Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 16, 2014

T. Eckert, Ed.
A. Zamfir
A. Choukir
C. Eckel
Cisco Systems, Inc.

July 15, 2013

Protocol Independent Encoding for Signaling Flow Characteristics
draft-choukir-tsvwg-flow-metadata-encoding-01

Abstract

This document describes a protocol independent encoding for flow characteristics (a.k.a. metadata). A flow is defined as a set of IP packets passing through a network in a given direction. All packets belonging to a particular flow have a set of common properties (e.g. IP, port, transport). Flow metadata exposes key characteristics of the flow such as the originating application, the type of media in use (e.g. audio, video) and others as defined in [I-D.eckert-intarea-flow-metadata-framework]. The flow characteristics are expressed in terms of information elements. These information elements are signaled either out of band or in band but always along the same path of the flow associated with the application.

[I-D.eckert-intarea-flow-metadata-framework] defines the overall framework for flow metadata and the definition of the flow characteristics, whereas this document captures the encoding of these characteristics. The mapping of flow metadata encoding to different signaling protocols is outside the scope of this document.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes a protocol independent encoding for flow characteristics (a.k.a. metadata). A flow is defined as a set of IP packets passing through a network in a given direction. All packets belonging to a particular flow have a set of common properties (e.g. IP, port, transport). Flow metadata exposes key characteristics of the flow such as the originating application, the type of media in use (e.g. audio, video) and others as defined in [I-D.eckert-intarea-flow-metadata-framework]. The flow characteristics are expressed in terms of information elements. These information elements are signaled either out of band or in band but always along the same path of the flow associated with the application.

As flow characteristics across different signaling protocols are identical, they benefit from a single definition and encoding irrespective of the signaling protocol in use (e.g. RSVP [I-D.zamfir-tsvwg-flow-metadata-rsvp], STUN [I-D.martinsen-mmusic-malice], and PCP [I-D.wing-pcp-flowdata]). Different network deployments call for different protocols or combination of protocols as described in [I-D.eckert-intarea-flow-metadata-framework]. The flow characteristics can be processed by intermediate network nodes for the purpose of applying a particular treatment to the flow or simply for gathering insight on the nature of the traffic crossing the network node.

Flows, and the corresponding metadata, are inherently unidirectional, in the direction from the source to the destination (e.g. from Alice to Bob). In some cases, there may be a related flow in the reverse direction (e.g. from Bob to Alice), but this is treated as a separate flow, not a bidirectional flow. The metadata can characterize data in the same direction as the flow (upstream) or in the opposite direction (downstream). The encoding mechanism enabling signaling for either or both directions. The metadata can be signaled by the application itself and/or by network elements that have visibility of the flow data. The encoding supports distinguishing between attribute information originated by an application from attribute information originated by a network device. The encoding allows to segregates information coming from the application from information coming from the network.

1.1. Requirements Language

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].

2. Encoding Overview

2.1. General Principles

This specification assumes that the flow is specified by the transport protocol which carries the metadata. As an example, in STUN, flow identifiers such as IP addresses and ports are present in layer 3 and 4 headers of STUN messages (see [I-D.martinsen-mmusic-malice]). In RSVP, the same is obtained from the SESSION and SENDER-TEMPLATE objects (see [I-D.zamfir-tsvwg-flow-metadata-rsvp]). In PCP the source IP is part of the request common header; other flow identifiers need to be embedded in an opcode data or an option (see[I-D.wing-pcp-flowdata]).

The Flow Metadata characteristics are to be interpreted in the context of the flow defined by the signaling protocol. In this specification Flow Metadata encoding does not carry any flow identifiers but merely the flow characteristics. The specification could be extended to carry the flow identifiers if needed.

The encoding defined herein does not relate to any specific signaling but rather allows different signaling protocols to transport flow characteristics. As the encoding is shared amongst several protocols, it is versioned independently to allow, if needed, its evolution without impacting the signaling protocol.

2.2. Encoding Goals

The following goals have been considered in the design of the encoding:

- Transport independence

- Allow for a standard namespace as well as vendor specific namespaces

- Support multiple producers of flow characteristics

- Ability to encode flow characteristics for both the flow itself (upstream) and the flow in the reverse direction (downstream).

- Ability to communicate flow characteristics from an application to the network as well as from the network back to the application

- Extensibility while allowing for backwards compatibility

- Flexibility

- Support for integrity, authentication and authorization on a per producer basis

- Compact encoding

2.2.1. Transport independence

One goal of this proposal is to provide an encoding that can be used by more than one transport protocol. This should help maintain consistency across standardization of flow metadata usage by various signaling protocols, and it should simplify implementations that make use of different signaling protocols when transporting flow metadata. One example is an application that may use different signaling protocols depending on the environment, peer protocol support, etc. Another is a middlebox on an administrative boundary that may need to perform protocol interworking functions.

2.2.2. Standard and Vendor Specific Namespaces

Vendors need the ability to define and use proprietary Metadata when they are delivering a pre-standard feature or product or when the encoded information is of commercially sensitive nature. This specification provides support for both standard and vendor specific defined flow characteristics.

2.2.3. Multiple Producers

Multiple producers may contribute flow characteristics to the Flow Metadata information associated with a given flow.

Applications are one category of candidates for generating Flow Metadata as they have precise knowledge of the flows they insert into the network. Middleboxes constitute a second class of Metadata producers. Deep Packet Inspection engines are deployed to recognize the originator and nature of the flows traversing a network. Media Termination Points (e.g. MCU, transcoders) are deployed to offer additional services to applications. Media Termination Points have knowledge of the transformations they apply on the flow they receive and can therefore update the characteristics of the flow. Other proxies and gateways exist for other applications and could produce information in relation to the flow.

2.2.4. Upstream and Downstream

As explained in the introduction, a flow is unidirectional by definition, but some use cases and signaling protocols require or allow to signal both upstream and downstream flow characteristics. For example, in the context of a home user that needs to prioritize its upstream and downstream flow an end-to-edge protocol can expose flow characteristics to the edge ISP node controlling its access link for both its upstream and downstream flow. This allows the edge node to apply proper treatment to both directions.

2.2.5. Application to Network and Network to Application

In accordance with [I-D.eckert-intarea-flow-metadata-framework], flow characteristics may be communicated both from application to network as well as from network to application. The encoding rules are the same regardless of the direction of the communication. The ability to differentiate between the two is provided by the transport protocol. For example, when using PCP, application to network communication is via a PCP request, and network to application communication is via a PCP response.

2.2.6. Extensibility

New use cases and new deployment scenarios will require the use of new flow characteristics. For this reason the encoding should support new metadata (i.e. new information elements) in a backwards compatible way. New information element definitions supplement but do not redefine existing definitions. An application or a network node always signals its currently supported set of information elements and devices leverage the subset they understand for the purpose of applying treatment to, or gathering information about, the application flows.

2.2.7. Flexibility

Distinct use cases and individual applications have a need for different subsets of information elements. The encoding should support the signaling of any subset of information elements for that purpose. For example, a video conferencing application might need to signal metadata for both its audio and video flows. A video surveillance application might signal video flows only, but may need to indicate which one has priority based on embedded analytics.

2.2.8. Per Producer Security

Treatment applied on the basis of metadata may involve the consumption of scarce network resources and therefore contribute to

their exhaustion. Consequently, integrity, authentication, and authorization are all important aspects of any security mechanism used to secure the metadata.

This specification defines an optional security element container; however, the actual security mechanism to be used is outside the scope of this specification.

2.2.9. Compact Encoding

One of the goals of the encoding described in this specification is to be compact and consume minimal space in the signaling protocol payload. Most of the protocols have limited space for Metadata purposes and do not support semantic fragmentation. The strategy of the encoding is to minimize the encoding structures used for the common signaling case. The common case is foreseen to be the application signaling standard flow characteristics.

3. Encoding specification

3.1. Layout

This section describes the encoding layout proposed by this specification. It describes the following:

- How the application and network producers coexist using sections in Figure 1

- Application of an optional security token to a section in Figure 2

- The division of a section into standard and vendor specific sub-sections in Figure 3

- The division of a sub-section into upstream and downstream blocks in Figure 4

- A full example using all the encoding building blocks in Figure 5

3.1.1. Sections

The flow characteristics are grouped in sections within the encoding. A section pertains to an application or to a network producer. To segregate application and network producer sections the encoding uses a network marker. The application section does not use a network marker and therefore must come first if present. The encoding MUST contain at least an application or a network section. Figure 1 shows an example that contains an application section and two network sections.

```
                    +--------------------------------+
                    |      Application Section       |
                    +--------------------------------+
                    +--------------------------------+
                    |         Network-1 Marker       |
                    +--------------------------------+
                    +--------------------------------+
                    |                                |
                    |                                |
                    |         Network-1 Section      |
                    |                                |
                    +--------------------------------+
                    +--------------------------------+
                    |         Network-2 Marker       |
                    +--------------------------------+
                    +--------------------------------+
                    |                                |
                    |                                |
                    |         Network-2 Section      |
                    |                                |
                    |                                |
                    |                                |
                    +--------------------------------+
```

Figure 1: Encoding section

3.1.2. Security Tokens

A section MAY include at most one security token. The security token, if present, MUST appear at the beginning of the section. In the following example, a separate security token is added to each section contained in the previous example.

```
                    +--------------------------------+
                    |         Security Token         |
                    +--------------------------------+
                    +--------------------------------+
                    |      Application Section       |
                    +--------------------------------+
                    +--------------------------------+
                    |         Network-1 Marker       |
                    +--------------------------------+
                    +--------------------------------+
                    |         Security Token N-1     |
                    +--------------------------------+
                    +--------------------------------+
                    |                                |
                    |                                |
```

```
                    |         Network-1 Section      |
                    |                                |
                    +--------------------------------+
                    +--------------------------------+
                    |         Network-2 Marker       |
                    +--------------------------------+
                    +--------------------------------+
                    |         Security Token N-2     |
                    +--------------------------------+
                    +--------------------------------+
                    |                                |
                    |                                |
                    |         Network-2 Section      |
                    |                                |
                    |                                |
                    |                                |
                    +--------------------------------+
```

Figure 2: Encoding security tokens

3.1.3. Subsections

A section may be divided into standard and vendor sub-sections. A section MUST at least have one subsection. A section MUST contain at most one standard sub-section and can contain multiple vendor subsections for different vendors. A standard and a vendor sub-section are segregated through a vendor marker. The standard subsection does not use the vendor marker and therefore must come first if present. Figure 3 shows a sample section content.

```
                    +--------------------------------+
                    |                                |
                    |                                |
                    |      Standard Subsection       |
                    |                                |
                    +--------------------------------+
                    +--------------------------------+
                    |         Vendor-1 Marker        |
                    +--------------------------------+
                    +--------------------------------+
                    |       Vendor-1 Subsection      |
                    |                                |
                    |                                |
                    |                                |
                    |                                |
                    |                                |
                    +--------------------------------+
                    +--------------------------------+
```

```
|         Vendor-2 Marker        |
+--------------------------------+
+--------------------------------+
|       Vendor-2 Subsection      |
|                                |
|                                |
|                                |
|                                |
|                                |
+--------------------------------+
```

Figure 3: Encoding subsections

3.1.4. Upstream and Downstream Blocks

A subsection MUST contain at least one upstream or downstream block. A subsection contains at most one upstream block and at most one downstream block. Upstream and downstream blocks are composed of metadata tags, with each tag representing an encoding of a specific information element. If the upstream and downstream blocks are both present, the upstream block MUST come first.

```
+--------------------------------+
|          Upstream block        |
| +------------+  +------------+ |
| |  MD tag    |  |  MD tag    | |
| +------------+  +------------+ |
+--------------------------------+
+--------------------------------+
|        Downstream block        |
| +------------+  +------------+ |
| |  MD tag    |  |  MD tag    | |
| +------------+  +------------+ |
+--------------------------------+
```

Figure 4: Encoding upstream and downstream blocks

3.1.5. Complete Encoding Example

Figure 5 shows a complete example combining the application and network sections together with their standard and vendor sub-sections. The metadata tags appearing in a standard and in a vendor sub-section are managed by separate registries. See [I-D.eckert-intarea-flow-metadata-framework] for a full coverage of the information model and how the registries are handled.

```
+--------------------------------+
|         Security Token         |
```

```
                 +--------------------------------+
                 +--------------------------------+ ^  ^
                 |          Upstream block        | |  |A
                 | +------------+  +------------+ | |  |P
                 | |  MD tag    |  |  MD tag    | | |S |P
                 | +------------+  +------------+ | |T |L
                 +--------------------------------+ |D |I
                 +--------------------------------+ |  |C
                 |         Downstream block       | |  |A
                 | +------------+                 | |  |T
                 | |  MD tag    |                 | |  |I
                 | +------------+                 | |  |O
                 +--------------------------------+ v  |N
                 +--------------------------------+    |
                 |     Vendor section marker      |    |S
                 +--------------------------------+    |E
                 +--------------------------------+ ^  |C
                 |          Upstream block        | |V |T
                 | +------------+  +------------+ | |N |I
                 | |  MD tag    |  |  MD tag    | | |D |O
                 | +------------+  +------------+ | |  |N
                 +--------------------------------+ v  v
                 +--------------------------------+    ^
                 |     Network section marker     |    |
                 +--------------------------------+    |
                 +--------------------------------+    |N
                 |         Security Token         |    |E
                 +--------------------------------+    |T
                 +--------------------------------+    |W
                 |         Downstream block       |    |O
                 | +------------+                 |    |R
                 | |  MD tag    |                 |    |K
                 | +------------+                 |    |
                 +--------------------------------+    |
                 +--------------------------------+    |S
                 |     Vendor section marker      |    |E
                 +--------------------------------+    |C
                 +--------------------------------+    |T
                 |         Downstream block       |    |I
                 | +------------+                 |    |O
                 | |  MD tag    |                 |    |N
                 | +------------+                 |    |
                 +--------------------------------+    v
```

Figure 5: Complete encoding example

3.1.6. Compact Encoding Example

Figure 6 shows an encoding example for flow metadata standard characteristics produced by an application for the upstream (same as 5-tuple) direction. As can be seen in the figure no network marker is used as we are signaling for the application. In the same way there is no vendor marker as we are signaling standard flow characteristics. This example also assumes a use case where no security token is needed. Further examples are given in Appendix A.

```
+---------------------------------+
|           Upstream block        |
| +------------+  +------------+  |
| |  MD tag    |  |  MD tag    |  |
| +------------+  +------------+  |
+---------------------------------+
```

Figure 6: Compact encoding example

3.2. Encoding Structures

This section explores the encoding looking more closely at the encoding structures.

Figure 7 shows the encoding used by an application using only standard metadata tags.

Figure 8 shows the encoding used by an application using only vendor specific metadata tags.

Figure 9 shows the encoding for network producers using only standard metadata tags.

The three scenarios expose all the encoding structures. These structures may be combined in various ways to support other scenarios.

The encoding makes use of Type Length Value (TLV) as the base building block, plus some level of nesting to create the different encoding structures. The type indicates which encoding structure is in use. In case of a marker, the length gives the size of the marker but not of the delimited section or sub-section.

As explained previously, application and network sections MUST contain at least one standard or vendor sub-section and MAY contain a security token. The value of the security token TLV is broken down in two parts, a security-scheme indicating the security method used and the security-value holding the security payload specific to the security scheme. The definition of the different security schemes and their payloads are left to a separate document.

The value of the upstream and downstream block TLVs are subdivided in metadata tags. Each tag is itself a TLV specifying a flow characteristic. A metadata tag MUST appear only once in an upstream or a downstream block. On the one hand the security token, the upstream and the downstream block, the vendor and network marker types are defined within the same registry. On the other hand the tag types are defined in a separate registry from the enclosing encoding structures. The separation of the registries is possible as the metadata tags are part of the upstream and downstream block TLV value and therefore do not collide with the encoding structures.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+^S
   |        Security-type          |             Length            ||E
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+|C
   |       Security-scheme         |                               :|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               :|T
   :                       Security-value                          :|O
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+vK
   |        Upstream-type          |             Length            |  ^U
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+^M|P
   |           Tag-type            |             Length            ||D|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| |B
   :                           Tag-value                           :|T|L
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+v |O
   :                            ...                                :  |C
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  vK
   |        Downstream-type        |             Length            |  ^D
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |N
   |           Tag-type            |             Length            |  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |B
   :                           Tag-value                           :  |L
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |O
   :                            ...                                :  |C
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  vK
```

Figure 7

Figure 8 adds the vendor sub-section marker which starts a vendor section. The vendor marker is a TLV whose type is defined in the same registry as the security token. Its value is the vendor's Private Enterprise Number (PEN) allocated by IANA. The vendor marker does not include the downstream and the upstream block but rather sets the context to interpret them. Multiple vendor sub-sections within the same application or network section are allowed as long as they pertain to different vendors.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        Security-type          |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |       Security-scheme         |                               :
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               :
   :                       Security-value                          :
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ^
   |           PEN-type            |             Length            | |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
   |                            PEN-id                             | |V
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |E
   |        Upstream-type          |             Length            | |N
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |D
   |           Tag-type            |             Length            | |O
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |R
   :                           Tag-value                           : |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |S
   :                            ...                                : |E
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |C
   |        Downstream-type        |             Length            | |T
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |I
   |           Tag-type            |             Length            | |O
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |N
   :                           Tag-value                           : |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
   :                            ...                                : |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ v
```

Figure 8

Figure 9 adds the network marker that starts a network section. The network marker is a TLV whose type is defined within the same registry as the security token. The value of the network marker is the network precedence that indicates the administrative preference for the network producer flow characteristics. The precedence allows to merge information from different network producers and retain only the preferred one.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ^
   |         Network-type          |             Length            | |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
   |                          Precedence                           | |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
   |        Security-type          |             Length            | |P
```

```
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |R
   |       Security-scheme         |                               : |O
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               : |D
   :                       Security-value                          : |U
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |C
   |        Upstream-type          |             Length            | |E
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |R
   |           Tag-type            |             Length            | |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |S
   :                           Tag-value                           : |E
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |C
   :                            ...                                : |T
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |I
   |        Downstream-type        |             Length            | |O
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |N
   |           Tag-type            |             Length            | |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
   :                           Tag-value                           : |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
   :                            ...                                : |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ v
```

Figure 9

All the constructs above can be combined to signal standard and vendor specific metadata tags for different producers and allow to secure each producer's information independently.

3.3. ABNF

```
   MD-block = Version (Application-block / 1*Network-block /
   (Application-block 1*Network-block))

   Network-blocks = Network-tlv Producer-block

   Application-block = Producer-block ; For the application we do not
   ; require the Producer-tlv

   Producer-block = [Security-tlv] (Standard-block / 1*Vendor-block /
   (Standard-block 1*Vendor-block))

   Vendor-blocks = PEN-tlv Flow-block

   Standard-block = Flow-block; We do not require the PEN-tlv
   ; for the standard metadata tags

   Flow-block = Upstream-tlv / Downstream-tlv /
   (Upstream-tlv Downstream-tlv)
```

```
   ; If both present, upstream must come first

   PEN-tlv = PEN-type Length PEN-id

   Network-tlv = Network-type Length Precedence

   Security-tlv = Security-type Length Security-scheme Security-value

   Upstream-tlv = Upstream-type Length Upstream-value

   Upstream-value = Attribute-list

   Downstream-tlv = Downstream-type Length Downstream-value

   Downstream-value = Attribute-list

   Attribute-list = 1*(Attribute-tlv)

   Attribute-tlv = Tag-type Length Attribute-value

   ;---------
   ;TERMINALS
   ;---------

   Version = %x01 ;  NEW-VER will be picked up as the first
   ;  version of the encoding

   PEN-id = 4(OCTET); Private Enterprise Number defined by IANA

   Length = 2(OCTET); 16-bit length field

   Precedence = 4(OCTET);  Indicates the preferred source of information
   ; for a producer-type

   Security-scheme = OCTET; Type of security used

   Security-value =  *(OCTET)
   ; length of this field must match Length of Security-tlv + 2

   Tag-type = 2(OCTET); Value according to IANA/Vendor-specific registry

   Producer-type = Zero %x01; The first foreseen producer is MD-NETWORK
   ; to cover for DPI engines, gateways and others
   ; Further values may be allocated later


   Security-type = Zero %x00 ;
```

```
   Upstream-type = Zero %x01 ;

   Downstream-type = Zero %x02 ;

   PEN-type = Zero %x03 ;

   Network-type = Zero %x04

   Attribute-value = *(OCTET) ;

   Zero = %x00
```

Figure 10

4. Security Considerations

A security token, as described in Section 3.1.2, is a mechanism provided as part of the encoding to protect flow characteristics. A signaling protocol used to transport the encoded metadata may provide additional security mechanisms. The protocol specific and encoding specific security mechanisms may be used in combination to achieve the required level of security.

5. Acknowledgements

We would like to thank Yann Poupet and Davide Cuda for reviewing this document and helping us proof its content. We would also like to express our gratitude to Sergio Mena de la Cruz for contributing ideas, proofing the text and for his work on validating the grammar.

6. References

6.1. Normative References

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

6.2. Informative References

[I-D.eckert-intarea-flow-metadata-framework]
Eckert, T., Penno, R., Choukir, A., and C. Eckel, "A Framework for Signaling Flow Characteristics between Applications and the Network", draft-eckert-intarea-flow-metadata-framework-00 (work in progress), July 2013.

[I-D.martinsen-mmusic-malice]

Penno, R., Martinsen, P., Wing, D., and A. Zamfir, "Meta-data Attribute signaLling with ICE", draft-martinsen-mmusic-malice-00 (work in progress), July 2013.

[I-D.wing-pcp-flowdata]
Wing, D., Penno, R., and T. Reddy, "PCP Flowdata Option", draft-wing-pcp-flowdata-00 (work in progress), July 2013.

[I-D.zamfir-tsvwg-flow-metadata-rsvp]
Eckert, T., Zamfir, A., and A. Choukir, "Flow Metadata Signaling with RSVP", draft-zamfir-tsvwg-flow-metadata-rsvp-00 (work in progress), July 2013.

Appendix A. Encoding usage examples

TBD

Authors' Addresses

Toerless Eckert (editor)
Cisco Systems, Inc.
San Jose
US

Email: eckert@cisco.com

Anca Zamfir
Cisco Systems, Inc.
Lausanne
CH

Email: ancaz@cisco.com

Amine Choukir
Cisco Systems, Inc.
Lausanne
CH

Email: amchouki@cisco.com

Charles Eckel
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
US

Email: eckelcu@cisco.com