l2tpext Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

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

October 30, 2017

A YANG Data Model for Keyed IPv6 Tunnels
draft-ietf-l2tpext-keyed-v6-tunnel-yang-03

Abstract

This document defines a YANG data model for the configuration and management of Keyed IPv6 tunnels. The data model includes both configuration 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 https://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 May 3, 2018.

Copyright Notice

Table of Contents

1. Introduction

Keyed IPv6 Tunnels [RFC8159] defines a mechanism for transporting L2 Ethernet frames over IPv6 using L2TPv3 tunnel encapsulation with a mandatory 64-bit cookie. It is a static layer 2 tunnelling mechanism that leverages IPv6's vast number of IP addresses to uniquely identify each tunnel, instead of using the L2TPv3 Session ID as the differentiator (as defined in [RFC3931]). The layer 2 circuit is mapped to an IPv6 address on the network side so typically, there is one session per-tunnel.

Since the L2TPv3 control plane is not used by Keyed IPv6 tunnels, the parameters for building a Keyed IPv6 tunnel need to be pre-configured on the two tunnel endpoint devices. 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. For a description of the symbols in these diagrams, please refer to [I-D.ietf-netmod-yang-tree-diagrams].

2. YANG Model Overview

The YANG model is comprised of 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 Section 3 of [RFC8159]. 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 tunnel-configurations
    |  +--rw tunnel-configuration* [tunnel-name]
    |     +--rw tunnel-name             string
    |     +--rw src-interface           if:interface-ref
    |     +--rw local-ipv6              inet:ipv6-address
    |     +--rw remote-ipv6             inet:ipv6-address
    |     +--rw local-session-id?       uint32
    |     +--rw remote-session-id?      uint32
    |     +--rw local-cookies
    |     |  +--rw local-cookie* [cookie-name]
    |     |     +--rw cookie-name     string
    |     |     +--rw cookie-value    uint64
    |     +--rw remote-cookie           uint64
    |     +--rw retain-fcs?             empty {fcs-retention}?
    |     +--rw enable-vccv! {vccv}?
    |     |  +--rw enable-bfd?   empty {vccv-bfd}?
    |     +--rw enable-bfd?             empty {bfd}?
    |     +--rw disable-pmtu?           empty
    |     +--rw enable-fragmentation! {l2tpv3-fragmentation}?
    |     |  +--rw fragment-mru?   uint16
    |     +--rw enable-sequencing?      empty {l2tpv3-sequencing}?
    +--ro tunnel-states
       +--ro tunnel-state* [tunnel-name]
          +--ro tunnel-name         string
          +--ro sent-packet?        yang:zero-based-counter64
          +--ro sent-byte?          yang:zero-based-counter64
          +--ro received-packet?    yang:zero-based-counter64
          +--ro received-byte?      yang:zero-based-counter64
          +--ro dropped-packet?     yang:zero-based-counter64
          +--ro dropped-byte?       yang:zero-based-counter64
          +--ro fragment-counter?   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" hold the local (source) and remote (destination) IPv6 addresses for the tunnel. The tuple of srcInterface and localIPv6 uniquely identify a tunnel endpoint.

If a virtual interface is used, the tuple of localIPv6 and remoteIPv6 MUST also be unique. "localCookie" is a list containing 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.

Nodes are defined for FCS-Retention, VCCV, BFD, VCCV-BFD and fragmentation, so that devices supporting all or some of these features can be configured.

4. Keyed IPv6 Tunnel YANG Model

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

```
<CODE BEGINS> file "ietf-keyed-v6-tunnel@2016-03-21.yang"
module ietf-keyed-v6-tunnel {
  yang-version 1.1;
  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
    "sunqi_ietf@163.com
     ian.farrer@telekom.de
     leo.liubing@huawei.com
     giheron@cisco.com
    ";

  description
    "Keyed IPv6 L2TPv3 Tunnel";

  revision 2016-10-24 {
    description
      "Update of model for -03";
    reference
      "draft-ietf-lwtpext-keyed-v6-tunnel-yang-03";
  }

  revision 2016-03-21 {
    description
      "Added sequencing feature";
```

```
    reference
      "draft-ietf-l2tpext-keyed-v6-tunnel-yang-02";
  }

  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 multihop BFD between tunnel endpoints 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";
  }
  feature l2tpv3-sequencing {
    description
      "Device supports frame sequencing as per section 4.6.1 of
```

```
      RFC3931";
  }

  /*
   * typedefs
   */

  /*
   * groupings
   */

  /*
   * config parameters
   */
  container tunnel-configurations {
    list tunnel-configuration {
      key "tunnel-name";
      unique "src-interface remote-ipv6";
      unique "local-ipv6 remote-ipv6";
      leaf tunnel-name {
        type string;
        description "Name for this keyed tunnel.";
      }
      leaf src-interface {
        type if:interface-ref;
        mandatory true;
        description
          "Source interface that identifies the L2 circuit
          endpoint.";
      }
      leaf local-ipv6 {
        type inet:ipv6-address;
        mandatory true;
        description "IPv6 address for local end of keyed tunnel.";
      }
      leaf remote-ipv6 {
        type inet:ipv6-address;
        mandatory true;
        description "IPv6 address for remote end of keyed tunnel.";
      }
      leaf local-session-id {
        type uint32;
        default "4294967295";
        description
          "As the IPv6 address is used to determine the tunnel
          and there is a single 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 remote-session-id {
        type uint32;
        default "4294967295";
        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 local-cookies {
        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 accept a received cookie matching either
          value during a change of cookie value.";
        list local-cookie {
          key "cookie-name";
          leaf cookie-name {
            type string;
            description "Name identifying this cookie.";
          }
          min-elements 2;
          max-elements 2;
          leaf cookie-value {
            type uint64;
            mandatory true;
            description "value of this cookie";
          }
          description
            "List of local cookies - to allow for cookie rollover
            without traffic loss, must be configured with two
            entries.";
        }
      }
      leaf remote-cookie {
        type uint64;
        mandatory true;
```

```
        description
          "The length of cookie MUST be 64-bit. A single
          remote cookie is used for sending packets.";
      }
      leaf retain-fcs {
        if-feature fcs-retention;
        type empty;
        description
          "If this parameter is present, the router is configued
           to retain the Frame Check Sequence (FCS). Any such router 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 between the tunnel endpoints[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 is
          disabled) minus 52 octets for the encapsulation
          overhead.";
      }
      leaf enable-sequencing {
        if-feature l2tpv3-sequencing;
        type empty;
        description
```

```
          "Enable L2TPv3 sequencing [RFC3931 section 4.6.1].";
      }
      description
        "A keyed-v6-tunnel typically supports one l2tpv3 session
        per tunnel. The src-interface and local-ipv6 both uniquely
        identify a tunnel endpoint. If a virtual interface
        is used, the local-ipv6 and remote-ipv6 as a pair MUST
        also be unique.";
    }
   description
     "Container for list of keyed-v6-tunnel entries";
  }
  container tunnel-states {
    config false;
    list tunnel-state {
      key "tunnel-name";
      leaf tunnel-name {
        type string;
        description "Name of this keyed tunnel.";
      }
      leaf sent-packet {
        type yang:zero-based-counter64;
        description
          "Counter for the number of packets sent over tunnel.";
      }
      leaf sent-byte {
        type yang:zero-based-counter64;
        description
          "Counter for total sent bytes (of inner packets).";
      }
      leaf received-packet {
        type yang:zero-based-counter64;
        description
          "Counter for the number of valid packets received from the
          tunnel.";
      }
      leaf received-byte {
        type yang:zero-based-counter64;
        description
          "Counter for total received bytes (of inner packets).";
      }
      leaf dropped-packet {
        type yang:zero-based-counter64;
        description
          "Counter for number of dropped packets matching this
          tunnel (e.g. due to invalid received cookie,
          insufficient resources to process).";
      }
```

```
      leaf dropped-byte {
        type yang:zero-based-counter64;
        description
          "Counter for total dropped bytes (of inner packets).";
      }
      leaf fragment-counter {
        type yang:zero-based-counter64;
        description
          "Counter for number of received fragments.";
      }
      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:

/tunnelConfigurations/tunnelConfiguration: Could allow traffic to be redirected, (man-in-the-middle attack) or mis-configured (denial-of-service attack).

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:

/tunnel-configurations/tunnel-configuration: Could allow an attacker to inject spoofed traffic into the network.

/tunnel-states/tunnel-state: Could allow an attacker to get unauthorized access to tunnel usage information.

6. IANA Considerations

This document registers the following YANG modules in the "YANG Module Names" registry [RFC6020].

```
name:           ietf-keyed-v6-tunnel
namespace:      urn:ietf:params:xml:ns:yang:ietf-keyed-v6-tunnel
prefix:         k6tun
reference:      TBD
```

7. Acknowledgements

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

8. References

8.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <https://www.rfc-editor.org/info/rfc6536>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <https://www.rfc-editor.org/info/rfc7223>.

[RFC8159] Konstantynowicz, M., Ed., Heron, G., Ed., Schatzmayr, R., and W. Henderickx, "Keyed IPv6 Tunnel", RFC 8159, DOI 10.17487/RFC8159, May 2017, <https://www.rfc-editor.org/info/rfc8159>.

8.2. Informative References

[I-D.ietf-netmod-yang-tree-diagrams] Bjorklund, M. and L. Berger, "YANG Tree Diagrams", draft-ietf-netmod-yang-tree-diagrams-02 (work in progress), October 2017.

[RFC3931] Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, DOI 10.17487/RFC3931, March 2005, <https://www.rfc-editor.org/info/rfc3931>.

Authors' Addresses

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

Email: sunqi_ietf@163.com

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