l2tpext Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2016

Q. Sun
I. Farrer
Deutsche Telekom AG
B. Liu
Huawei Technologies
G. Heron
Cisco Systems

July 6, 2015

A YANG Data Model for Keyed IPv6 Tunnels
draft-sun-l2tpext-keyed-v6-tunnel-yang-01

Abstract

This document defines a YANG data model for the configuration and management of Keyed IPv6 tunnels. The data model includes configuration data and state data. Due to the stateless nature of keyed IPv6 tunnels, a model for NETCONF notifications is not necessary.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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.

Table of Contents

1. Introduction

Keyed IPv6 Tunnels [I-D.ietf-l2tpext-keyed-ipv6-tunnel] defines a mechanism for transporting L2 Ethernet frames over IPv6 using L2TPv3 tunnel encapsulation with a mandatory 64-bit cookie. This is used for connecting L2 Ethernet circuits identified by their IPv6 addresses. It is a static L2 tunnelling mechanism that leverages the key property that IPv6 offers: a vast number of unique IP addresses, to identify a tunnel. This differs from differentiating tunnels based on Session ID as defined in [RFC3931]. The layer 2 circuit is mapped to an IPv6 address on the network side so typically, there is a "one session per tunnel" pattern.

Since the control plane of L2TPv3 is not used by Keyed IPv6 tunnels,

the parameters for running Keyed IPv6 tunnel need to be pre-configured. NETCONF [RFC6241]/YANG [RFC6020] provide mechanisms for such configuration. This document defines a YANG data model for the configuration and management of Keyed IPv6 Tunnels.

1.1. Terminology

1.1.1. Requirements Notations

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].

1.1.2. NETCONF Terms

The following terms are defined in [RFC6241] and are not redefined here:

- Client
- Server
- Operation

1.1.3. YANG Terms

The following terms are defined in [RFC6020] and are not redefined here:

- configuration data
- data node
- data tree
- module
- namespace
- YANG

1.1.4. Tree Diagrams

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

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

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

2. YANG Model Overview

The YANG model includes two modules, one for configuration and one for state. To correctly identify a tunnel and create the mapping between the L2 circuit and the IPv6 address, the tuple of source interface, local IPv6 address and remote IPv6 address MUST be unique. Because Session ID is not mandatory for a Keyed IPv6 tunnel to function, Session ID related parameters are optional in the model. Cookies MUST be 64-bit long according to the Section 3 of [I-D.ietf-l2tpext-keyed-ipv6-tunnel]. The requirement for 64-bit cookie constrains interoperability with existing RFC3931 implementations to those configured with a 64-bit cookie.

The data model also includes read-only counters so that statistics for sent and received octets and packets, received packets with errors, and packets that could not be sent due to errors can be read.

This model defines three features for OAM parameters. Those features enable devices to perform related OAM operations on the tunnel if related functions are supported.

3. Keyed IPv6 Tunnel YANG Tree Diagrams

This section describes the tree diagram for the Keyed IPv6 Tunnel YANG model:

```
module: ietf-keyed-v6-tunnel
   +--rw tunnelConfigurations
   |  +--rw tunnelConfiguration* [tunnelName]
   |  |  +--rw tunnelName           string
   |  |  +--rw srcInterface         if:interface-ref
   |  |  +--rw localIPv6            inet:ipv6-address
   |  |  +--rw remoteIPv6           inet:ipv6-address
   |  |  +--rw localSessionId?      uint32
   |  |  +--rw remoteSessionId?     uint32
   |  |  +--rw localCookies
   |  |  |  +--rw localCookie* [cookieName]
   |  |  |     +--rw cookieName     string
   |  |  |     +--rw cookieValue    uint64
   |  |  +--rw remoteCookie         uint64
   |  |  +--rw retainFCS?           empty
   |  |  +--rw enable-vccv!
   |  |  |  +--rw enable-bfd?       empty
   |  |  +--rw enable-bfd?          empty
   |  +--rw disable-pmtu?           empty
   |  +--rw enable-fragmentation!
   |     +--rw fragment-mru?        uint16
   +--ro tunnelStates
      +--ro tunnelState* [tunnelName]
         +--ro tunnelName           string
         +--ro sentPacket?          yang:zero-based-counter64
         +--ro sentByte?            yang:zero-based-counter64
         +--ro rcvdPacket?          yang:zero-based-counter64
         +--ro rcvdByte?            yang:zero-based-counter64
         +--ro droppedPacket?       yang:zero-based-counter64
         +--ro droppedByte?         yang:zero-based-counter64
         +--ro fragmentCounter?     yang:zero-based-counter64
```

Figure 1: Keyed IPv6 Tunnel Tree

The data model defines a configuration container and a state container.

In the configuration container, "srcInterface" is used to identify a L2 circuit endpoint. "localIPv6" and "remoteIPv6" respectively represent the local (source) and remote (destination) IPv6 addresses for the tunnel. The srcInterface and localIPv6 both uniquely identify a tunnel endpoint. If a virtual interface is used, the localIPv6 and remoteIPv6 as a pair MUST also be unique. "localCookie" is a list and has two cookies: one is the newly configured cookie, and the other is previously configured. This is used for the purpose of correctly receiving packets while changing cookies.

Features are defined for FCS-Retention, VCCV, BFD, VCCV-BFD and fragmentation, so that devices supporting those features (or some of which) can enable related functions.

4. Keyed IPv6 Tunnel YANG Model

This module imports typedefs from [RFC6991] and [RFC7223].

```
<CODE BEGINS> file "ietf-keyed-v6-tunnel@2015-07-06.yang"
module ietf-keyed-v6-tunnel {
  namespace "urn:ietf:params:xml:ns:yang:ietf-keyed-v6-tunnel";
  prefix k6tun;

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

  organization "IETF l2tpext Working Group";

  contact
    "qui.sun@external.telekom.de
     ian.farrer@telekom.de
     leo.liubing@huawei.com
     giheron@cisco.com
    ";

  description
    "Keyed IPv6 L2TPv3 Tunnel";

  revision 2015-07-06 {
    description
      "General clean-up"
      ;
    reference
      "draft-sun-l2tpext-keyed-v6-tunnel-yang-01";
  }

  revision 2015-03-09 {
    description
      "Initial version.
      ";
    reference
```

```
        "draft-sun-l2tpext-keyed-v6-tunnel-yang-00";
    }

    /*
     * features
     */
    feature FCS-Retention {
      description
        "Device supports the retention of frame check sequence (FCS)
        as per Section 4.7 of RFC4720";
    }
    feature VCCV {
      description
        "Device supports the Pseudowire Virtual Circuit Connectivity
        Verification (VCCV) as per RFC5085";
    }
    feature BFD {
      description
        "Device supports BFD over the tunnel as per RFC5883";
    }
    feature VCCV-BFD {
      description
        "Device supports BFD over VCCV as per RFC5885";
    }
    feature l2tpv3-fragmentation {
      description
        "Device supports L2TPv3 fragmentation as per RFC4623";
    }

    /*
     * typedefs
     */

    /*
     * groupings
     */

    /*
     * config parameters
     */
    container tunnelConfigurations {
      list tunnelConfiguration {
        key "tunnelName";
        unique "srcInterface remoteIPv6";
        unique "localIPv6 remoteIPv6";
        leaf tunnelName {
          type string;
          description "name of this keyed tunnel";
```

```
        }
        leaf srcInterface {
          type if:interface-ref;
          mandatory true;
          description
            "Source interface that identifies the L2 circuit
            endpoint.";
        }
        leaf localIPv6 {
          type inet:ipv6-address;
          mandatory true;
          description "IPv6 address for local end of keyed tunnel";
        }
        leaf remoteIPv6 {
          type inet:ipv6-address;
          mandatory true;
          description "IPv6 address for remote end of keyed tunnel";
        }
        leaf localSessionId {
          type uint32;
          default 0xFFFFFFFF;
          description
            "Since IPv6 address is used to determine the tunnel
            and there is one session per tunnel, the Session ID
            can be ignored upon receipt. For compatibility with
            other tunnel termination platforms supporting two-stage
            resolution (IPv6 address + Session ID) the Session ID
            is configured with a random value other than all zeros.
            If both ends support one-stage (IPv6 address), then
            the Session ID is recommended to be set to all ones.";
        }
        leaf remoteSessionId {
          type uint32;
          default 0xFFFFFFFF;
          description
            "Since IPv6 address is used to determine the tunnel
            and there is one session per tunnel, the Session ID
            can be ignored upon receipt. For compatibility with
            other tunnel termination platforms supporting two-stage
            resolution (IPv6 address + Session ID) the Session ID
            is configured with a random value other than all zeros.
            If both ends support one-stage (IPv6 address), then
            the Session ID is recommended to be set to all ones.";
        }
        container localCookies {
          list localCookie {
            key "cookieName";
            leaf cookieName {
```

```
            type string;
            description "name identifying this cookie";
          }
          min-elements 2;
          max-elements 2;
          leaf cookieValue {
            type uint64;
            mandatory true;
            description "value of this cookie";
          }
          description
            "List of local cookies - must have two entries at
            all times to enable lossless cookie rollover";
        }
        description
          "The length of cookie MUST be 64-bit. It MUST be
          possible to change the cookie value at any time
          in a manner that does not drop any legitimate
          tunneled packets - i.e. the receiver
          must be willing to accept both 'old' and 'new'
          cookie values during a change of cookie value.";
      }
      leaf remoteCookie {
        type uint64;
        mandatory true;
        description
          "The length of cookie MUST be 64-bit. Only be one
          remote cookie is used for sending packets.";
      }
      leaf retainFCS {
        if-feature FCS-Retention;
        type empty;
        description
          "If this parameter presents, the router is configued
           to retain FCS. Any such router for a tunnel MUST
           retain the FCS for all frames sent over that tunnel.
          ";
      }
      container enable-vccv {
        if-feature VCCV;
        presence "Enable VCCV [RFC5085]";
        leaf enable-bfd {
          if-feature VCCV-BFD;
          type empty;
            description "Enable VCCV-BFD [RFC5885].";
        }
        description "Enable VCCV [RFC5085]";
      }
```

```
        leaf enable-bfd {
          if-feature BFD;
          type empty;
          description
            "Enable BFD over the tunnel [RFC5883].";
        }
        leaf disable-pmtu {
          type empty;
            description "Disable IPv6 PMTU discovery [RFC1981]";
        }
        container enable-fragmentation {
          if-feature l2tpv3-fragmentation;
          presence "Enable L2TPv3 fragmentation [RFC4623]";
          leaf fragment-mru {
            type uint16;
            description "Explicit override for fragmentation MRU";
          }
          description
            "Default is to fragment to PMTU (or 1500 if PMTU disabled)
            minus 52 octet encaps overhead";
        }
        description
          "keyed-v6-tunnel typically supports one l2tpv3 session
          per tunnel. The srcInterface and localIPv6 both uniquely
          identify a tunnel endpoint. If a virtual interface
          is used, the localIPv6 and remoteIPv6 as a pair MUST
          also be unique.
          ";
      }
     description
       "container for list of keyed-v6-tunnel entries";
    }
    container tunnelStates {
      config false;
      list tunnelState {
        key "tunnelName";
        leaf tunnelName {
          type string;
          description "name of this keyed tunnel";
        }
        leaf sentPacket {
          type yang:zero-based-counter64;
          description
            "number of packets sent over tunnel";
        }
        leaf sentByte {
          type yang:zero-based-counter64;
          description
```

```
            "total sent bytes (of inner packets)";
        }
        leaf rcvdPacket {
          type yang:zero-based-counter64;
          description
            "number of packets received from tunnel";
        }
        leaf rcvdByte {
          type yang:zero-based-counter64;
          description
            "total received bytes (of inner packets)";
        }
        leaf droppedPacket {
          type yang:zero-based-counter64;
          description
            "Number of dropped packets";
        }
        leaf droppedByte {
          type yang:zero-based-counter64;
          description
            "Total dropped bytes (of inner packets)";
        }
        leaf fragmentCounter {
          type yang:zero-based-counter64;
          description
            "This is used for counting the fragments of inner
            packets.";
        }
        description "per-tunnel statistics";
      }
      description "container for list of tunnel statistics";
    }
  }
  <CODE ENDS>
```

5. Security Considerations

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 this YANG module which are writable/creatable/deletable (i.e. config true, which is the default). These data nodes may be considered sensitive or vulnerable

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. These are the subtrees and data nodes and their sensitivity/vulnerability:

subtrees and data nodes and state why they are sensitive

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g. via get, get-config or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

subtrees and data nodes and state why they are sensitive

6. IANA Considerations

TBD

7. Acknowledgements

The authors would like to thank Haoxing Shen for his valuable comments.

8. References

8.1. Normative References

[I-D.ietf-l2tpext-keyed-ipv6-tunnel] Konstantynowicz, M., Heron, G., Schatzmayr, R., and W. Henderickx, "Keyed IPv6 Tunnel", draft-ietf-l2tpext-keyed-ipv6-tunnel-04 (work in progress), March 2015.

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

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

[RFC6021] Schoenwaelder, J., "Common YANG Data Types", RFC 6021, October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, 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.

[RFC6991] Schoenwaelder, J., "Common YANG Data Types", RFC 6991, July 2013.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, May 2014.

8.2. Informative References

[RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

Authors' Addresses

Qi Sun
Deutsche Telekom AG
CTO-ATI,Landgrabenweg 151
Bonn, NRW 53227
Germany

Email: qui.sun@external.telekom.de

Ian Farrer
Deutsche Telekom AG
CTO-ATI,Landgrabenweg 151
Bonn, NRW 53227
Germany

Email: ian.farrer@telekom.de

Bing Liu
Huawei Technologies
Q14, Huawei Campus, No.156 Beiqing Road
Beijing, Hai-Dian District 100095
P.R. China

Email: leo.liubing@huawei.com

Giles Heron
Cisco Systems
9-11 New Square, Bedfont Lakes
Feltham, Middlesex TW14 8HA
United Kingdom

Email: giheron@cisco.com