I2RS working group
Internet-Draft
Intended status: Standards Track
Expires: February 28, 2016

S. Hares
Huawei
D. Migault
J. Halpern
Ericsson

August 27, 2015

I2RS Security Related Requirements
draft-hares-i2rs-auth-trans-05

Abstract

This presents security-related requirements for the I2RS protocol for mutual authentication, transport protocols, data transfer and transactions.

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 28, 2016.

Copyright Notice

Copyright (c) 2015 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 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. Introduction

The Interface to the Routing System (I2RS) provides read and write access to information and state within the routing process. The I2RS client interacts with one or more I2RS agents to collect information from network routing systems.

This document describes the requirements for the I2RS protocol in the security-related areas of mutual authentication of the I2RS client and agent, the transport protocol carrying the I2RS protocol messages, and the atomicity of the transactions. These requirements align with the description of the I2RS architecture found in [I-D.ietf-i2rs-architecture] document.

[I-D.haas-i2rs-ephemeral-state-reqs] discusses I2RS roles-based write conflict resolution in the ephemeral data store using the I2RS Client Identity, I2RS Secondary Identity and priority. The draft [I-D.ietf-i2rs-traceability] describes the traceability framework and its requirements for I2RS. The draft [I-D.ietf-i2rs-pub-sub-requirements] describes the requirements for I2RS to be able to publish information or have a remote client subscribe to an information data stream.

1.1. Definitions

This document utilizes the definitions found in the following drafts: [RFC4949], and [I-D.ietf-i2rs-architecture]

Specifically, this document utilizes the following definitions:

access control

[RFC4949] defines access control as the following:

1)(I))protection of system resources against unauthorized use;

2)(I)process by which use of system resources is regulated according to a security policy and is permitted only by authorized entities (users, programs, processes, or other systems) according to that policy;

3)(I) (formal model) Limitations on interactions between subjects and objects in an information system;

4)(O) "The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner.";

5.(O) /U.S. Government/ A system using physical, electronic, or human controls to identify or admit personnel with properly authorized access to a SCIF.

Authentication

[RFC4949] describes authentication as the process of verifying (i.e., establishing the truth of) an attribute value claimed by or for a system entity or system resource. Authentication has two steps: identify and verify.

Data Confidentiality

[RFC4949] describes data confidentiality as having two properties: a) data is not disclosed to system entities unless they have been authorized to know, and b) data is not disclosed to unauthorized individuals, entities or processes. The key point is that confidentiality implies that the originator has the ability to authorize where the information goes. Confidentiality is important for both read and write scope of the data.

Data Integrity

[RFC4949] states data integrity includes

1. (I)The property that data has not been changed, destroyed, or

2. (O) "The property that information has not been modified or destroyed in an unauthorized manner."

Data Privacy

[RFC4949] describes data privacy as a synonym for data confidentiality. This I2RS document will utilize data privacy as a synonym for data confidentiality.

Mutual Authentication

[RFC4949] implies that mutual authentication exists between two interacting system entities. Mutual authentication in I2RS implies that both sides move from a state of mutual suspicion to mutually authenticated communication afte each system has been identified and validated by its peer system.

role

[RFC4949] describes role as:

1) (I) A job function or employment position to which people or other system entities may be assigned in a system. (See: role-based access control. Compare: duty, billet, principal, user.)

2) (O) /Common Criteria/ A pre-defined set of rules establishing the allowed interactions between a user and the TOE.

The I2RS uses the common criteria definition.

role

[RFC4949] describes role-based access control as: (I) A form of identity-based access control wherein the system entities that are identified and controlled are functional positions in an organization or process.

Security audit trail

[RFC4949] (page 254) describes a security audit trail as a chronological record of system activities that is sufficient to enable the reconstruction and examination of the sequence environments and activities surrounding or leading to an operation, procedure, or event in a security-relevant transaction from inception to final results. Requirements to support a security audit is not covered in this document. The draft [I-D.ietf-i2rs-traceability] describes traceability for I2RS interface and protocol. Traceability is not equivalent to a security audit trail.

I2RS the following phrase that incorporates an [RFC4949] definition:

I2RS protocol data integrity

The transfer of data via the I2RS protocol has the property of data integrity described in [RFC4949].

2. Security-Related Requirements

The security for the I2RS protocol requires mutually authenticated I2RS clients and I2RS agents. The I2RS client and I2RS agent using the I2RS protocol MUST be able to exchange data over a secure transport, but some functions may operate on non-secure transport. The I2RS protocol MUST BE able to provide atomicity of a transaction, but it is not required to have multi-message atomicity and rollback mechanism transactions. Multiple messages transactions may be impacted by the interdependency of data. This section discusses these details of these security requirements.

2.1. Mutual authentication of I2RS client and I2RS Agent

The I2RS architecture [I-D.ietf-i2rs-architecture] sets the following requirements:

- SEC-REQ-01: All I2RS clients and I2RS agents MUST have at least one unique identifier that uniquely identifies each party.
- SEC-REQ-02: The I2RS protocol MUST utilize these identifiers for mutual identification of the I2RS client and I2RS agent.
- SEC-REQ-03:An I2RS agent, upon receiving an I2RS message from a I2RS client, MUST confirm that the I2RS client has a valid identifier.
- SEC-REQ-04: The I2RS client, upon receiving an I2RS message from an I2RS agent, MUST confirm the I2RS agent’s identifier .
- SEC-REQ-05: Identifier distribution and the loading of these identifiers into I2RS agent and I2RS Client SHOULD occur outside the I2RS protocol.
- SEC-REQ-06: The I2RS protocol SHOULD assume some mechanism (IETF or private) will distribute or load identifiers so that the I2RS client/agent has these identifiers prior to the I2RS protocol establishing a connection between I2RS client and I2RS agent.
- SEC-REQ-07: Each Identifier MUST be linked to one priority

- SEC-REQ-08: Each Identifier is associated with one secondary identifier during a particular read/write sequence, but the secondary identifier may vary during the time a connection between the I2RS client and I2RS agent is active. The variance of the secondary identifier allows the I2RS client to be associated with multiple applications and pass along an identifier for these applications in the secondary identifier.

2.2. Transport Requirements Based on Mutual Authentication

SEC-REQ-09: The I2RS protocol MUST be able to transfer data over a secure transport and optionally be able to transfer data over a non-secure transport. A secure transport MUST provide data confidentiality, data integrity, and replay prevention.

Note:The non-secure transport be used for publishing telemetry data that was specifically indicated to non-confidential in the data model. The configuration of ephemeral data in the I2RS Agent by the I2RS client SHOULD be done over a secure transport. It is anticipated that the passing of most I2RS ephemeral state operational status SHOULD be done over a secure transport. Data models SHOULD clearly annotate what data nodes can be passed over an insecure connection. The default transport is a secure transport.

SEC-REQ-10: A secure transport MUST be associated with a key management solution that can guarantee that only the entities having sufficient privileges can get the keys to encrypt/decrypt the sensitive data. Per BCP107 [RFC4107] this key management system SHOULD be automatic, but MAY BE manual if the following constraints from BCP107:

a)environment has limited bandwidth or high round-trip times,

b)the information being protected has a low value and

c)the total volume over the entire lifetime of the long-term session key will be very low,

d)the scale of the deployment is limited.

Most I2RS environments (I2RS Client - I2S Agents) will not have this environment, but a few I2RS use case provide limited non-secure light-weight telemetry messages that have these requirements. An I2RS data model must indicate which portions can be served by manual key management.

SEC-REQ-11: The I2RS protocol MUST be able to support multiple secure transport sessions providing protocol and data communication between

an I2RS Agent and an I2RS client. However, a single I2RS Agent to I2RS client connection MAY elect to use a single secure transport session or a single non-secure transport session.

SEC-REQ-12: The I2RS Client and I2RS Agent protocol SHOULD implement mechanisms that mitigate DoS attacks

2.3. Data Confidentiality Requirements

SEC-REQ-13: In a critical infrastructure, certain data within routing elements is sensitive and read/write operations on such data MUST be controlled in order to protect its confidentiality. For example, most carriers do not want a router's configuration and data flow statistics known by hackers or their competitors. While carriers may share peering information, most carriers do not share configuration and traffic statistics. To achieve this, access control to sensitive data needs to be provided, and the confidentiality protection on such data during transportation needs to be enforced.

2.4. Data Integrity Requirements

SEC-REQ-14: An integrity protection mechanism for I2RS SHOULD be able to ensure the following: 1) the data being protected is not modified without detection during its transportation and 2) the data is actually from where it is expected to come from 3) the data is not repeated from some earlier interaction of the protocol. That is, when both confidentiality and integrity of data is properly protected, it is possible to ensure that encrypted data is not modified or replayed without detection.

SEC-REQ-15: The integrity that the message data is not repeated means that I2RS client to I2RS agent transport SHOULD protect against replay attack

Requirements SEC-REQ-13 and SEC-REQ-14 are SHOULD requirements only because it is recognized that some I2RS Client to I2RS agent communication occurs over a non-secure channel. The I2RS client to I2RS agent over a secure channel would implement these features. In order to provide some traceability or notification for the non-secure protocol, SEC-REQ-16 suggests traceability and notification are important to include for any non-secure protocol.

SEC-REQ-17: The I2RS message traceability and notification requirements requirements found in [I-D.ietf-i2rs-traceability] and [I-D.ietf-i2rs-pub-sub-requirements] SHOULD be supported in communication channel that is non-secure to trace or notify about potential security issues

2.5. Role-Based Data Model Security

The [I-D.ietf-i2rs-architecture] defines a role or security role as specifying read, write, or notification access by a I2RS client to data within an agent's data model.

SEC-REQ-18: The rules around what role is permitted to access and manipulate what information plus a secure transport (which protects the data in transit) SHOULD ensure that data of any level of sensitivity is reasonably protected from being observed by those without permission to view it, so that privacy requirements are met.

SEC-REQ-19: Role security MUST work when multiple transport connections are being used between the I2RS client and I2RS agent as the I2RS architecture [I-D.ietf-i2rs-architecture] states. These transport message streams may start/stop without affecting the existence of the client/agent data exchange. TCP supports a single stream of data. SCTP [RFC4960] provides security for multiple streams plus end-to-end transport of data.

SEC-REQ-20: I2RS clients MAY be used by multiple applications to configure routing via I2RS agents, receive status reports, turn on the I2RS audit stream, or turn on I2RS traceability. Application software using I2RS client functions may host several multiple secure identities, but each connection will use only one identifier with one priority. Therefore, the security of each I2RS Client to I2RS Agent connection is unique.

Please note the security of the application to I2RS client connection is outside of the I2RS protocol or I2RS interface.

3. Acknowledgement

The author would like to thank Wes George, Ahmed Abro, Qin Wu, Eric Yu, Joel Halpern, Scott Brim, Nancy Cam-Winget, DaCheng Zhang, Alia Atlas, and Jeff Haas for their contributions to the I2RS security requirements discussion and this document.

4. IANA Considerations

This draft includes no request to IANA.

5. Security Considerations

This is a document about security requirements for the I2RS protocol and data modules. The whole document is security considerations.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, DOI 10.17487/RFC4107, June 2005, <http://www.rfc-editor.org/info/rfc4107>.

6.2. Informative References

[I-D.haas-i2rs-ephemeral-state-reqs]
Haas, J., "I2RS Ephemeral State Requirements", draft-haas-i2rs-ephemeral-state-reqs-00 (work in progress), May 2015.

[I-D.ietf-i2rs-architecture]
Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", draft-ietf-i2rs-architecture-09 (work in progress), March 2015.

[I-D.ietf-i2rs-problem-statement]
Atlas, A., Nadeau, T., and D. Ward, "Interface to the Routing System Problem Statement", draft-ietf-i2rs-problem-statement-06 (work in progress), January 2015.

[I-D.ietf-i2rs-pub-sub-requirements]
Voit, E., Clemm, A., and A. Prieto, "Requirements for Subscription to YANG Datastores", draft-ietf-i2rs-pub-sub-requirements-02 (work in progress), March 2015.

[I-D.ietf-i2rs-rib-info-model]
Bahadur, N., Folkes, R., Kini, S., and J. Medved, "Routing Information Base Info Model", draft-ietf-i2rs-rib-info-model-06 (work in progress), March 2015.

[I-D.ietf-i2rs-traceability]
Clarke, J., Salgueiro, G., and C. Pignataro, "Interface to the Routing System (I2RS) Traceability: Framework and Information Model", draft-ietf-i2rs-traceability-03 (work in progress), May 2015.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <http://www.rfc-editor.org/info/rfc4949>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <http://www.rfc-editor.org/info/rfc4960>.

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com

Daniel Migault
Ericsson
8400 boulevard Decarie
Montreal, QC HAP 2N2
Canada

Email: daniel.migault@ericsson.com

Joel Halpern
Ericsson
US

Email: joel.halpern@ericsson.com

I2RS
Internet-Draft
Intended status: Standards Track
Expires: August 15, 2016

S. Hares
L. Dunbar
Huawei

February 12, 2016

A Yang model for I2RS service topology
draft-hares-i2rs-service-topo-dm-06.txt

Abstract

This document defines I2RS protocol-independent service layer virtual topology data model. This data model utilizes the concepts in the generic I2RS topology model of virtual networks (node, links, termination points) and cross-layer topologies. This virtual service topology may be a composite layer created from the combination of protocol-dependent service layers. Protocol-dependent services layers include: L3VPN, L2VPN, EVPN, E-Tree, and others.

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 August 15, 2016.

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 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. Introduction

Service topology in [I-D.ietf-i2rs-yang-network-topo] includes the a virtual topology for a service layer above the L1, L2, and L3 layers. This virtual topology has the generic topology elements of node, link, and terminating point. The virtual service topology is a network-wide topology stored on one routing system which an I2RS agent is connected to.

The virtual service topology is a composite summary of the services available services gathered from the lower layer indications of L3VPN, L2VPN, and EVPN services, E-TREE services, Seamless MPLS topologies within an As and others. This is a "bottoms up" yang module providing composite protocol independent service topology based on these protocol services.

This "bottoms-up" yang model does provide a mechanism to link this bottoms up model to a top-down service model. One example of a top-down service model for L3 VPNs is the L3 Service yang data model [I-D.ietf-l3sm-l3vpn-service-model]. Although the two models are linked, the top-down service model cannot be derived from the lower layers.

1.1. 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 RFC2119 [RFC2119].

1.2. Base Model: the Service-Topology Component

The following diagram contains an informal graphical depiction of the main elements of the information model:

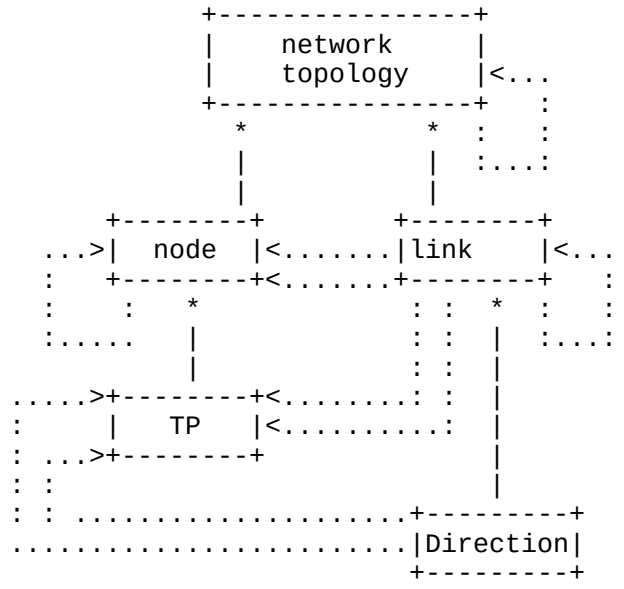

Figure 1

The link between the upper layer and the lower layer occurs by linking the bottoms up service network topologies to Top-down service topologies at certain service nodes to support transport of service across a virtual service link.

An example of the top-down service topology link to the bottoms up composite service topology may help. Suppose a bottoms up topology contains a composite of 3 L3VPN network topologies, 2 L2VPN network topologies, and 2 EVPN network topologies. Each of these physical networks can support virtual networks on top of the physical network. The service network base bottoms up is formed topology map with all of these topologies.

Suppose an L3SM has three VPN services topologies which support three services over 9 virtual topologies transiting the 2 of the L3VPN networks.

- VPN-svc 1: supporting hub-spoke flow for Customer 1 with 2 Cloud identifiers (2 topologies) with connecting the customers access at 3 sites

- o VPN-svc 2: supporting hub-spoke flow disjoint for Customer 2 with 2 Cloud identifiers (2 topologies) at 3 sites,

- o VPN-svc 3: supporting any-to-any flow for Customer 3 with 1 Cloud Identifier (5 topologies) at 3 sites.

Let us examine how VPN-svc 1 links to the composite cloud. Let us assume for simplicity of the example that the nodes providing the L3VPN provider equipment (PE) and the customer equipment (CE) at all sites are all unique. The diagram of how the top-down service topology meets the bottom up service topology is shown in figure

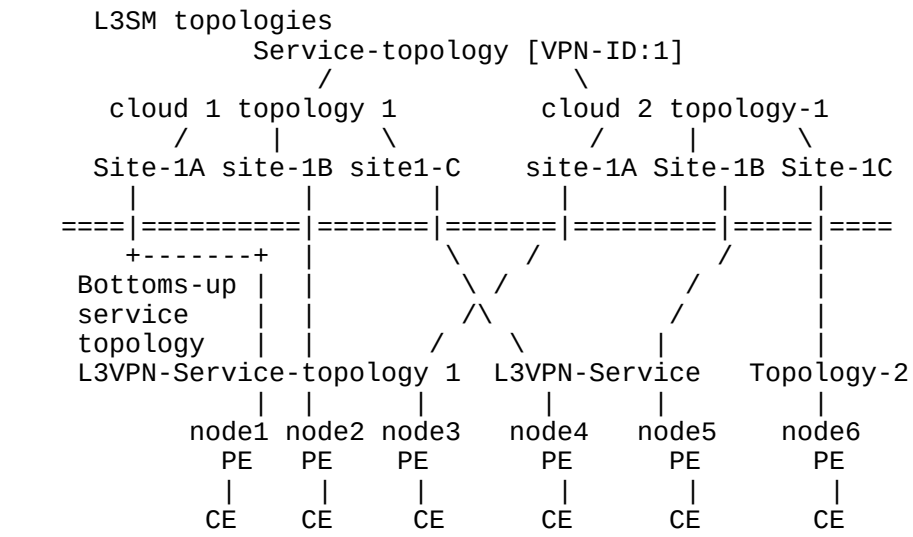

Figure 2

These two layers of service topologies are by two different composite models composite models and different supporiting models as follows:

- o Top-Down Provider Services with supporting L3SM model,

- o Bottom-Up I2RS Composite Services with supporting model from L3VPN, L2VPN, EVPN (only L3VPN used). The links between topologies occur at specific nodes.

2. High level Yang architecture

This section describes the Yang High level architecture.

2.1. Network level

The service topology network level defines the following high-level yang architeture:

```
module: i2rs-service-topologies
   augment /nw:network/nw:network-types:
      +--rw service-topologies-types
   augment /nw:network:
     +--rw service-topology-attributes
         +--rw name?   string
                 +--rw description? string
                 +--rw composite-flag* identity-ref
         +--rw tdsvc-supports-attributes*
                    [tdsvc-attr-name]
                    +--rw tdsvc-attr-name string
                        +--rw tdsvc-supports-attribute* identityref
```

Note: Composite flags are bottoms-up flags

Figure 3

The service topology attributes for a network include the following

name - name of the service topology,

description - description of service topology

composite-flags - bit mask with flags of service layer topologies network topology node available to create service topology from. These topologies include: L3VPN, L2VPN, and EVPN services, E-TREE services, Seamless MPLS topologies within an AS and others.

tdsvc-supports-attributes - composite topology supports top-down services topology attributes

tdsvc-supports-attr-name - name of top-down service attribute

tdsvc-supports-attribute - identity ref of service attribute (e.g. L3SM service for any-to-any)

2.2. Node level

```
module: i2rs-service-topologies
....
   augment /nw:network/nw:node
      +--rw node-service-attributes
         +--rw c-svc-node-name?   inet:domain-name
         +--rw c-svc-node-flag*   identityref;
                 +--rw tdsvc-node-supports-attributes*
                      [tdsvc-node-attr-name]
                    +--rw tdsvc-node-attr-name string;
                    +--rw tdsvc-node-supports-attribute identityref
                   // Top down attributes supported
```

The additional fields in the service attributes are the following:

c-svc-node-name - name of network node,

c-svc-node-flag - composite service topology node flag. The service node can be a member of one of the existing topology type (L3VPN, L2VPN, EVPN, E-TREE, Seamless MPLS, MPLS-TE, MPLS node, or I2RS created).

tdsvc-node-supports-attributes - node supports top-down services topology attributes

tdsvc-supports-node-attr-name - name of top-down service attribute

tdsvc-supports-node-attribute - identity ref of service attribute (e.g. L3SM service for any-to-any)

2.3. Service Link and Termination point

```
augment /nw:network/nt:link:
   +--rw service-link-attributes
      +--rw c-svc-link-name?     string
      +--rw c-svc-link-type identityref
      +--rw c-svc-link-metric?   uint32
              +--rw tdsvc-link-supports-attr* [name]
                 +--rw tdsvc-link-attr-name string
                     +--rw tdsvc-link-attribute identityref
augment /nw:network/nw:node/nt:termination-point:
   +--rw service-termination-point-attributes
          +--rw svc-tp-name string
              +--rw svc-tp-type identityref
              +--rw tdsvc-tp-support-attributes
                    +--rw tdsvc-tp-attr-name
                        +--rw tdsvc-tp-support-attribute
```

The augmentation to the service topology is the service link attributes which include:

c-svc-link-name - name of the link,

c-svc-link-type - the service link type supported by this logical link.

metric - the metric of the service type. This metric allows the composite link to store a svc level metric. 0 = no servic metric. 1-n values (1 best, n worse).

svc-attributes - the composite attributes of link

tdsvc-td-support-attributes - link support of Top-down attributes

tdsvc-supports-node-attr-name - name of top-down service attribute

tdsvc-supports-node-attribute - identity ref of service attribute (e.g. L3SM service for any-to-any)

The augmentation to the termination point include the following

svc-tp-name - name of termination point,

tp-type - type of link (L3VPN, L2VPN, combined)

tdsvc-tp-support-attributes - list of top-level domain-name attributes this links supports.

3. Yang Data Model

<CODE BEGINS> file "ietf-i2rs-service-topology@2016-02-q0.yang"

```
 module ietf-i2rs-service-topology{
  namespace "urn:ietf:params:xml:ns:yang:ietf-i2rs-service-topology";
  prefix i2rs-st;

    import ietf-inet-types {
    prefix inet;
  }

   import ietf-network {
     prefix nw;
   }
   import ietf-network-topology {
     prefix "nt";
   }

    organization "IETF";
          contact
     "email: shares@ndzh.com;
      email: linda.dunbar@huawei.com;
         ";

   description
     "This module defines a model for the service topology.
          This service model imports
          - ietf-network and ietf-network-topology from
             draft-ietf-i2rs-yang-network-topo-02.txt,
          - ietf-routing from draft-ietf-netmod-routing-cfg,
          - ietf-l3vpn-svc from
              draft-ietf-l3sm-l3vpn-service-model.
                  (not defined yet )
          ";

  revision 2016-02-12 {
    description
      "Version 1 - initial version;
           Version 2 - yang format fixed
           Version 3 - errro in xml file
           version 4 - remove next-hops attribute.
           version 5- links to top-level attributes.
            version 6 - Remove extra parameters.";

     reference "draft-hares-i2rs-service-topo-dm-05.txt";
  }
```

```
        identity svc-topo-flag-identity {
          description "Base type for svc flags";
          }
        identity l3vpn-svc-topo {
          base svc-topo-flag-identity;
          description "L3VPN service type";
         }
        identity l2vpn-svc-topo {
          base svc-topo-flag-identity;
          description "L2VPN service type";
         }
         identity EVPN-svc-topo {
          base svc-topo-flag-identity;
          description "EVPN service type";
         }
          identity Seamless-MPLS-svc-topo {
          base svc-topo-flag-identity;
          description "Seamless MPLS service type";
         }
          identity Etree-svc-topo {
          base svc-topo-flag-identity;
          description "Seamless MPLS service type";
         }
          identity I2rs-svc-topo {
          base svc-topo-flag-identity;
          description "I2RS create service topo";
         }

        identity svc-tp-type {
          description "Base type for service
            termination-point type flags";
          }
        identity svc-tp-type-service {
          base svc-tp-type;
          description "service type";
         }
        identity svc-tp-type-ip {
          base svc-tp-type;
          description "service IP";
         }
        identity svc-tp-type-unnum {
                  base svc-tp-type;
          description "service unnumbered link";
         }


            identity svc-link-type {
             description "Base type for composite
```

```
                  service link attribute flags";
              }
                  identity svc-link-ip-te {
                    base svc-link-type;
                    description "service link
                    that support IP traffic engineering";
                  }

                  identity svc-link-ip-multicast {
                    base svc-link-type;
                    description "service link that
                    supports IP multicast.";
                  }

             identity tdsvc-support-identity {
           description "Base type for svc flags";
           }

                  identity td-L3sm-hub-spoke {
                      base tdsvc-support-identity;
           description "Supports L3SM hub-spoke";
           }
               identity td-L3sm-hub-spoke-disjoint {
                       base tdsvc-support-identity;
           description "Supports L3SM hub-spoke disjoint";
           }

                  identity td-L3sm-any-any {
                      base tdsvc-support-identity;
           description "Supports L3SM any-any";
           }


         grouping svc-combo-network-type {
           description "Identify the topology type to be
                    composite service topology.";
           container svc-combo-network {
             presence "indicates Service layer Network";
             description
             "The presence of the container node indicates
              Service layer which combines networks
                          L3VPN, L2VPN, and others";
                    }
         }


        grouping service-topology-attributes {
          leaf name {
```

```
                type string;
          description "name of service
          topology";
                }
        leaf description {
                type string;
                description "description
                of service attribute";
           }
               leaf composite-flag {
          type identityref {
             base svc-topo-flag-identity;
                 }
          description "other topologies
          this topology is configured to
           be a composite of
             (L3VPN, L2VPN, I2RS only)";
        }

                list tdsvc-supports-attributes {
                   key tdsvc-attr-name;
                   leaf tdsvc-attr-name {
                    type string;
                        description "top-down
                        service support attribute name";
                   }
                  leaf tdsvc-supports-attribute {
            type identityref {
             base tdsvc-support-identity;
                     }
                     description "top-down service
                    attribute this topology supports.";

                   }
                   description "supporting top-down
          service attributes. ";
                }
                description  "Group of attributes for
             service topology";

      }


      grouping node-svc-attribute {
        leaf c-svc-node-name{
          type inet:domain-name;
          description "Domain name for node";
        }
```

```
            leaf c-svc-flag {
              type identityref {
                 base svc-topo-flag-identity;
                      }
              description "virtual network
               node can be composite of the
               topologies list
               (L3VPN, L2VPN, I2RS only)";
            }

                    list tdsvc-node-supports-attributes {
                       key tdsvc-node-attr-name;
                       leaf tdsvc-node-attr-name {
                            type string;
                            description "name of top-down
                             service attribute ";
                        }
                        leaf tdsvc-node-supports-attribute {
                     type identityref {
                      base tdsvc-support-identity;
                              }
                              description "top-down service
                             attribute this topology supports.";

                       }
                       description "list of top-down service
                       attributes this node supports";
                     }

                 description
                 "grouping of composite flag";
          }

            grouping service-link-attributes {
            leaf c-svc-link-name {
                      type string;
              description "name of
                service link";
                    }
            leaf c-svc-link-type {
              type identityref {
                 base svc-link-type;
                      }
              description "other topologies
                this link is current a
                             composite of
                (L3VPN, L2VPN, I2RS only)";
            }
```

```
          leaf c-svc-link-metric {
            type uint32;
            description "link metric
                      for servicest to
                          allow TE loading at composite
                          service level";
          }
                  list tdsvc-link-supports-attributes {
                     key tdsvc-link-attr-name;
                     leaf tdsvc-link-attr-name {
                      type string;
                          description "top-down
                          service support attribute name";
                     }
                    leaf c-svc-link-td-support-attribute {
               type identityref {
                base tdsvc-support-identity;
                        }
                       description "top-down service
                        attribute this link supports.";
                     }
                    description "list of service level
                     link attributes";
                  }
              description "grouping of
                   service link attribute";
        }

     grouping service-termination-point-attributes {
          leaf svc-tp-name {
                   type string;
                   description "name of service
                    termination point";
                  }
          leaf svc-tp-type {
              type identityref {
                base svc-topo-flag-identity;
                             }
            description "other topologies
              this link termination point is
              part of (L3VPN, L2VPN,
                      or I2RS only)";
                    }
          list tdsvc-tp-support-attributes {
                     key tdsvc-tp-attr-name;
                     leaf tdsvc-tp-attr-name {
                      type string;
                          description "top-down
```

```
                        service support attribute name";
                     }
                    leaf tdsvc-tp-support-attribute {
               type identityref {
                base tdsvc-support-identity;
                       }
                       description "top-down service
                      attribute this link supports.";
                     }
                    description "list of service level
                     link attributes";
                  }
              description "grouping of
                   service link attribute";

         }


             augment "/nw:networks/nw:network/nw:network-types" {
           uses svc-combo-network-type;
                  description
                  "augment the network-tpyes with
                  the service-topology-types grouping";
                  }

                  augment "/nw:networks/nw:network" {
            when "nw:network-types/svc-combo-network" {
              description
                "Augmentation parameters apply only for
                            service network with bottoms up topology";
            }
            description
              "Augment with combo service
                       topology attributes";
            uses service-topology-attributes;
           }

       augment "/nw:networks/nw:network/nw:node"{
              when "nw:network-types/svc-combo-network" {
              description
                "Augmentation parameters apply only for
                            service network with bottoms up topology";
            }
            uses node-svc-attribute;
                description
               "augment the node with the node-svc-attribute";
             }
```

```
           augment "/nw:networks/nw:network/nt:link" {
               when "nw:network-types/svc-combo-network" {
              description
                "Augmentation parameters apply only for
                            service network with bottoms up topology";
           }
                   uses service-link-attributes;
                   description
                   "augment the link with
                   service-link-attributes";
           }
       augment "/nw:networks/nw:network/nw:node/nt:termination-point"{
                when "nw:network-types/svc-combo-network" {
              description
                "Augmentation parameters apply only for
                            service network with bottoms up topology";
            }
                   uses service-termination-point-attributes;
                   description
                   "augment the termination-point with
                   service-termination-point-attributes";
           }

   } // module i2rs-service-topology
   } // module i2rs-service-topology
```

<CODE ENDS>

4. IANA Considerations

TBD

5. Security Considerations

TBD

6. References

6.1. Normative References

[I-D.ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Varga, R., Tkacik, T., Bahadur, N., and H. Ananthakrishnan, "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo-02 (work in progress), December 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

6.2. Informative References

[I-D.ietf-i2rs-yang-l3-topology]
Clemm, A., Medved, J., Varga, R., Tkacik, T., Liu, X., Bryskin, I., Guo, A., Ananthakrishnan, H., Bahadur, N., and V. Beeram, "A YANG Data Model for Layer 3 Topologies", draft-ietf-i2rs-yang-l3-topology-01 (work in progress), December 2015.

[I-D.ietf-l3sm-l3vpn-service-model]
Litkowski, S., Shakir, R., Tomotaki, L., and K. D'Souza, "YANG Data Model for L3VPN service delivery", draft-ietf-l3sm-l3vpn-service-model-02 (work in progress), December 2015.

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com

Linda Dunbar
Huawei
USA

Email: linda.dunbar@huawei.com

I2RS WG
Internet-Draft
Intended status: Informational
Expires: January 7, 2016

D. Migault, Ed.
J. Halpern
Ericsson
S. Hares
Huawei

July 6, 2015

I2RS Environment Security Requirements
draft-mglt-i2rs-security-environment-reqs-00

Abstract

This document provides environment security requirements for the I2RS architecture. Environment security requirements are independent of the protocol used for I2RS. As a result, the requirements provided in this document are intended to provide good security practise so I2RS can be securely deployed and operated.

These security requirements are designated as environment security requirements as opposed to the protocol security requirements described in [I-D.hares-i2rs-auth-trans]. The reason to have separate document is that protocol security requirements are intended to help the design of the I2RS protocol.

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 January 7, 2016.

Copyright Notice

Copyright (c) 2015 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 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. Introduction

This document addresses security considerations for the I2RS architecture. These requirements are also designated as environment security requirements. These security requirements are independent from the I2RS protocol used, and as such do not address requirements the I2RS protocol is expected to meet. The security requirement

provided in this document are intended to provide guidance and security principles to guarantee the stability of the I2RS architecture. This documents provides an analysis of the security issues of the I2RS architecture beyond those already listed in [I-D.ietf-i2rs-architecture].

On the other hand, security requirements for the I2RS protocol design are described in a separate document [I-D.hares-i2rs-auth-trans].

Even though I2RS is mostly concerned by the interface between the I2RS Client and the I2RS Agent, the security recommendations must consider the entire I2RS architecture, specifying where security functions may be hosted, and what should be met so to address any new attack vectors exposed by deploying this architecture. In other words, security has to be considered globally over the complete I2RS architecture and not only on the interfaces.

I2RS architecture depicted in [I-D.ietf-i2rs-architecture] describes the I2RS components and their interactions to provide a programmatic interface for the routing system. I2RS components as well as their interactions have not yet been considered in conventional routing systems. As such it introduces a need to interface with the routing system designated as I2RS plane in this document.

This document is built as follows. Section 3 describes how the I2RS plane can be contained or isolated from existing management plane, control plane and forwarding plane. The remaining sections of the document focuses on the security within the I2RS plane. Section 4 analyzes how the I2RS Authentication Authorization and Access Control (I2RS AAA) can be deployed throughout the I2RS plane in order to only grant access to the routing system resources to authorized components with the authorized privileges. This also includes providing a robust communication system between the components. Then, Section 5 details how I2RS keeps applications isolated one from another and do not affect the I2RS components. Applications may be independent, with different scopes, owned by different tenants. In addition, they modify the routing system that may be in an automatic way.

The reader is expected to be familiar with the [I-D.ietf-i2rs-architecture]. The document provides a list of environment security requirements. Motivations are placed before the requirements are announced.

[QUESTION: Some suggested to use system instead of plane. Which is the more appropriate terminology?]

2. Terminology and Acronyms

- Environment Security Requirements :

- I2RS plane : The environment the I2RS process is running on

- I2RS user : The user of the I2RS client software or system.

- I2RS AAA : The AAA information passed regarding I2RS Agents or Clients.

- I2RS Client AAA : The AAA process on the system the I2RS client is operating on.

- I2RS Agent AAA : The AAA process on the system the I2RS Agent is on.

3. I2RS Plane Isolation

Isolating the I2RS plane from other network plane, such as the control plane, is foundational to the security of the I2RS environment. Clearly differentiating I2RS components from the rest of the network protects the I2RS components from vulnerabilities in other parts of the network, and protect other systems vital to the health of the network from vulnerabilities in the I2RS plane. Separating the I2RS plane from other network control and forwarding planes is similar to the best common practice of containerizing software into modules, and defense in depth in the larger world of network security.

That said the I2RS plane cannot be considered as completely isolated from other planes, and interactions should be identified and controlled. Follows a brief description on how the I2RS plane positions itself in regard to the other planes. The description is indicative, and may not be exhaustive.

3.1. I2RS plane and management plane

The I2RS plane and the management plane both interact with several common elements on forwarding and packet processing devices. [I-D.ietf-i2rs-architecture] describes several of these interaction points such as the local configuration, the static system state, routing, and signalling. Because of this potential overlaps, a routing resource may be accessed by different means (APIs, applications) and different planes. To keep these overlaps under control, one could either control the access to these resources with northbound APIs for example. Northbound APIs are provided to limit the scope of the applications toward the routing resources. In our

case, the northbound API may be provided for the I2RS applications by the I2RS Client as well as to the management plane. In case conflicting overlaps cannot be avoided, and routing resource can be accessed by both the management plane and the I2RS plane, then, they should be resolved in a deterministic way.

On the northbound side, there must be clear protections against the I2RS system "infecting" the management system with bad information, or the management system "infecting" the I2RS system with bad information. The primary protection in this space is going to need to be validation rules on the speed of information flow, value limits on the data presented, and other protections of this type.

On the conflicting side/issues, there should be clear rules about which plan’s commands win in the case of conflict in order to prevent attacks where the two systems can be forced to deadlock.

3.2. I2RS plane and forwarding plane

Applications hosted on I2RS Client belongs to the I2RS plane, but remains hard to remain constrained into the I2RS plane, and even within the I2RS plane to have a limited scope.

Applications using I2RS are part of the I2RS plane but may also interact with other components outside the I2RS plane. A common example may be an application uses I2RS to configure the network according to security or monitored events. As these events are monitored on the forwarding plane and not the I2RS plane, the application breaks plane isolation.

In addition, applications may communicate with multiple I2RS Clients; as such, any given application may have a broader view of the current and potential states of the network and the I2RS plane itself. Because of this, any individual application could be an effective attack vector against the operation of the network, the I2RS plane, or any plane with which the I2RS plane interacts. There is little the I2RS plane can do to validate applications with which it interacts, other than to provide some broad general validations against common misconfigurations or errors. As with the separation between the management plane and the I2RS plane, this should minimally take the form of limits on information accepted, limits on the rate at which information is accepted, and rudimentary checks against intentionally formed routing loops or injecting information that would cause the control plane to fail to converge. Other forms of protection may be necessary.

3.3. I2RS plane and Control plane

The network control plane consists of the processes and protocols that discover topology, advertise reachability, and determine the shortest path between any location on the network and any destination. It is not anticipated there will be any interaction between the on-the-wire signalling used by the control plane. However, in some situations the I2RS system could modify information in the local databases of the control plane. This is not normally recommended, as it can bypass the normal loop free, loop free alternate, and convergence properties of the control plane. However, if the I2RS system does directly inject information into these tables, the I2RS system should ensure that loop free routing is preserved, including loop free alternates, tunnelled interfaces, virtual overlays, and other such constructions. Any information injected into the control plane directly could cause the control plane to fail to converge, resulting in a complete network outage.

3.4. Recommendations

To isolate I2RS transactions from other planes, it is recommended that:

REQ 1: Application-to-routing system resources communications should use an isolated communication channel. Various level of isolation can be considered. The highest level of isolation may be provided by using a physically isolated network. Alternatives may also consider logical isolation; for example by using vLAN. Eventually, in virtual environment that shares a common infrastructure, encryption may also be used as a way to enforce isolation.

REQ 2: The interface (like the IP address) used by the routing element to receive I2RS transactions should be a dedicated physical or logical interface. As previously, mentioned a dedicated physical interface may contribute to a higher isolation, however logical isolation be also be considered for example by using a dedicated IP address or dedicated port.

REQ 3: The I2RS Agent validates data to ensure injecting the information will not create a deadlock with any other system, nor will it create a routing loop, nor will it cause the control plane to fail to converge.

When the I2RS Agent performs an action on a routing element, the action is performed via process(es) associated to a system user . In a typical UNIX system, the user is designated with a user id (uid)

and belong to groups designated by group ids (gid). These users are dependent of the routing element's operation system and are designated I2RS System Users. Some implementation may use a I2RS System User for the I2RS Agent that proxies the different I2RS Client, other implementations may use I2RS System User for each different I2RS Clients.

REQ 4: I2RS Agent should have permissions separate from any other entity (for example any internal system management processes or CLI processes).

I2RS resource may be shared with the management plane and the control plane. It is hardly possible to prevent interactions between the planes. I2RS routing system resource management is limited to the I2RS plane. As such, update of I2RS routing system outside of the I2RS plane may be remain unnoticed unless explicitly notified to the I2RS plane. Such notification is expected to trigger synchronization of the I2RS resource state within each I2RS component. This guarantees that I2RS resource are maintained in a coherent state among the I2RS plane. In addition, depending on the I2RS resource that is updated as well as the origin of the modification performed, the I2RS Authentication Authorization and Access Control policies (I2RS AAA) may be impacted. More especially, a I2RS Client is more likely to update an I2RS resources that has been updated by itself, then by the management plane for example.

REQ 5: I2RS plane should be informed when a routing system resource is modified by a user outside the I2RS plane access. The notification is not expected to flood the I2RS plane. Instead, notification is expected to be provided to the I2RS components interacting, configuring or monitoring the routing system resource. The notification is at least provided by the I2RS Agent to the various I2RS Client, but additional mechanisms might eventually be required so I2RS Client can relay the notification to the I2RS applications. This is designated as "I2RS resource modified out of I2RS plane". This requirements is also described in section 7.6 of [I-D.ietf-i2rs-architecture] for the I2RS Client. This document extends the requirement to the I2RS plane, in case future evolution of the I2RS plane.

REQ 6: I2RS plane should define an "I2RS plane overwrite policy". Such policy defines how an I2RS is able to update and overwrite a resource set by a user outside the I2RS plane. Such hierarchy has been described in section 6.3 and 7.8 of [I-D.ietf-i2rs-architecture]

4. I2RS Authentication and Authorization Access Policy for routing system resources

This section details the I2RS Authentication and Authorization Access Policy (I2RS AAA) associated to the routing system resources. These policies only apply within the I2RS plane for I2RS users.

4.1. I2RS AAA architecture

Applications access to routing system resource via numerous intermediaries nodes. The application communicates with an I2RS Client. In some cases, the I2RS Client is only associated to a single application, but the I2RS Client may also act as a broker. The I2RS Client, then, communicates with the I2RS Agent that may eventually access the resource.

The I2RS Client broker approach provides scalability to the I2RS architecture as it avoids that each Application be registered to the I2RS Agent. Similarly, the I2RS AAA should be able to scale numerous applications.

REQ 7: I2RS AAA should be performed through the whole I2RS plane. I2RS AAA should not be enforced by the I2RS Agent only within the routing element. Instead, the I2RS Client should enforce the I2RS Client AAA against applications and the I2RS Agent should enforce the I2RS Agent AAA against the I2RS Client. Note that I2RS Client AAA is not in the scope of the I2RS architecture [I-D.ietf-i2rs-architecture], which exclusively focuses on the I2RS Agent AAA.

This results in a layered and hierarchical I2RS AAA. An application will be able to access a routing system resource only if both the I2RS Client is granted access by the I2RS Agent AAA and the application is granted access by the I2RS Client AAA.

REQ 8: In case the AAA on the I2RS Client system or the AAA on the I2RS Agent system does not grant the access to a routing system resource, the Application should be able to define the I2RS AAA that generated this reject, as well as the reason. More specifically, the I2RS Agent may reject the request based on the I2RS Client privileges, and the I2RS Client should return a message to the application, indicating the I2RS Client does not have enough privileges. Similarly, if the I2RS Client does not grant the access to the application, the I2RS Client should also inform the application. The error message returned should be for example: "Read failure: you do not have the read permission", "Write failure: you do not have write permission" or "Write failure: resource

accessed by someone else". Note that although multiple rejects may occur, that is both by the I2RS Client and the I2RS Agent, only the first reject from the I2RS Client should be mandatory. This requirement has been written in a generic manner as it concerns various interactions: interactions between the application and the I2RS Client, interactions between the I2RS Client and the I2RS Agent. In the latest case, the requirement is part of the protocol security requirements addressed by [I-D.hares-i2rs-auth-trans].

In order to limit the number of access request that result in an error, each component should be able to retrieve the global I2RS AAA policies that applies to it. This subset of rules is designated as the "I2RS AAA component's subset policies". As they are subject to changes, a dynamic synchronization mechanism should be provided. This requirements is expressed by various sub requirements. This may be considered as a protocol security requirement when the I2RS Client and the I2RS Agent are involved. However, for completeness of the security requirements over the I2RS environment, they are are still listed below.

REQ 9: The I2RS Client should be able to request for its I2RS AAA Agent subset policies to the I2RS Agent AAA, so to limit forwarding unnecessary queries to the I2RS Agent.

REQ 10: The I2RS Client should be able to be notified when its I2RS AAA Agent subset policies have been updated.

Similarly, for the application

REQ 11: The Application may be able to request for its I2RS AAA Client subset policies, so to limit forwarding unnecessary queries to the I2RS Client.

REQ 12: The Application may be able to subscribe a service that provides notification when its I2RS AAA Client subset policies have been updated.

I2RS AAA should be appropriately be balanced between the I2RS Client and the I2RS Agent which can be illustrated by two extreme cases:

- 1) I2RS Clients are dedicated to a single Application: In this case, it is likely that I2RS AAA is enforced only by the I2RS Agent AAA, as the I2RS Client is likely to accept all access request of the application. However, it is recommended that even in this case, I2RS Client AAA is not based on an "Allow anything from application" policy, but instead the I2RS Client specifies accesses that are enabled. In addition, the I2RS

Client may sync its associated I2RS Agent AAA with the I2RS Agent to limit the number of refused access requests being sent to the I2RS Agent. The I2RS Client is expected to balance pro and cons between sync the I2RS Agent AAA and simply guessing the access request to the I2RS Agent.

- 2) A single I2RS Client acts as a broker for all Applications: In the case the I2RS Agent has a single I2RS Client. Such architecture results in I2RS Client with high privileges, as it sums the privileges of all applications. As end-to-end authentication is not provided between the Application and the I2RS Agent, if the I2RS Client becomes corrupted, it is possible for the malicious application escalates its privileges and make the I2RS Client perform some action on behalf of the application with more privileges. This would not have been possible with end-to-end authentication. In order to mitigate such attack, the I2RS Client that acts as a broker is expected to host application with an equivalent level of privileges.

REQ 13: The I2RS AAA should explicitly specify accesses that are granted. More specifically, anything not explicitly granted -- the default rule-- should be denied.

In order to keep the I2RS AAA architecture as distributed as possible,

REQ 14: I2RS Client should be distributed and act as brokers for applications that share roughly similar permissions. This avoids ending with over privileges I2RS Client compared to hosted applications and thus discourages applications to perform privilege escalation within an I2RS Client.

REQ 15: I2RS Agent should be avoided being granted over privileges regarding to their authorized I2RS Client. I2RS Agent should be shared by I2RS Client with roughly similar permissions. More explicitly, an I2RS Agent shared between n I2RS Clients that are only provided read access to the routing system resources. This I2RS Agent does not need to perform any write access, and so should not be provided these accesses. Suppose an I2RS Client requires write access to the resources. It is not recommended to grant the I2RS Agent the write access in order to satisfy a unique I2RS Client. Instead, the I2RS Client that requires write access should be connected to a I2RS Agent that is already shared by I2RS Client that requires a write access.

4.2. I2RS Agent AAA

The I2RS Agent AAA restricts the routing system resource access to authorized components. Possible access policies may be none, read or write. The component represents the one originating the access request. The origin of the query is always an application. However, the I2RS Agent may not be able to authenticate the application as the I2RS Client may act as a broker. Similarly, multiple I2RS Agents may be used, and different access privilege may be provided depending on the I2RS Agent used. As a result, the origin of the query may be represented in multiple ways, and each way be may associated to a specific AAA. In some cases, the origin of the I2RS query is only represented by the I2RS Client, and the I2RS Agent does not have any means to associate the request to an application. In some cases, the I2RS Agent may identify the application by the I2RS Client or via other means. In addition, there is not a single way to represent an I2RS Client, and multiple identities may be used (FQDN, public key, certificates)

REQ 16: I2RS Agent AAA may use various ways to represent the origin of the access request of a routing system resource. However, representation of the origin should be based on information that can be authenticated. The I2RS Client, optionally the I2RS Agent in case of multiple I2RS Agents go into this category. On the hand, unless some additional means for authentication have been provided, the secondary identity used to tag the application as defined in [I-D.ietf-i2rs-architecture] should not be considered.

The I2RS Agent AAA may evolve over time as resource may also be updated outside the I2RS plane. Similarly, a given resource may be accessed by multiple I2RS users within the I2RS plane. Although this is considered as an error, depending on the I2RS Client that performed the update, the I2RS may accept or refuse to overwrite the routing system resource.

REQ 17: Each routing system resource updated by a I2RS Agent should be informed of the component that performed the last update. On an environment perspective, the I2RS Agent MUST be aware when the resource has been modified by a component outside the I2RS plane, as well as the priority associated to this component towards the I2RS plane. Similar requirements exist for components within the I2RS plane, but belongs to the protocol security requirements.

REQ 18: the I2RS Agent should have a "I2RS Agent overwrite Policy" that indicates how the originating components can be prioritized. This requirements is also described in section

7.6 of [I-D.ietf-i2rs-architecture]. Similar requirements exist for components within the I2RS plane, but belongs to the protocol security requirements.

4.3. I2RS Client AAA

The I2RS Client AAA works similarly to the I2RS Agent AAA. The main difference is that components are applications. As a result,

REQ 19: The I2RS Client should be able to authenticate its application.

In case, no authentication mechanisms have being provided between the I2RS Client and the application, then I2RS Client may not act as broker, and be instead dedicated to a single application. By doing so, application authentication may rely on the I2RS authentication mechanisms between the I2RS Client and the I2RS Agent. On the other hand, although this is not recommended, the I2RS AAA is only enforced by the I2RS Agent AAA.

4.4. I2RS AAA Security Domain

I2RS AAA enforcement should not remain local, and the security domain resulting from this enforcement must be extended throughout the network. More specifically I2RS AAA policies enforced on one point remain reliable for another point as long as the communication between the two points is reliable too. This means communications should remain:

- 1) Available at any time, and it should be robust to potential attacks, or misbehaviors.

- 2) Trusted.

These characteristics are mostly the goal of a security transport layer. As such:

REQ 20: I2RS communications should be based on a security transport layer.

4.4.1. Available I2RS Communication Channel

Communication is considered available if and only if all components are available as well as the communication channel itself. In order to maintain it available here are the considered aspects:

- 1) Make communication robust to DoS by design

- 2) Provide active ways to mitigate an DoS attack

- 3) Limit damages when a DoS event occurs

Protocols used to communicate between components should not provide means that would result in a component's resource exhaustion.

If non secure transport layer is used, when possible, protocols that do not implement any mechanisms to check the origin reachability should be avoided (like UDP). Instead, if possible, protocols like TCP or SCTP with origin reachability verification should be preferred.

Anti DoS mechanisms should also be considered at other layers including the application layer. In our case the application layer may be the I2RS protocols itself or the applications that are using the I2RS protocol. More specifically, it should be avoided to perform actions that generate heavy computation on a component. At least the component should be able to post-pone and re-schedule the action. Similarly, DoS by amplification should be avoided, and special attention should be given to small access request that generate massive network traffic without any control. An example of asymmetric dialogue could be the subscription of information streams like prefix announcement from OSPF. In addition, some service may also provide the ability to redirect these streams to a third party. In the case of information stream, registration by an I2RS Client may provide the possibility to redirect the stream on a shared directory, so it can be accessed by multiple I2RS Clients, while not flooding the network. In this case, special attention should be provided so the shared directory can agree based on its available resources the service subscription by the I2RS Client. Otherwise, the shared directory may become overloaded.

REQ 21: Resources (CPU, memory or bandwidth) allocated to each components should be agreed between the component requesting the resource and the component providing the resources.

Components should be able to control the computing resource they allocate to each other components, or each actions. Based on available resource, requests should be differed, or returned an error.

REQ 22: I2RS Client and I2RS Agent should implement mechanisms within their environments to mitigate DoS attacks.

One alternative way to mitigate a DoS attack or event is to limit the damages when resource exhaustion happens. This can be done by appropriately group or ungroup applications. For example, critical

applications may not share their I2RS Client with multiple other Applications. This limits the probability of I2RS Client failure for the critical application. Similarly, I2RS Agent may also be selective regarding their I2RS Client as well as to the scope of their routing system resources. In fact some, some I2RS Client may be less trusted than others and some routing system resource access may be more sensitive than the others. Note that trust of an I2RS Client is orthogonal to authentication and rather involves, for example, the quality of the hosted Applications.

REQ 23: Application, I2RS Client and I2RS Agent should be distributed among the I2RS Plane to minimize the impact of a failure.

Even though this should be considered, it does not address the high availability issue. In order to reduce the impact of a single I2RS Client failure, remote applications may load balance their access request against multiple I2RS Clients. Non remote I2RS Client or I2RS Agent are bound the system hosting the application or to the routing element. This makes high availability be provided by the system, and thus implementation dependent.

REQ 24: I2RS Client should provide resilient and high availability for the hosted applications.

4.4.2. Trusted I2RS Communications Channel

Section 2.2 of [I-D.hares-i2rs-auth-trans] provides requirements to establish a secure communication between the I2RS Agent and the I2RS Client. These requirements can be generalized to any I2RS communications within the I2RS plane. This may include for example a remote application connected to the I2RS Client.

5. I2RS Application Isolation

A key aspect of the I2RS architecture is the network oriented application. As these application are supposed to be independent, controlled by independent and various tenants. In addition to independent logic, these applications may be malicious. Then, these applications introduce also programmability which results in fast network settings.

The I2RS architecture should remain robust to these applications and make sure an application does not impact the other applications. This section discusses both security aspects related to programmability as well as application isolation in the I2RS architecture.

5.1. Robustness toward programmability

I2RS provides a programmatic interface in and out of the Internet routing system. This feature, in addition to the global network view provided by the centralized architecture comes with a few advantages in term of security.

The use of automation reduces configuration errors. In addition, this interface enables fast network reconfiguration. Agility provides a key advantage in term of deployment as side effect configuration may be easily addressed. Finally, it also provides facilities to monitor and mitigate an attack when the network is under attack.

On the other hand programmability also comes with a few drawbacks. First, applications can belong to multiple tenants with different objectives. This absence of coordination may result in unstable routing configurations such as oscillations between network configurations, and creation of loops for example. A typical example would be an application monitoring a state and changing its state. If another application performs the reverse operation, the routing system may become unstable. Data and application isolation is expected to prevent such situations to happen, however, to guarantee the network stability, constant monitoring and error detection are recommended to be activated.

REQ 25: I2RS should monitor constantly parts of the system for which clients have requested notification. It should also be able to detect components that lead the routing system in an unstable state.

5.2. Application Isolation

5.2.1. DoS

Requirements for robustness to Dos Attacks have been addressed in the Communication channel section [I-D.ietf-i2rs-architecture].

The I2RS interface is used by application to interact with the routing states. As the I2RS Agent is shared between multiple applications, one application can prevent an application by performing DoS or DDoS attacks on the I2RS Agent or on the network. DoS attack targeting the I2RS Agent would consist in providing requests that keep the I2RS Agent busy for a long time. This may involve heavy computation by the I2RS Agent for example to blocking operations like disk access. In addition, DoS attacks targeting the network may use specific commands like monitoring stream over the network. Then, DoS attack may be also targeting the application

directly by performing reflection attacks. Such an attack could be performed by indicating the target application as the target for some information like the listing of the RIB. Reflection may be performed at various levels and can be based on the use of UDP or at the service level like redirection of information to a specific repository.

REQ 26: In order to prevent DoS, it is recommended the I2RS Agent controls the resources allocated to each I2RS Clients. I2RS Client that acts as broker may not be protected as efficiently against these attacks unless they perform resource controls themselves of their hosted applications.

REQ 27: I2RS Agent does not make response redirection possible unless the redirection is previously validated and agreed by the destination.

REQ 28: avoid the use of underlying protocols that are not robust to reflection attacks.

5.2.2. Application Control

Requirements for Application Control have been addressed in the I2RS plane isolation as well as in the trusted Communication Channel sections.

Applications use the I2RS interface in order to update the routing system. These updates may be driven by behavior on the forwarding plane or any external behaviors. In this case, correlating observation to the I2RS traffic may enable to derive the application logic. Once the application logic has been derived, a malicious application may generate traffic or any event in the network in order to activate the alternate application.

REQ 29: Application logic should remain opaque to external listeners. Application logic may be partly hidden by encrypting the communication between the I2RS Client and the I2RS Agent. Additional ways to obfuscate the communications may involve sending random messages of various sizes. Such strategies have to be balanced with network load. Note that I2RS Client broker are more likely to hide the application logic compared to I2RS Client associated to a single application.

6. Security Considerations

The whole document is about security.

7. Privacy Considerations

8. IANA Considerations

9. Acknowledgments

We would like to thanks Russ White for its review and editorial contributions.

10. References

10.1. Normative References

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

10.2. Informative References

[I-D.ietf-i2rs-architecture]
Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", draft-ietf-i2rs-architecture-09 (work in progress), March 2015.

[I-D.hares-i2rs-auth-trans]
Hares, S., "I2RS Security Related Requirements", draft-hares-i2rs-auth-trans-03 (work in progress), June 2015.

Authors' Addresses

Daniel Migault (editor)
Ericsson
8400 boulevard Decarie
Montreal, QC H4P 2N2
Canada

Phone: +1 514-452-2160
Email: daniel.migault@ericsson.com

Joel Halpern
Ericsson

Email: Joel.Halpern@ericsson.com

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com

I2RS Working Group
Internet-Draft
Intended status: Standards Track

Xian Zhang
Baoquan Rao
Huawei
Xufeng Liu
Ericsson

Expires: September 9, 2015 March 9, 2015

A YANG Data Model for Layer 1 Network Topology

draft-zhang-i2rs-l1-topo-yang-model-01.txt

Abstract

This draft describes a YANG data model to manipulate the topologies of a layer 1 network. It is independent of data plan technologies and control plane protocols. It can be augmented to include technology-specific data, such as for Optical Transport Networks (OTN).

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 9, 2015.

Copyright Notice

Copyright (c) 2015 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 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. Introduction

This document defines a data model of a layer one network topology, using YANG [RFC6020]. The model can be used by an application via the I2RS interface [draft-ietf-i2rs-architecture], in the following ways (but not limited to):

o to obtain a whole view of the network topology information of its interest;

o to receive notifications with regard to the information of the change of the network topology of its interest;

o to enforce the establishment/update of a network topology with the characteristic specified in the data model;

This model is confined to describe layer 1 networks, but it is data plane technology independent and can be augmented to specify the topology for networks such as Optical Transport networks (OTN), Synchronous Digital Network/ (SDH/SONET) DWDM (Dense Wavelength Division Multiplexing).

[Editor's Note: The authors are aware that there are other drafts closely relating to this draft. Coordination works have been undergoing to get these drafts aligned. The authors are working on obtaining layer one topology by augmenting the data model proposed in draft-clemm-i2rs-yang-network-topo in the next version of this draft.]

2. 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].

3. Terminology and Notations

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in the YANG data tree presented later in this draft is defined in [ietf-netmod-rfc6087bis]. They are provided below for reference.

- Brackets "[" and "]" enclose list keys.

- Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).

- Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and 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.

4. YANG Data Model for Layer 1 Topology

4.1. YANG Tree

```
module: ietf-layer1topology
   +--rw layer-one-topology
      +--rw topology* [topology-id]
         +--rw topology-id            topology-id
         +--rw name?                  string
         +--rw supporting-topology* [topo-ref]
         |  +--rw topo-ref    leafref
         +--rw node* [node-id]
         |  +--rw node-id               node-id
         |  +--rw interface* [interface-id]
         |  |  +--rw interface-id             interface-id
         |  |  +--rw interface-name?          if:interface-state-ref
         |  |  +--rw adaptation-capability
         |  +--rw connectivity-matrix* [id]
         |     +--rw id               uint32
         |     +--rw type?            enumeration
         |     +--rw in-interface* [interface-ref]
         |     |  +--rw interface-ref    leafref
         |     +--rw out-interface* [interface-ref]
         |     |  +--rw interface-ref    leafref
         |     +--rw dir?             enumeration
         +--rw link* [link-id]
            +--rw link-id            link-id
            +--rw local
            |  +--rw local-node         leafref
            |  +--rw local-interface    leafref
            +--rw remote
            |  +--rw remote-node         leafref
            |  +--rw remote-interface    leafref
            +--rw supporting-path* [supporting-path-index]
            |  +--rw supporting-path-index     uint32
            |  +--rw topo-ref?                 leafref
            |  +--rw server-path-identifier
            |  +--rw server-path-srlg
            |     +--rw srlg-values* [srlg-value]
            |        +--rw srlg-value    uint32
            +--rw attributes
               +--ro information-source?             enumeration
               +--ro credibility-preference?         uint16
               +--rw admin-status?                   enumeration
               +--ro oper-status?                    enumeration
               +--rw area-id?                        binary
               +--rw max-link-bandwidth?             decimal64
```

```
               +--rw unreserved-bandwidth* [priority]
               |  +--rw priority     uint8
               |  +--rw bandwidth?   decimal64
               +--ro distance?                          uint32
               +--rw te-metric?                         uint32
               +--rw link-protection-type?              enumeration
               +--rw switching-capability?              switching-
capabilities
               +--rw encoding?                          encoding-types
               +--rw switching-capability-specific
               +--rw srlg
                  +--rw srlg-values* [srlg-value]
                     +--rw srlg-value    uint32
notifications:
   +---n link-failure
   |  +--ro topology-id     leafref
   |  +--ro link-id         leafref
   |  +--ro admin-status?   leafref
   |  +--ro oper-status     leafref
   +---n node-failure
      +--ro topology-id    leafref
      +--ro link-id        leafref
```

4.1.1. The node and link list

The Layer One Topology module contains all the nodes and links information pertaining to a layer one network. The node is identified by the node-id, which is unique within the network. Within the nodes, all the interfaces pertaining to this node and their potential capabilities/constraints SHOULD be present. Besides this, the constraints associated with a node as a whole SHOULD also be present, such as the connectivity constraints introduced due to abstraction or due to the hardware limitations. The link is identified by the link-id, which is unique within a node. It includes the association with nodes as well as interfaces. Moreover, it includes information that is of interest to the I2RS client, for purposes, such as path computation, monitoring etc.

4.1.2. Notification

Two types of notifications are introduced: node failure and link failure.

4.2. YANG Code

<CODE BEGINS> file "l1topo.yang"

```
module ietf-layer1topology {
  yang-version 1;

  namespace
    "urn:ietf:params:xml:ns:yang:ietf-layer1topology";
  prefix "l1topo";

  import ietf-inet-types {
    prefix "inet";
  }
  import ietf-interfaces {
    prefix "if";
  }

  organization
    "Internet Engineering Task Force (IETF) I2RS WG";
  contact
    "ID-draft editor: zhang.xian@huawei.com";

  description
    "This module defines a data-plan technology/protocol
    independent Layer One topology data model.";

  revision 2015-03-09 {
    description
      "Initial version.";
    reference
      "draft-zhang-i2rs-l1-topo-yang-model-01.txt";
  }

  /*
  * Typedefs
  */

  typedef topology-id {
    type inet:uri;
    description "the identifier for a topology";
  }

  typedef node-id {
    type inet:ip-address;
    description
      "the identifier for a node";
  }

  typedef interface-id {
    type union {
```

```
      type inet:ip-address; // IPv4 or IPv6 address
      type int32;          // Un-numbered
    }
  description
    "the identifier of an interface within a node, supporting both
    numbered/unnumbered";
  }

  typedef link-id {
    type inet:ip-address; // IPv4 or IPv6 address
    description "the identifier of a link";
  }

  typedef switching-capabilities {
    type enumeration {
      enum "psc-1" {
        value 1;
        description
          "Packet-Switch Capable-1 (PSC-1)";
      }
      enum "evpl" {
        value 30;
        description
          "Ethernet Virtual Private Line (EVPL)";
      }

      enum "pbb-te"{
        value 40;
        description
          "802_1 PBB-TE";
      }

      enum "l2sc" {
        value 51;
        description
          "Layer-2 Switch Capable (L2SC)";
      }
      enum "tdm" {
        value 100;
        description
          "Time-Division-Multiplex Capable (TDM)";
      }
      enum "otn-tdm" {
        value 110;
        description
          "OTN-TDM Capable";
      }
```

```
      enum "lsc" {
        value 150;
        description
          "Lambda-Switch Capable (LSC)";
      }
      enum "fsc" {
        value 200;
        description
          "Fiber-Switch Capable (FSC)";
      }
    }

    description
      "Switching capability of an interface.
      Only a subset of the above-mentioned values are applicable
      to Layer 1 network.
      Here it is included for completeness and will later be
      updated if a base model is augmented to create layer 1
      network topology YANG data model.";

    reference
      "The definition of switching types, their values and the
      relevant RFCs can be found at:
      http://www.iana.org/assignments/gmpls-sig-parameters/gmpls
      -sig-parameters.xhtml#gmpls-sig-parameters-3";
  }

  typedef encoding-types {
    type enumeration {
      enum "packet" {
        value 1;
        description "Packet";
      }
      enum "ethernet" {
        value 2;
        description "Ethernet";
      }
      enum "pdh" {
        value 3;
        description "PDH";
      }
      enum "sdh-sonet" {
        value 5;
        description "SDH/SONET";
      }
      enum "digital-wrapper" {
        value 7;
```

```
        description "Digital Wrapper";
      }
      enum "lambda" {
        value 8;
        description "Lambda(photonic)";
      }
      enum "fiber" {
        value 9;
        description "Fiber";
      }
      enum "fiber-channel" {
        value 11;
        description "FiberChannel";
      }
      enum "oduk" {
        value 12;
        description
          "G.709 OKUk (Digital Path)";
      }
      enum "optical-channel" {
        value 13;
        description "G.709 Optical Channel";
      }
      enum "line" {
        value 14;
        description "Line (e.g., 8B/10B)";
      }
    }
    description
      "The encoding type supported by an interface or link.
      Not all encoding types are applicable to Layer one network
      nodes. They are included here for completeness and will be
      updated if a base model is available to augment
      so as to build a layer-one specific YANG data model.";
    reference
      "The definition of encoding types, their values and the
      relevant RFCs can be found at http://www.iana.org/
      assignments/gmpls-sig-parameters/gmpls-sig-parameters.xhtml#
      gmpls-sig-parameters-3";
  }

  /*
   * Groupings
  */

  grouping srlg-attribute {
    description
```

```
      "Shared Risk Link Group Attributes";
    reference
      "RFC 4203: OSPF Extensions in Support of Generalized
      Multi-Protocol Label Switching (GMPLS)";
    list srlg-values {
      key "srlg-value";
      leaf srlg-value {
        type uint32;
        description "SRLG value";
      }
    description
      "the SRLG value list";
    }
  }

  /*
   * Configuration data nodes
   */

  container layer-one-topology {
    description
      "this container holds all the inforamtion to layer
      one network. It includes one or multiple topologies";

    list topology {
      key "topology-id";

      description
        "This contains all the information to one topoogy";

      leaf topology-id {
        type topology-id;
        description "topology identifier";
      }

      leaf name {
        type string;
        description "topology name";
      }

      list supporting-topology {
        key "topo-ref";
        leaf topo-ref {
          type leafref {
            path "/layer-one-topology/topology/topology-id";
          }
          description
```

```
            "a Layer-One network might be supported by a lower
            layer network and this is a pointer to the suporting
            topology if there is one";
        }
        description "underlaying topology information";
      }

      list node {
        key "node-id";
        description "the list of nodes within the topology";

        leaf node-id {
          type node-id;
          description "node identifier";
        }

        list interface {
          key "interface-id";
          leaf interface-id {
            type interface-id;
            description "interface identifier";
          }
          leaf interface-name {
            type if:interface-state-ref;
            description
            "Name of the incoming interface.";
          }
          container adaptation-capability {
            description
            "TBD -to add for technology specific information";
          }
          description "interface list pertaining to a node";
        }

        list connectivity-matrix {
          key "id";

          description
            "This describes the connectivity contraints within
            a node in the network. It can be one matrix or a set
            of matrixes. Further details, read the reference
            provided below.";
          reference
            "https://tools.ietf.org/html/draft-ietf-ccamp-general
            -constraint-encode-16 Section 2.1";

          leaf id {
```

```
            type uint32;
            description "matrix id";
          }
          leaf type {
            type enumeration {
              enum fixed {
                value 0;
                description "Fixed";
              }
              enum dynamic {
                value 1;
                description "Dynamic/changeable";
              }
            }
            description
              "This field describes the attribute of a
              connectivity matrix, i.e., whether it is
              fixed or switched.";
          }
          list in-interface {
            key "interface-ref";

            description
              "This list describes a (sub)-set of ingoing
              interfaces within a node that may have
              connectivity constraints.
              Note: directionality may not be relevant
               and it is decided by the dir parameter.";

            leaf interface-ref {
              type leafref {
                path "/layer-one-topology/topology/node/" +
                "interface/interface-id";
              }
              description "reference to an incoming interface";
            }
          }
          list out-interface {
            key "interface-ref";

            description
              "This list describes a (sub)-set of outgoing
              interfaces within a node that may have
              connectivity constraints.
              Note: directionality may not be relevant and
              it is decided by the dir parameter.";
```

```
            leaf interface-ref {
              type leafref {
                path "/layer-one-topology/topology/node/"+
                "interface/interface-id";
              }
              description "reference to an outgoing interface";
            }
          }
          leaf dir{
            type enumeration{
              enum "uni-dir"{
                description
                "the matrix is unidirectional.";
              }
              enum "bi-dir"{
                description
                "this matrix is bidirecdtional.";
              }
            }
            description
             "the directionality attribute of a connc. matrix.";
          }
        }
      }// end of node data node

      list link {
        key "link-id";

        description "list of the links within a topology";

        leaf link-id {
          type link-id;
          description
            "remaining issue: if there is no IP addresses
             associated with this link, what would be the key?";
        }

        container local {
          description "near end information for this link";

          leaf local-node {
            type leafref {
              path "/l1topo:layer-one-topology/topology"+
              "/node/node-id";
            }
            mandatory true;
            description "refence to the local node";
```

```
          }
          leaf local-interface {
            type leafref {
              path "/l1topo:layer-one-topology/topology/node/"
              +"interface/interface-id";
            }
            mandatory true;
            description "reference to the local interface";
          }
        }
        container remote {
          description "far end information of this link";

          leaf remote-node {
            type leafref {
              path "/l1topo:layer-one-topology/topology"+
              "/node/node-id";
            }
            mandatory true;
            description "reference to the remote node";
          }
          leaf remote-interface {
            type leafref {
              path "/l1topo:layer-one-topology/topology/node/"
              + "interface/interface-id";
            }
            mandatory true;
            description "reference to the remote interface";
          }
        }
        list supporting-path {
          key "supporting-path-index";

          description
            "information pertaining to the underlying path if
             there is any";

          leaf supporting-path-index {
            type uint32;
            description "the identifer of the supporting path";
          }
          leaf topo-ref {
            type leafref {
              path "/l1topo:layer-one-topology/"+
              "topology/topology-id";
            }
            description "reference to the underlying topology";
```

```
          }
          container server-path-identifier {
            description "TBD";
          }
          container server-path-srlg {
            uses srlg-attribute;
            description "the SRLG values of the server path";
          }
        }

        container attributes {

          description "additional information of the link";

          leaf information-source {
            type enumeration {
              enum "unknown" {
                description "The source is unknown";
              }
              enum "locally-configured" {
                description "Configured TE link";
              }
              enum "ospfv2" {
                description "OSPFv2";
              }
              enum "ospfv3" {
                description "OSPFv3";
              }
              enum "isis" {
                description "ISIS";
              }
            }
            config false;

            description
              "Indicates the source of the information about
              the link. remaining issue: if configuration of
              a link is allowed, what additional types are
              needed to add?";
          }
          leaf credibility-preference {
            type uint16;
            config false;
            description "the level of credibility";
          }
          leaf admin-status {
            type enumeration {
```

```
            enum up {
              value 1;
              description "up";
            }
            enum down {
              value 2;
              description "down";
            }
            enum testing {
              value 3;
              description "testing - in some test mode.";
            }
          }
          description
            "The adminstrative state of the link.";
          reference
            "RFC2863: The Interfaces Group MIB.";
        }
        leaf oper-status {
          type enumeration {
            enum up {
              value 1;
              description "up";
            }
            enum down {
              value 2;
              description "down";
            }
            enum testing {
              value 3;
              description "testing - in some test mode";
            }
            enum unknown {
              value 4;
              description "unknown - status cannot be
              determined for some reason.";
            }
            enum dormant{
              value 5;
              description "dormant";
            }
          }
          config false;
          description
            "The current operational state of the link.";
          reference
            "RFC2863: The Interfaces Group MIB.";
```

```
          }
          leaf area-id {
            type binary {
              length 1..13;
            }
            description
              "This object indicates the area identifier of
              the IGP. If OSPF is used to advertise LSA,
              this represents an ospfArea. If IS-IS is used,
               this represents an area address.
               Otherwise, this is zero.";
            reference
              "RFC4920: Crankback Signaling Extensions for MPLS
              and GMPLS RSVP-TE.";
          }

          leaf max-link-bandwidth {
            type decimal64 {
              fraction-digits 2;
            }
            description
              "the max bandwidth supported by this link";
          }

          list unreserved-bandwidth {
            key "priority";
            max-elements "8";

            description
              "This describes the unreserved bandwidth (in
              Bytes/second) on a level basis ( level 0-7).";

            leaf priority {
              type uint8{
                range "0..7";
              }
              description "priority level";
            }
            leaf bandwidth {
              type decimal64 {
                fraction-digits 2;
              }
              description "badnwidth per priority";
            }
          }

          leaf distance {
```

```
            type  uint32;
            units "kilometers";

            config false;
            description
              "the distance this link spans.";
          }
          leaf te-metric {
            type uint32;
            description "the metric supported by the link";
          }

          leaf link-protection-type {
            type enumeration {
              enum "extra-traffic" {
                value 1;
                description "Extra traffic";
              }
              enum "unprotected" {
                value 2;
                description "unprotected";
              }
              enum "shared" {
                 value 4;
                description "Shared";
              }
              enum "1-for-1" {
                value 8;
                description "Dedicated one for one protection";
              }
              enum "1-plus-1" {
                value 16;
               description "Dedicated one plus one protection";
              }
              enum "enhanced" {
                value 32;
                description "a protection type that is
                more reliable than Dedicated 1+1,
                e.g.,4 fiber BLSR/MS-SPRING.";
              }
            }
            description
              "Link Protection Type configured for this link";
            reference
              "RFC3471: Generalized MUlti-Protocol Label
               Switching (GMPLS) Signaling Functional
               Description.";
```

```
          }

          leaf switching-capability {
            type switching-capabilities;
            description
              "the switching capability supported by the link";
          }
          leaf encoding {
            type encoding-types;
            description
              "the encoding type supported by this link.";
          }

          container switching-capability-specific {
            description
            "TBD - to add for technology specific information";
          }
          container srlg {
            uses srlg-attribute;
            description " the SRLG values of a link";
          }
        }// end of link attributes
      }// end of link leaf data node
    }
  }// end of configuring data nodes

  /*
  * notifications - only provide operational change information.
  * reply to topology/node/link creation is acked via rpc-reply.
  */

  notification link-failure {
    leaf topology-id {
      type leafref {
        path "/layer-one-topology/topology/topology-id";
      }
      mandatory true;
      description "";
    }
    leaf link-id {
      type leafref {
        path
          "/layer-one-topology/topology[topology-id="+
          "current ()/../topology-id]/link/link-id";
      }
      mandatory true;
      description "";
```

```
    }
    leaf admin-status {
      type leafref {
        path
          "/layer-one-topology/topology/link[link-id =" +
          "current()/../link-id]/attributes/admin-status";
      }
      description "";
    }
    leaf oper-status {
      type leafref {
        path
          "/layer-one-topology/topology/" +
          "link[link-id = current()/../link-id]"
          + "/attributes/oper-status";
      }
      mandatory true;
      description "";
    }
    description
      "link failure information";
  } //notification

  notification node-failure {
    leaf topology-id {
      type leafref {
        path "/layer-one-topology/topology/topology-id";
      }
      mandatory true;
      description "";
    }
    leaf link-id {
      type leafref {
        path
          "/layer-one-topology/topology[topology-id= "
          + "current ()/../topology-id]/node/node-id";
      }
      mandatory true;
      description "";
    }
    description
    "node failure information";
  } //notification
}//module

<CODE ENDS>
```

5. Security Considerations

Since the data model defined in this draft is manipulated vis the I2RS interface. The security concerns mentioned in [draft-ietf-i2rs-architecture] also applies to this draft.

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or ulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

[Editor’s note: to List specific subtrees and data nodes and their sensitivity/vulnerability.]

6. Manageability Considerations

TBD.

7. IANA Considerations

TBD.

8. Acknowledgements

The initial YANG model specified in this draft is based on draft-clemm-i2rs-yang-network-topo but it is modified according to the features of the layer one networks.

We would like to thank the authors of the above mentioned draft for their helpful discussion during the creation of this draft.

9. References

9.1. Normative References

[RFC2119] S. Bradner, "Key words for use in RFCs to indicate requirements levels", RFC 2119, March 1997.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

9.2. Informative References

[draft-ietf-i2rs-architecture] Atlas, A., Halpern, J., Hares, S., Ward, D., Nadeau T., "An Architecture for the Interface to the Routing System", draft-ietf-i2rs-architecture-08, work in progress, January 2015;

[draft-clemm-i2rs-yang-network-topo] Clemm A., Medved J., Tkacik T., Varga R., et al, "A YANG Data Model for Network Topologies", draft-clemm-i2rs-yang-network-topo-01, work in progress, October 2014;

[ietf-netmod-rfc6087bis] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-01, work in progress, October 2014.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC6241, June 2011.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, March 2012.

10. Contributors' Addresses

TBD.

11. Authors' Addresses

Xian Zhang
Huawei Technologies
Email: zhang.xian@huawei.com

Baoquan Rao
Huawei Technologies
raobaoquan@huawei.com

Xufeng Liu
Ericsson
xufeng.liu@ericsson.com