NETCONF Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 6, 2018

K. Watsen
Juniper Networks

June 4, 2018

YANG Data Model for Global Trust Anchors
draft-ietf-netconf-trust-anchors-00

Abstract

This document defines a YANG 1.1 data model for configuring global sets of X.509 certificates and SSH host-keys that can be referenced by other data models for trust. While the SSH host-keys are uniquely for the SSH protocol, the X.509 certificates may have multiple uses, including authenticating protocol peers and verifying signatures.

Editorial Note (To be removed by RFC Editor)

This draft contains many placeholder values that need to be replaced with finalized values at the time of publication. This note summarizes all of the substitutions that are needed. No other RFC Editor instructions are specified elsewhere in this document.

Artwork in this document contains shorthand references to drafts in progress. Please apply the following replacements:

- "XXXX" --> the assigned RFC value for this draft
- "YYYY" --> the assigned RFC value for draft-ietf-netconf-crypto-types

Artwork in this document contains placeholder values for the date of publication of this draft. Please apply the following replacement:

- "2018-06-04" --> the publication date of this draft

The following Appendix section is to be removed prior to publication:

- Appendix A. Change Log

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 December 6, 2018.

Table of Contents

1. Introduction

This document defines a YANG 1.1 [RFC7950] data model for configuring global sets of X.509 certificates and SSH host-keys that can be referenced by other data models for trust. While the SSH host-keys are uniquely for the SSH protocol, the X.509 certificates may be used for multiple uses, including authenticating protocol peers and verifying signatures.

This document in compliant with Network Management Datastore Architecture (NMDA) [RFC8342]. For instance, to support trust anchors installed during manufacturing, it is expected that such data may appear only in <operational>.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Tree Diagram Notation

Tree diagrams used in this document follow the notation defined in [RFC8340].

2. The Trust Anchors Model

2.1. Tree Diagram

The following tree diagram provides an overview of the "ietf-trust-anchors" module.

```
module: ietf-trust-anchors
  +--rw trust-anchors
     +--rw pinned-certificates* [name]
     |  +--rw name                  string
     |  +--rw description?          string
     |  +--rw pinned-certificate* [name]
     |     +--rw name                      string
     |     +--rw cert                      ct:trust-anchor-cert-cms
     |     +---n certificate-expiration
     |        +-- expiration-date?   yang:date-and-time
     +--rw pinned-host-keys* [name]
        +--rw name               string
        +--rw description?       string
        +--rw pinned-host-key* [name]
           +--rw name        string
           +--rw host-key    ct:ssh-host-key
```

2.2. Example Usage

The following example illustrates trust anchors in <operational> as described by Section 5.3 in [RFC8342]. This datastore view illustrates data set by the manufacturing process alongside conventional configuration. This trust anchors instance has five sets of pinned certificates and one set of pinned host keys.

```
<trust-anchors
   xmlns="urn:ietf:params:xml:ns:yang:ietf-trust-anchors"
   xmlns:or="urn:ietf:params:xml:ns:yang:ietf-origin">

  <!-- Manufacturer's trusted root CA certs -->
  <pinned-certificates or:origin="or:system">
    <name>manufacturers-root-ca-certs</name>
    <description>
      Certificates built into the device for authenticating
      manufacturer-signed objects, such as TLS server certificates,
      vouchers, etc.  Note, though listed here, these are not
      configurable; any attempt to do so will be denied.
    </description>
    <pinned-certificate>
      <name>Manufacturer Root CA cert 1</name>
      <cert>base64encodedvalue==</cert>
    </pinned-certificate>
    <pinned-certificate>
      <name>Manufacturer Root CA cert 2</name>
      <cert>base64encodedvalue==</cert>
    </pinned-certificate>
  </pinned-certificates>
```

```
<!-- specific end-entity certs for authenticating servers -->
<pinned-certificates or:origin="or:intended">
  <name>explicitly-trusted-server-certs</name>
  <description>
    Specific server authentication certificates for explicitly
    trusted servers.  These are needed for server certificates
    that are not signed by a pinned CA.
  </description>
  <pinned-certificate>
    <name>Fred Flintstone</name>
    <cert>base64encodedvalue==</cert>
  </pinned-certificate>
</pinned-certificates>

<!-- trusted CA certs for authenticating servers -->
<pinned-certificates or:origin="or:intended">
  <name>explicitly-trusted-server-ca-certs</name>
  <description>
    Trust anchors (i.e. CA certs) that are used to authenticate
    server connections.  Servers are authenticated if their
    certificate has a chain of trust to one of these CA
    certificates.
  </description>
  <pinned-certificate>
    <name>ca.example.com</name>
    <cert>base64encodedvalue==</cert>
  </pinned-certificate>
</pinned-certificates>

<!-- specific end-entity certs for authenticating clients -->
<pinned-certificates or:origin="or:intended">
  <name>explicitly-trusted-client-certs</name>
  <description>
    Specific client authentication certificates for explicitly
    trusted clients.  These are needed for client certificates
    that are not signed by a pinned CA.
  </description>
  <pinned-certificate>
    <name>George Jetson</name>
    <cert>base64encodedvalue==</cert>
  </pinned-certificate>
</pinned-certificates>

<!-- trusted CA certs for authenticating clients -->
<pinned-certificates or:origin="or:intended">
  <name>explicitly-trusted-client-ca-certs</name>
  <description>
    Trust anchors (i.e. CA certs) that are used to authenticate
```

```
        client connections.  Clients are authenticated if their
        certificate has a chain of trust to one of these CA
        certificates.
      </description>
      <pinned-certificate>
        <name>ca.example.com</name>
        <cert>base64encodedvalue==</cert>
      </pinned-certificate>
    </pinned-certificates>

    <!-- trusted CA certs for random HTTPS servers on Internet -->
    <pinned-certificates or:origin="or:system">
      <name>common-ca-certs</name>
      <description>
        Trusted certificates to authenticate common HTTPS servers.
        These certificates are similar to those that might be
        shipped with a web browser.
      </description>
      <pinned-certificate>
        <name>ex-certificate-authority</name>
        <cert>base64encodedvalue==</cert>
      </pinned-certificate>
    </pinned-certificates>

    <!-- specific SSH host keys for authenticating clients -->
    <pinned-host-keys or:origin="or:intended">
      <name>explicitly-trusted-ssh-host-keys</name>
      <description>
        Trusted SSH host keys used to authenticate SSH servers.
        These host keys would be analogous to those stored in
        a known_hosts file in OpenSSH.
      </description>
      <pinned-host-key>
        <name>corp-fw1</name>
        <host-key>base64encodedvalue==</host-key>
      </pinned-host-key>
    </pinned-host-keys>

  </trust-anchors>
```

The following example illustrates the "certificate-expiration" notification in use with the NETCONF protocol.

```
   [Note: '\' line wrapping for formatting only]

   <notification
     xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
     <eventTime>2018-05-25T00:01:00Z</eventTime>
     <trust-anchors
       xmlns="urn:ietf:params:xml:ns:yang:ietf-trust-anchors">
       <pinned-certificates>
         <name>explicitly-trusted-client-certs</name>
         <pinned-certificate>
           <name>George Jetson</name>
           <certificate-expiration>
             <expiration-date>2018-08-05T14:18:53-05:00</expiration-dat\
   e>
           </certificate-expiration>
         </pinned-certificate>
       </pinned-certificates>
     </trust-anchors>
   </notification>
```

2.3. YANG Module

This YANG module imports modules from [RFC6536], [RFC6991] and [I-D.ietf-netconf-crypto-types].

```
     <CODE BEGINS> file "ietf-trust-anchors@2018-06-04.yang"
   module ietf-trust-anchors {
     yang-version 1.1;

     namespace "urn:ietf:params:xml:ns:yang:ietf-trust-anchors";
     prefix "ta";

     import ietf-yang-types {
       prefix yang;
       reference
         "RFC 6991: Common YANG Data Types";
     }

     import ietf-crypto-types {
       prefix ct;
       reference
         "RFC YYYY: Common YANG Data Types for Cryptography";
     }

     organization
      "IETF NETCONF (Network Configuration) Working Group";
```

```
     contact
      "WG Web:   <http://datatracker.ietf.org/wg/netconf/>
       WG List:  <mailto:netconf@ietf.org>

       Author:   Kent Watsen
                 <mailto:kwatsen@juniper.net>";


     description
      "This module defines a data model for configuring global
       trust anchors used by other data models.  The data model
       enables the configuration of sets of trust anchors.
       This data model supports configuring trust anchors for
       both X.509 certificates and SSH host keys.

       Copyright (c) 2018 IETF Trust and the persons identified
       as authors of the code. All rights reserved.

       Redistribution and use in source and binary forms, with
       or without modification, is permitted pursuant to, and
       subject to the license terms contained in, the Simplified
       BSD License set forth in Section 4.c of the IETF Trust's
       Legal Provisions Relating to IETF Documents
       (http://trustee.ietf.org/license-info).

       This version of this YANG module is part of RFC XXXX; see
       the RFC itself for full legal notices.";

     revision "2018-06-04" {
       description
        "Initial version";
       reference
        "RFC XXXX: YANG Data Model for Global Trust Anchors";
     }


     /*****************************************************************/
     /*   Typedefs for leafrefs to commonly referenced objects   */
     /*****************************************************************/

     typedef pinned-certificates-ref {
       type leafref {
         path "/ta:trust-anchors/ta:pinned-certificates/ta:name";
         require-instance false;
       }
       description
         "This typedef enables importing modules to easily define a
          leafref to a 'pinned-certificates' object.  The require
```

```
          instance attribute is false to enable the referencing of
          pinned certificates that exist only in <operational>.";
      reference
        "RFC 8342: Network Management Datastore Architecture (NMDA)";
    }

    typedef pinned-host-keys-ref {
      type leafref {
        path "/ta:trust-anchors/ta:pinned-host-keys/ta:name";
        require-instance false;
      }
      description
        "This typedef enables importing modules to easily define a
         leafref to a 'pinned-host-keys' object.  The require
         instance attribute is false to enable the referencing of
         pinned host keys that exist only in <operational>.";
      reference
        "RFC 8342: Network Management Datastore Architecture (NMDA)";
    }


    /*********************************/
    /*   Protocol accessible nodes   */
    /*********************************/

    container trust-anchors {
      description
        "Contains sets of X.509 certificates and SSH host keys.";

      list pinned-certificates {
        key name;
        description
          "A list of pinned certificates.  These certificates can be
           used by a server to authenticate clients, or by a client
           to authenticate servers.  Each list of pinned certificates
           SHOULD be specific to a purpose, as the list as a whole
           may be referenced by other modules.  For instance, a
           NETCONF server's configuration might use a specific list
           of pinned certificates for when authenticating NETCONF
           client connections.";
        leaf name {
          type string;
          description
            "An arbitrary name for this list of pinned
             certificates.";
        }
        leaf description {
          type string;
```

```
          description
            "An arbitrary description for this list of pinned
             certificates.";
        }
        list pinned-certificate {
          key name;
          description
            "A pinned certificate.";
          leaf name {
            type string;
            description
              "An arbitrary name for this pinned certificate. The
               name must be unique across all lists of pinned
               certificates (not just this list) so that leafrefs
               from another module can resolve to unique values.";
          }
          leaf cert {
            type ct:trust-anchor-cert-cms;
            mandatory true;
            description
              "The binary certificate data for this pinned
               certificate.";
            reference
              "RFC YYYY: Common YANG Data Types for Cryptography";
          }
          notification certificate-expiration {
            description
             "A notification indicating that the configured trust
              anchor is either about to expire or has already expired.
              When to send notifications is an implementation specific
              decision, but it is RECOMMENDED that a notification be
              sent once a month for 3 months, then once a week for
              four weeks, and then once a day thereafter until the
              issue is resolved.";
            leaf expiration-date {
              type yang:date-and-time;
              //mandatory true;
              description
                "Identifies the expiration date on the certificate.";
            }
          }

        }
      }
      list pinned-host-keys {
        key name;
        description
          "A list of pinned host keys.  These pinned host-keys can
```

```
           be used by clients to authenticate SSH servers.  Each
           list of pinned host keys SHOULD be specific to a purpose,
           so the list as a whole may be referenced by other modules.
           For instance, a NETCONF client's configuration might
           point to a specific list of pinned host keys for when
           authenticating specific SSH servers.";
         leaf name {
           type string;
           description
             "An arbitrary name for this list of pinned SSH
              host keys.";
         }
         leaf description {
           type string;
           description
             "An arbitrary description for this list of pinned SSH
              host keys.";
         }
         list pinned-host-key {
           key name;
           description
             "A pinned host key.";
           leaf name {
             type string;
             description
               "An arbitrary name for this pinned host-key. Must be
                unique across all lists of pinned host-keys (not just
                this list) so that a leafref to it from another module
                can resolve to unique values.";
           }
           leaf host-key {
             type ct:ssh-host-key;
             mandatory true;
             description
               "The binary public key data for this pinned host key.";
             reference
               "RFC YYYY: Common YANG Data Types for Cryptography";
           }
         }
       }
     }

   }
     <CODE ENDS>
```

3. Security Considerations

The YANG module defined in this document is designed to be accessed via YANG based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. Both of these protocols have mandatory-to-implement secure transport layers (e.g., SSH, TLS) with mutual authentication.

The NETCONF access control model (NACM) [RFC6536] provides the means to restrict access for particular users to a pre-configured subset of all available protocol operations and content.

There are a number of data nodes defined in this YANG module that 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:

/: The entire data tree defined by this module is sensitive to write operations. For instance, the addition or removal of any trust anchor may dramatically alter the implemented security policy. However, no NACM annotations are applied as the data SHOULD be editable by users other than a designated 'recovery session'.

None of the readable data nodes in this YANG module are considered sensitive or vulnerable in network environments.

This module does not define any RPCs, actions, or notifications, and thus the security consideration for such is not provided here.

4. IANA Considerations

4.1. The IETF XML Registry

This document registers one URI in the "ns" subregistry of the IETF XML Registry [RFC3688]. Following the format in [RFC3688], the following registration is requested:

URI: urn:ietf:params:xml:ns:yang:ietf-trust-anchors
Registrant Contact: The NETCONF WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

4.2. The YANG Module Names Registry

This document registers one YANG module in the YANG Module Names registry [RFC6020]. Following the format in [RFC6020], the the following registration is requested:

```
name:         ietf-trust-anchors
namespace:    urn:ietf:params:xml:ns:yang:ietf-trust-anchors
prefix:       ta
reference:    RFC XXXX
```

5. References

5.1. Normative References

[I-D.ietf-netconf-crypto-types]
Watsen, K., "Common YANG Data Types for Cryptography", draft-ietf-netconf-crypto-types-00 (work in progress), June 2018.

[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>.

[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>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

5.2. Informative References

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[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>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

Appendix A. Change Log

A.1. I-D to 00

- Now imports and uses the crypto-types module.
- FIXME
- FIXME
- FIXME: added notification example...

Acknowledgements

The authors would like to thank for following for lively discussions on list and in the halls (ordered by last name): Martin Bjorklund, Balazs Kovacs, Eric Voit, and Liang Xia.

Author’s Address

Kent Watsen
Juniper Networks

EMail: kwatsen@juniper.net