16777216";
}
mandatory true;
description "Selector identifier";
}
leaf application-name {
type string;
mandatory true;
description "The name of the application";
}
leaf application-description {
type string;
description "The description of the application";
}
leaf application-category-name {
  type string;
  description "An attribute that provides a first-level categorization for each Application ID. Examples include browsing, email, file-sharing, gaming, instant messaging, voice-and-video, etc. The category attribute is encoded by the application-category-name Information Element";
}
leaf application-sub-category-name {
  type string;
  description "An attribute that provides a second-level categorization for each Application ID. Examples include backup-systems, client-server, database, routing-protocol, etc. The sub-category attribute is encoded by the application-sub-category-name Information Element";
}
leaf application-group-name {
  type string;
  description "An attribute that groups multiple Application IDs that belong to the same networking application. For example, the ftp-group contains ftp-data (port 20), ftp (port 20), ni-ftp (port 47), sftp (port 115), bftp (port 152), ftp-agent (port 574), ftps-data (port 989). The application-group attribute is encoded by the application-group-name Information Element";
}

<CODE ENDS>
5. Service Function Chaining Metadata

When a Deep Packet Inspection (DPI), Firewall or any other Service Function (SF) that can identify applications want to convey this knowledge to other SFs it encoded in the format discussed earlier and add to the context metadata.

As defined in [I-D.ietf-sfc-nsh], there are two formats for the NSH Metadata, or the portion of the NSH header beyond the mandatory Base Header and Service Path Header: MD-Type 1 and MD-Type 2.

The Application Identification data structure (see Figure 1) can be carried both in MD-Type 1 and MD-Type 2. This document specifies the encoding within NSH MD-Type 1 (see Figure 2), and encoding for NSH MD-Type 2 is provided with the Application ID TLV [I-D.quinn-sfc-nsh-tlv].

The Example in Figure 2 shows the encoding of the SNMP application using MD-Type 1.

```
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +--------------------------------------------------+
<p>|       3       |        0      |              161              |
|--------------------------------------------------|</p>
<table>
<thead>
<tr>
<th>Network Shared Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Platform Context</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Service Shared Context</td>
</tr>
</tbody>
</table>
  +--------------------------------------------------+
```

**Figure 2: Example of Metadata Including the SNMP Application Identification**

In this example, the Classification Engine IDs of 3 indicates "IANA-L4", and 161 is the well-known port number for SNMP (with its upper bits zero-valued).

Other Services Functions that need application information associated with a packet or flow can look at this metadata (encoded in either MD-Type 1 or MD-Type 2) and easily find out its value.

6. Relationship to existing YANG Modules

[RFC6728] specifies a data model for the IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) protocols. It is for configuring and monitoring Selection Processes, Caches, Exporting Processes, and
Collecting Processes of IPFIX- and PSAMP-compliant Monitoring Devices using the Network Configuration Protocol (NETCONF). The data model is defined using UML (Unified Modeling Language) class diagrams and formally specified using YANG.

The YANG model in this document allows the configuration of the application id IPFIX information elements (ieId), which in turn, may be used in a template definition (TemplateId).

[I-D.penno-sfc-yang] To be done

7. Expected Usage

Devices or controllers will download the [ETHERTYPE], [IANA-PROTO] and [IANA-PORTS] from the appropriate URIs. However, the configuration of the applications is required for applications not registered in an industry-wide agreed-upon registry. In this case, the Proprietary Assigned Number Authority (PANA) registries (PANA-L2, PANA-L3, PANA-L4, PANA-L7), or the User-Defined registry, must be used to identify new application.

Furthermore, the following attributes are statically assigned per Application ID, and needs to be configured: category, sub-category, application-group.

8. IANA Considerations

TBD

9. Security Considerations

TODO: Update with privacy and security considerations, as requested in Prague IETF93.

10. Acknowledgements

The authors wish to thank Kengo Naito for a thorough review and insightful comments.

11. Changes

12. Informative References

[ETHERTYPE]

"ETHERTYPE", 1984,
Internet-Draft               SFC Traceroute                  August 2016

[I-D.ietf-nvo3-vxlan-gpe]
Kreeger, L. and U. Elzur, "Generic Protocol Extension for
VXLAN", draft-ietf-nvo3-vxlan-gpe-02 (work in progress),
April 2016.

[I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-
ietf-sfc-nsh-05 (work in progress), May 2016.

[I-D.penno-sfc-yang]
Penno, R., Quinn, P., Zhou, D., and J. Li, "Yang Data
Model for Service Function Chaining", draft-penno-sfc-
yang-15 (work in progress), June 2016.

[I-D.quinn-sfc-nsh-tlv]
Quinn, P., Elzur, U., Majee, S., and J. Halpern, "Network
Service Header TLVs", draft-quinn-sfc-nsh-tlv-01 (work in
progress), April 2016.

[IANA-IPFIX]
IANA, "IP Flow Information Export (IPFIX) Entities", 2015,

[IANA-PORTS]
IANA, "IANA-L4", 1984,
<http://www.iana.org/assignments/port-numbers>.

[IANA-PROTO]
IANA, "IANA-L3", 1984,

[LLDP]     IEEE, "Std 802.1AB-2005, "Standard for Local and
metropolitan area networks - Station and Media Access
Control Connectivity Discovery", IEEE Std 802.1AB-2005",
2005, <LLDP>.

[RFC6728]  Muenz, G., Claise, B., and P. Aitken, "Configuration Data
Model for the IP Flow Information Export (IPFIX) and
Packet Sampling (PSAMP) Protocols", RFC 6728,
DOI 10.17487/RFC6728, October 2012,

[RFC6759]  Claise, B., Aitken, P., and N. Ben-Dvora, "Cisco Systems
Export of Application Information in IP Flow Information
Export (IPFIX)", RFC 6759, DOI 10.17487/RFC6759, November


Authors’ Addresses

Reinaldo Penno
Cisco Systems, Inc.
170 West Tasman Dr
San Jose CA
USA

Email: repenno@cisco.com

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

Email: bclaise@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709-4987
USA

Email: cpignata@cisco.com