IPFIX Working Group
Internet-Draft
Intended Status: Informational
Expires: April 16, 2011

B. Claise
P. Aitken
N.ir Ben-Dvora
Cisco Systems, Inc.
October 16, 2010

Export of Application Information in IPFIX
draft-claise-export-application-info-in-ipfix-00

Status of this Memo

Abstract

This document specifies an extension to the IP Flow Information eXport (IPFIX) protocol specification in [RFC5101] and the IPFIX information model specified in [RFC5102] to export application information.

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 RFC 2119 [RFC2119].

Table of Contents

List of Figures

1. Overview

1.1. 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 the IPFIX Architecture [RFC5470], per the requirements defined in RFC 3917 [RFC3917].

The IPFIX Architecture [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 the IPFIX information model [RFC5102].

In order to gain a level of confidence in the IPFIX implementation, probe the conformity and robustness, and allow interoperability, the Guidelines for IPFIX Testing [RFC5471] presents a list of tests for implementers of compliant Exporting Processes and Collecting Processes.

The Bidirectional Flow Export [RFC5103] specifies a method for exporting bidirectional flow (biflow) information using the IP Flow Information Export (IPFIX) protocol, representing each Biflow using a single Flow Record.

The "Reducing Redundancy in IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Reports" [RFC5473] specifies a bandwidth saving method for exporting Flow or packet information, by separating information common to several Flow Records from information specific to an individual Flow Record: common Flow information is exported only once.

2. Introduction

Today service providers and network administrators are looking for visibility into the packet content rather than just the packet header. Some network devices Metering Processes inspect the packet content and identify the applications that are utilizing

the network traffic. Applications in this context are defined as the user processes that exchange packets between them (such as the web applications, peer to peer applications, file transfer, e-mail applications, etc.)

The application identification is based on different kind of methods or even a combination of such methods:

1. L2 protocols (such as ARP, PPP, LLDP)
2. IP protocols (such as ICMP, IGMP, GRE)
3. TCP or UDP ports (such as HTTP, Telnet, FTP)
4. Application headers
5. Packet content signatures
6. Traffic behavior

The exact application identification methods are part of the Metering Process internals that aims to provide an accurate identification with a minimum false identification. This task requires a sophisticated Metering Process since the protocols do not behave in a standard manner.

1. Applications use port obfuscation where the application run on different port than the IANA assigned one. For example a HTTP server might run a TCP port 23 (assigned to telnet in [IANA-PORTS])
2. IANA does not accurately reflect how certain ports are "commonly" used today. Some ports are reserved, but the application either never became prevalent or is not in use today.
3. The signatures become more and more complex

For that reason, such Metering Processes usually detect application based on multiple mechanisms in parallel. Detecting applications based only on port matching might wrongly identify the traffic. Note that this example stresses the need for the engine strength. If the Metering Process is capable of detecting applications more accurately it is considered as stronger and more accurate.

Similarly, a reporting mechanism that uses L4 port based applications only, such as L4:<known port>, would have a similar issues. The reporting system should be capable of reporting the applications classified using all types for mechanisms. In particular applications that does not have any IANA port definition. While a mechanism to export application information should be defined, the L4 port being in use must be exported using the destination port (destinationTransportPort at [IANA-IPFIX]) in the corresponding NetFlow record.

Cisco Systems uses the IPFIX application tag as described in section 4. to export the application information with the IPFIX protocol [RFC5101].

Application could be defined at different OSI layers, from the layer 2 to the layer 7. Examples: Cisco Discovery Protocol is layer 2 application, ICMP is layer 3 application [IANA-PROTO], HTTP is layer 4 application [IANA-PORTS], and skype is layer 7.

While an ideal solution would be an IANA registry for applications above (or inside the payload of) the well known ports [IANA-PORTS], this solution is not always possible as the some applications require well known specifications. Therefore, some reverse engineering is required, as well as a ubiquitous language for application signature. Clearly not realistic.

As this specification focuses on the application information encoding, this document doesn't contain an application registry for non IANA applications. However, a reference to the Cisco assigned numbers can be found at [CISCO].

2.1. Application Information Use Case

There are several use cases on which the application information is used:

1. Network Visibility

   This is one of the main use cases for using the application information. This use case is also called application visibility. Network administrators are using such application visibility to understand the main network consumers, network trends and user behavior.

2. Billing Services

   In some cases, network providers are willing to bill different applications differently. For example, provide different billing for VoIP and Web browsing.

3. Congestion Control

   While the traffic demand is increasing (mainly due to the high usage of peer to peer applications, video applications and web download applications), the providers revenue doesn't grow. Providers are looking at a more efficient way to control and

prioritize the network utilization. An application aware bandwidth control system is used to prioritize the traffic based on the applications, giving the critical applications priority over the non-critical applications.

4. Security Functions

Application knowledge is sometimes used in security functions in order to provide comprehensive functions such as Application based firewall, URL filtering, Parental control, Intrusion detection, etc.

All of the above use cases require exporting of application information to provide the network function itself or to log the network function operation.

3. Terminology

IPFIX-specific terminology used in this document is defined in Section 2 of the IPFIX protocol specification [RFC5101]. As in [RFC5101], these IPFIX-specific terms have the first letter of a word capitalized when used in this document.

3.1. New Terminology

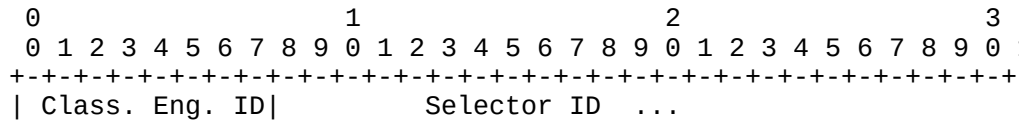
Application Tag

A unique identifier for an application. The Application Tag consists of a Classification Engine ID and a Selector ID [RFC5476].

4. applicationTag Information Element Specification

This document specifies the applicationTag Information Element, which is composed of two parts:

1. 8 bits of Classification Engine ID.
2. m bits of Selector ID. The Selector ID length varies depending on the engine.



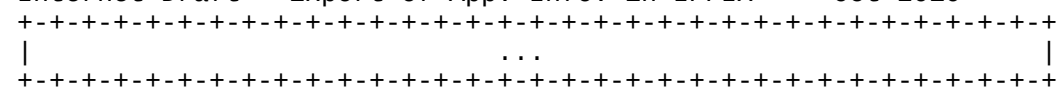


Figure 1: applicationTag Information Element

Classification Engine ID

A unique identifier for the engine which determined the Selector ID. Thus the Classification Engine ID defines the context for the Selector ID.

Selector ID

A unique identifier of the application for a specific Classification Engine ID.

Note that the Selector ID term is in sync with the PSAMP terminology. See [RFC5476], Packet Sampling (PSAMP) Protocol Specifications.

When an application is detected, the most granular application is encoded in the Application Tag: for example, ICMP would be encoded as layer 3 value 1, SNMP as layer 4 value 161, bittorent as layer 7 value 69.

The overall length of the applicationTag Information Element may be specified either in the IPFIX Template Record or by using an IPFIX Variable-Length Information Element. The receiver / decoder must respect this length rather than using the Classification Engine ID to make an assumption about the Selector ID size.

When exporting applicationTag information in IPFIX, the applicationTag SHOULD be encoded in a variable-length Information Element [RFC5101]. However, if a legacy protocol such as NetFlow version 9 is used, and this protocol doesn’t support variable length Information Elements, then either multiple templates (one per applicationTag length), or a single template corresponding to the maximum sized applicationTag MUST be used. This avoids the need for multiple Template Records with different applicationTag lengths when the IPFIX variable length encoding [RFC5101] is not available.

As a consequence, although some Application Tags can be encoded in a smaller number of bytes (eg, an IANA L3 protocol encoding

would take 2 bytes, while an IANA L4 port encoding would take 3 bytes), nothing prevents an Exporting Process from exporting all Application Tags with a larger fixed length.

Note that the Selector ID value is always encoded in the least significant bits as shown:

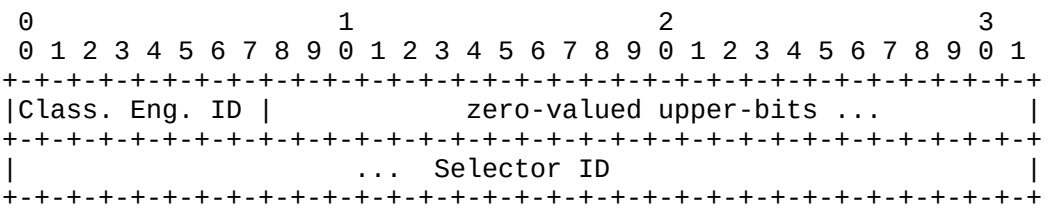


Figure 2: Selector ID encoding

4.1. Existing Classification Engine IDs

The following Engine IDs have been allocated by Cisco Systems.

Name	Value	Description
	0	Invalid.
IANA-L3	1	The IANA protocol (layer 3) number is exported in the Selector ID. See http://www.iana.org/assignments/protocol-numbers.
CANA-L3	2	Cisco Systems proprietary layer 3 definition. Cisco Systems can still export its own layer 3 protocol numbers, while waiting for IANA to assign it. The Selector ID has a global significance for all Cisco Systems devices under CANA governance. Hopefully the same IDs will be maintained after the IANA standardization.
IANA-L4	3	IANA layer 4 well-known port number is exported in the Selector ID. See http://www.iana.org/assignments/port-numbers. Note: as a flow is unidirectional, it contains the destination port in a flow from the client to the server.
CANA-	4	Cisco Systems proprietary layer 4

L4		definition. Cisco Systems can still export its own layer 4 port numbers, while waiting for IANA to assign it. The Selector ID has global significance for all Cisco Systems devices under CANA governance. Hopefully the same ID will be maintained after the IANA standardization. Example: IPFIX had the port 4739 pre-assigned in the IETF draft for years. While waiting for the IANA registration, we could use this Selector ID.
	5	Reserved.
	6	Reserved.
	7	Reserved.
	8	Reserved.
	9	Reserved.
	10	Reserved.
	11	Reserved.
CANA -L2	12	The Selector ID represents the Cisco Systems unique global layer 2 applications. The Selector ID has a global significance.
CANA -L7	13	The Selector ID represents the Cisco Systems unique global ID for the layer 7 applications. The Selector ID has global significance for all Cisco Systems devices.
	14	Reserved.
	15	Reserved.
	16	Reserved.
	17 to 254	Available.
MAX	255	255 is the maximum Engine ID.

Table 1: Existing Classification Engine IDs

Note 1: "CANA = Cisco Systems Assigned Number Authority", Cisco Systems’s version of IANA for internal IDs.

Note 2: This is an extensible list, and new Classification Engine IDs may be allocated at any time. See [CISCO] for the latest version.

4.2. Options Template Record for the Application Name

For engines which specify locally unique Application Tags (which means unique per engine and per router), an Options Template Record (see [RFC5101]) MUST be used to export the correspondence between the Application Tag, the Application Name, and the Application Description. This is called the "options application-table". For engines which specify globally unique Application Tags, an Options Template Record SHOULD be used to export the correspondence between the Application Tag, the Application Name and the Application Description, unless the mapping is hardcoded in the NetFlow Collector, or known out of band (for example, by polling a MIB).

4.3. Resolving IANA L4 port collisions

Even if the IANA L4 ports usually point to the same protocols for both UDP and TCP, there are some exceptions. 10 ports in the first 1K range of ports have different protocols assigned for TCP and UDP:

```
exec            512/tcp    remote process execution;
#                          authentication performed using
#                          passwords and UNIX login names
comsat/biff     512/udp    used by mail system to notify users
#                          of new mail received; currently
#                          receives messages only from
#                          processes on the same machine
login           513/tcp    remote login a la telnet;
#                          automatic authentication performed
#                          based on priviledged port numbers
#                          and distributed data bases which
#                          identify "authentication domains"
who             513/udp    maintains data bases showing who's
#                          logged in to machines on a local
#                          net and the load average of the
#                          machine
shell           514/tcp    cmd
#                          like exec, but automatic
authentication
#                          is performed as for login server
syslog          514/udp
oob-ws-https    664/tcp    DMTF out-of-band secure web services
#                          management protocol
#                          Jim Davis
<jim.davis&wbemsolutions.com>
```

#		June 2007
asf-secure-rmcp	664/udp	ASF Secure Remote Management
#		and Control Protocol
rfile	750/tcp	
kerberos-iv	750/udp	kerberos version iv
submit	773/tcp	
notify	773/udp	
rpasswd	774/tcp	
acmaint_dbd	774/udp	
entomb	775/tcp	
acmaint_transd	775/udp	
busboy	998/tcp	
puparp	998/udp	
garcon	999/tcp	
applix	999/udp	Applix ac

Table 2: IANA layer 4 port collisions

Instead of imposing the protocol (UDP/TCP) in the scope of the "options application-table" Options Template for all applications (on top of having the protocol as key-field in the Flow Record definition), we define that the L4 application is always TCP related, by convention. So, whenever the collector has a conflict in looking up IANA, it would choose the TCP choice. The following UDP L4 applications are assigned in the Cisco L7 Application Tag range (ie, under Classification Engine ID 13):

comsat/biff	256/udp	used by mail system to notify users
who	257/udp	maintains data bases showing who’s
syslog	41/udp	
asf-secure-rmcp	258/udp	ASF Secure Remote Management and
#		Control Protocol
kerberos-iv	259/udp	kerberos version iv
notify	260/udp	
acmaint_dbd	261/udp	
acmaint_transd	262/udp	
puparp	263/udp	
applix	264/udp	Applix ac

Table 3: Resolving layer 4 UDP ports

5. Application Tag Examples

The following examples are created solely for the purpose of illustrating how the extensions proposed in this document are encoded.

5.1. Example 1: Layer 2 Protocol

From the list of Classification Engine IDs in Table 1, we can see that the layer 2 Classification Engine ID is 12:

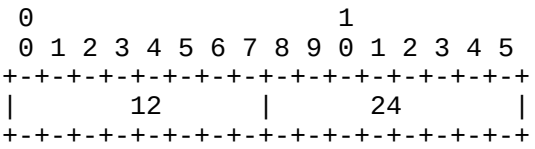
L2	12	The Selector ID represents the layer 2 applications. The Selector ID has a global significance.

From the list of layer 2 protocols at [cisco], we can see that PPP has the value 24:

NAME	Selector ID
ppp	24

So, in the case of layer 2 protocol PPP, the Classification Engine ID is 12 while the Selector ID has the value 24.

Therefore the Application Tag is encoded as:



So the Application Tag has the value of 3097. Instead of representing the Application Tag in hexadecimal format, the format '12...24' is used for simplicity in the examples below.

Flexible NetFlow creates a Template Record with a few Information Elements: amongst other things, the Application Tag. For example:

- sourceIPv4Address (key field)
- destinationIPv4Address (key field)
- ipDiffServCodePoint (key field)
- applicationTag (key field)

- octetTotalCount (non key field)

For example, a Flow Record corresponding to the above Template Record may contain:

{ sourceIPv4Address=1.1.1.1, destinationIPv4Address=2.2.2.2,
ipDiffServCodePoint=0, applicationTag='12...24',
octetTotalCount=123456 }

The Collector has all the required information to determine that the application is PPP, because the Application Tag uses a global and well know registry, ie the IANA protocol number. The 24 value is globally unique within Cisco Systems for Classification Engine ID 12, so the Collector can determine which application is represented by the Application Tag by loading the registry out of band.

5.2. Example 2: Standardized IANA Layer 3 Protocol

From the list of Classification Engine IDs in Table 1, we can see that the IANA layer 3 Classification Engine ID is 1:

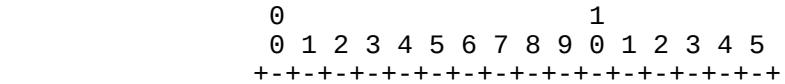
IANA-L3	1	The IANA protocol (layer 3) number is exported in the Selector ID. See http://www.iana.org/assignments/protocol-numbers..

From the list of IANA layer 3 protocols (see [IANA-PROTO]), we can see that ICMP has the value 1:

Decimal	Keyword	Protocol	Reference
1	ICMP	Internet Control Message	[RFC792]

So in the case of the standardized IANA layer 3 protocol ICMP, the Classification Engine ID is 1, and the Selector ID has the value of 1.

Therefore the Application Tag is encoded as:



```
      |       1       |       1       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

So the Application Tag has the value of 257. Instead of representing the Application Tag in hexadecimal format, the format '1...1' is used for simplicity in the examples below.

Flexible NetFlow creates a Template Record with a few Information Elements: amongst other things, the Application Tag. For example:

- sourceIPv4Address (key field)
- destinationIPv4Address (key field)
- ipDiffServCodePoint (key field)
- applicationTag (key field)
- octetTotalCount (non key field)

For example, a Flow Record corresponding to the above Template Record may contain:

```
    { sourceIPv4Address=1.1.1.1, destinationIPv4Address=2.2.2.2,
      ipDiffServCodePoint=0, applicationTag='1...1',
      octetTotalCount=123456 }
```

The Collector has all the required information to determine that the application is ICMP, because the Application Tag uses a global and well know registry, ie the IANA L3 protocol number.

5.3. Example 3: Cisco Systems Proprietary Layer 3 Protocol

Assume that Cisco Systems has specified a new layer 3 protocol called "foo".

From the list of Classification Engine IDs in Table 1, we can see that the Cisco Systems layer 3 Classification Engine ID is 2:

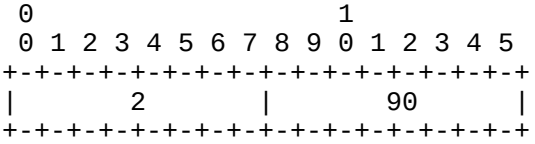
CANA-L3	2	Cisco Systems proprietary layer 3 definition. Cisco Systems can still export its own layer 3 protocol numbers, while waiting for IANA to assign it. The Selector ID has a global significance for all Cisco Systems devices under CANA governance. Hopefully the same IDs will be

A global registry within Cisco Systems specifies that the "foo" protocol has the value 90:

Protocol	Protocol Id
foo	90

So in the case of Cisco Systems layer 3 protocol foo, the Classification Engine ID is 2, and the Selector ID has the value of 90.

Therefore the Application Tag is encoded as:



So the Application Tag has the value of 602. Instead of representing the Application Tag in hexadecimal format, the format ’2..90’ is used for simplicity in the examples below.

Flexible NetFlow creates a Template Record with a few Information Elements: amongst other things, the Application Tag. For example:

- sourceIPv4Address (key field)
- destinationIPv4Address (key field)
- ipDiffServCodePoint (key field)
- applicationTag (key field)
- octetTotalCount (non key field)

For example, a Flow Record corresponding to the above Template Record may contain:

```
{ sourceIPv4Address=1.1.1.1, destinationIPv4Address=2.2.2.2,
  ipDiffServCodePoint=0, applicationTag=’2...90’,
  octetTotalCount=123456 }
```

Along with this Flow Record, a new Options Template Record would be exported, as shown in Section 5.7.

5.4. Example 4: Standardized IANA Layer 4 Port

From the list of Classification Engine IDs in Table 1, we can see that the IANA layer 4 Classification Engine ID is 3:

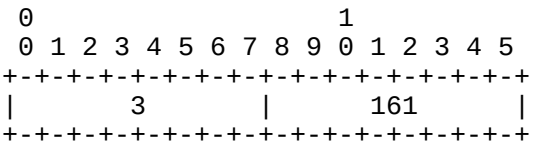
IANA-L4	3	IANA layer 4 well-known port number is exported in the selector ID. See http://www.iana.org/assignments/port-numbers. Note: as a flow is unidirectional, it contains the destination port in a flow from the client to the server.

From the list of IANA layer 4 ports (see [IANA-PORTS]), we can see that SNMP has the value 161:

Keyword	Decimal	Description
snmp	161/tcp	SNMP
snmp	161/udp	SNMP

So in the case of the standardized IANA layer 4 SNMP port, the Classification Engine ID is 3, and the Selector ID has the value of 161.

Therefore the Application Tag is encoded as:



Flexible NetFlow creates a Template Record with a few Information Elements: amongst other things, the Application Tag. For example:

- sourceIPv4Address (key field)
- destinationIPv4Address (key field)
- protocol (key field)
- ipDiffServCodePoint (key field)
- applicationTag (key field)
- octetTotalCount (non key field)

For example, a Flow Record corresponding to the above Template Record may contain:

{ sourceIPv4Address=1.1.1.1, destinationIPv4Address=2.2.2.2,
protocol=17, ipDiffServCodePoint=0,
applicationTag='3..161', octetTotalCount=123456 }

The Collector has all the required information to determine that the application is SNMP, because the Application Tag uses a global and well know registry, ie the IANA L4 protocol number.

5.5. Example 4: Layer 7 Application

In this example, the Metering Process has observes some Citrix traffic.

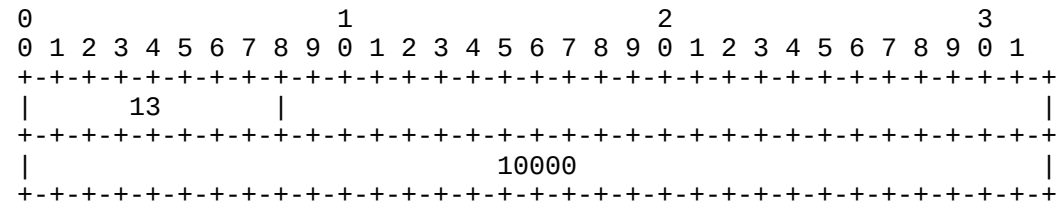
From the list of Classification Engine IDs in Table 1, we can see that the L7 unique Engine ID is 13:

L7	13	The Selector ID represents the Cisco Systems unique global ID for the layer 7 application. The Selector ID has a global significance for all Cisco Systems devices.

Suppose that the Metering Process returns the ID 10000 for Citrix traffic.

So, in the case of this Citrix application, the Classification Engine ID is 13 and the Selector ID has the value of 10000.

Therefore the Application Tag is encoded as:



So the Application Tag has the value of '13..10000'.

Note that the figure shows that the Exporting Process exports the value 10000 in 7 bytes: this is pure speculation. However, it doesn't matter as the applicationTag would be exported in a variable length Information Element.

Flexible NetFlow creates a Template Record with a few Information Elements: amongst other things, the Application Tag. For example:

- sourceIPv4Address (key field)
- destinationIPv4Address (key field)
- ipDiffServCodePoint (key field)
- applicationTag (key field)
- octetTotalCount (non key field)

For example, a Flow Record corresponding to the above Template Record may contain:

```
{ sourceIPv4Address=1.1.1.1, destinationIPv4Address=2.2.2.2,
  ipDiffServCodePoint=0, applicationTag='13...10000',
  octetTotalCount=123456 }
```

The 10000 value is globally unique within Cisco Systems, so the Collector can determine which application is represented by the Application Tag by loading the registry out of band.

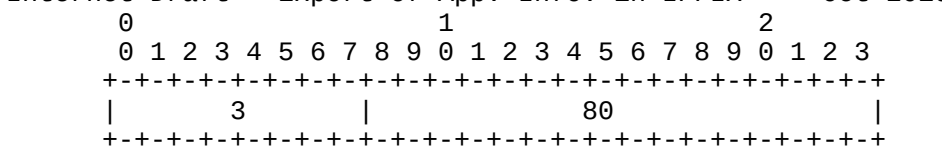
Along with this Flow Record, a new Options Template Record would be exported, as shown in Section 5.7.

5.6. Example: port Obfuscation

For example, a HTTP server might run a TCP port 23 (assigned to telnet in [IANA-PORTS]). If the Metering Process is capable of detecting HTTP in the same case, the Application Tag representation must contain HTTP. However, if the reporting application wants to determine whether or the default HTTP port 80 or 8080 was used, it must export the destination port (destinationTransportPort at [IANA-IPFIX]) in the corresponding NetFlow record.

In the case of a standardized IANA layer 4 port, the Classification Engine ID is 2, and the Selector ID has the value of 80 for HTTP (see [IANA-PORTS]).

Therefore the Application Tag is encoded as:



Flexible NetFlow creates a Template Record with a few Information Elements: amongst other things, the Application Tag. For example:

- sourceIPv4Address (key field)
- destinationIPv4Address (key field)
- protocol (key field)
- destinationTransportPort (key field)
- applicationTag (key field)
- octetTotalCount (non key field)

For example, a Flow Record corresponding to the above Template Record may contain:

{ sourceIPv4Address=1.1.1.1, destinationIPv4Address=2.2.2.2,
protocol=17, destinationTransportPort=23,
applicationTag='3..80', octetTotalCount=123456 }

The Collector has all the required information to determine that the application is HTTP, but runs on port 23.

5.7. Example: Application Mapping Options Template

Along with the Flow Records shown in the above examples, a new Options Template Record would be exported to express the Application Name and Application Description associated with each Application Tag.

The Options Template Record would contain the following Information Elements:

1. Scope = applicationTag.

   From RFC 5101: "The scope, which is only available in the Options Template Set, gives the context of the reported Information Elements in the Data Records."

2. applicationName.

3. applicationDescription.

The Options Data Record associated with the examples above would contain, for example:

```
    { scope=applicationTag='2...90',
      applicationName="foo",
      applicationDescription="The Cisco foo protocol",

      scope=applicationTag='13...10000',
      applicationName="Citrix",
      applicationDescription="A Citrix application" }
```

When combined with the example Flow Records above, these Options Template Records tell the NetFlow collector:

1. A flow of 123456 bytes exists from sourceIPv4Address 1.1.1.1 to destinationIPv4address 2.2.2.2 with a DSCP value of 0 and an applicationTag of '12...90', which maps to the "foo" application.

2. A flow of 123456 bytes exists from sourceIPv4Address 1.1.1.1 to destinationIPv4address 2.2.2.2 with a DSCP value of 0 and an Application Tag of '13...10000', which maps to the "Citrix" application.

6. IANA Considerations

This document specifies three new IPFIX Information Elements: the applicationDescription, applicationTag and the applicationName.

New Information Elements to be added to the IPFIX Information Element registry at [IANA-IPFIX] are listed below.

EDITOR'S NOTE: the XML specification in Appendix A must be updated with the elementID values allocated below.

6.1. applicationDescription

Name: applicationDescription
Description:
  Specifies the description of an application.
Abstract Data Type: string
Data Type Semantics:
ElementId: 94
Status: current

6.2. applicationTag

```
Name: applicationTag
Description:
  Specifies an Application Tag.
  (EDITOR'S NOTE: reference this document).
Abstract Data Type: octetArray
Data Type Semantics: identifer
ElementId: 95
Status: current
```

6.3. applicationName

```
Name: applicationName
Description:
  Specifies the name of an application.
Abstract Data Type: string
Data Type Semantics:
ElementId: 96
Status: current
```

7. Security Considerations

The same security considerations as for the IPFIX Protocol [RFC5101] apply.

8. References

8.1. Normative References

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

[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.

8.2. Informative References

[RFC792] J. Postel, Internet Control Message Protocol, RFC 792, September 1981.

[RFC3917] Quittek, J., Zseby, T., Claise, B., and S. Zander, Requirements for IP Flow Information Export, RFC 3917, October 2004.

[RFC5103] Trammell, B., and E. Boschi, "Bidirectional Flow Export Using IP Flow Information Export (IPFIX)", RFC 5103, January 2008.

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

[RFC5471] Schmoll, C., Aitken, P., and B. Claise, "Guidelines for IP Flow Information Export (IPFIX) Testing", RFC 5471, March 2009.

[RFC5473] Boschi, E., Mark, L., and B. Claise, "Reducing Redundancy in IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Reports", RFC 5473, March 2009.

[RFC5476] Claise, B., Ed., "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, March 2009.

[IANA-IPFIX] http://www.iana.org/assignments/ipfix/ipfix.xhtml

[IANA-PORTS] http://www.iana.org/assignments/port-numbers

[IANA-PROTO] http://www.iana.org/assignments/protocol-numbers

[CISCO] http://www.cisco.com

9. Acknowledgement

The authors would like to thank their many colleagues across Cisco Systems who made this work possible.

10. Authors' Addresses

Benoit Claise
Cisco Systems Inc.
De Kleetlaan 6a b1
Diegem 1813
Belgium

Phone: +32 2 704 5622
EMail: bclaise@cisco.com

Paul Aitken
Cisco Systems (Scotland) Ltd.
96 Commercial Quay
Commercial Street
Edinburgh, EH6 6LX, United Kingdom

Phone: +44 131 561 3616
EMail: paitken@cisco.com

Nir Ben-Dvora
Cisco Systems Inc.
32 HaMelacha St.,
P.O.Box 8735, I.Z.Sapir
South Netanya, 42504
Israel

Phone: +972 9 892 7187
EMail: nirbd@cisco.com

Appendix A. Additions to XML Specification of IPFIX Information Elements

This appendix contains additions to the machine-readable description of the IPFIX information model coded in XML in Appendix A and Appendix B in [RFC5102]. Note that this appendix is of informational nature, while the text in section Error! Reference source not found.(generated from this appendix) is normative.

The following field definitions are appended to the IPFIX information model in Appendix A of [RFC5102].

```
<field name="applicationDescription"
         dataType="string"
         group="application"
         elementId="94" applicability="all" status="current">
    <description>
      <paragraph>
         Specifies the description of an application.
      </paragraph>
    </description>
  </field>

  <field name="applicationTag"
         dataType="octetArray"
         group="application"
         dataTypeSemantics="identifer"
         elementId="95" applicability="all" status="current">
    <description>
      <paragraph>
         Specifies an Application Tag.
      </paragraph>
    </description>
  </field>

  <field name="applicationName"
         dataType="string"
         group="application"
         elementId="96" applicability="all" status="current">
    <description>
      <paragraph>
         Specifies the name of an application.
      </paragraph>
    </description>
  </field>
```