NETCONF Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2020

K. Watsen
Watsen Networks
H. Birkholz
Fraunhofer SIT

November 1, 2019

A YANG Data Model for a Truststore
draft-ietf-netconf-trust-anchors-07

Abstract

This document defines a YANG 1.1 data model for configuring global sets of X.509 certificates, SSH host-keys, raw public keys, and PSKs (pairwise-symmetric or pre-shared keys) that can be referenced by other data models for trust. While the SSH host-keys are uniquely for the SSH protocol, certificates, raw public keys, and PSKs 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:

- "2019-11-02" --> 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

Copyright Notice

Table of Contents

1. Introduction

This document defines a YANG 1.1 [RFC7950] data model for configuring global sets of X.509 certificates, SSH host-keys, raw public keys, and PSKs (pairwise-symmetric or pre-shared keys) that can be referenced by other data models for trust. While the SSH host-keys are uniquely for the SSH protocol, certificates, raw public keys, and PSKs may have 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 would 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-truststore" module.

```
module: ietf-truststore
  +--rw truststore
```

```
        +--rw certificates* [name] {x509-certificates}?
        |  +--rw name           string
        |  +--rw description?   string
        |  +--rw certificate* [name]
        |     +--rw name                    string
        |     +--rw cert                    trust-anchor-cert-cms
        |     +---n certificate-expiration
        |        +-- expiration-date    yang:date-and-time
        +--rw host-keys* [name] {ssh-host-keys}?
        |  +--rw name           string
        |  +--rw description?   string
        |  +--rw host-key* [name]
        |     +--rw name        string
        |     +--rw host-key    ct:ssh-host-key
        +--rw psks* [name] {psks}?
        |  +--rw name           string
        |  +--rw description?   string
        |  +--rw psk* [name]
        |     +--rw name    string
        |     +--rw psk     ct:psk
        +--rw raw-public-keys* [name] {raw-public-keys}?
           +--rw name              string
           +--rw description?      string
           +--rw raw-public-key* [name]
              +--rw name              string
              +--rw raw-public-key    ct:raw-public-key

     grouping local-or-truststore-certs-grouping
       +-- (local-or-truststore)
          +--:(local) {local-definitions-supported}?
          |  +-- local-definition
          |     +-- cert*                     trust-anchor-cert-cms
          |     +---n certificate-expiration
          |        +-- expiration-date    yang:date-and-time
          +--:(truststore) {truststore-supported,x509-certificates}?
             +-- truststore-reference?   ts:certificates-ref
     grouping local-or-truststore-host-keys-grouping
       +-- (local-or-truststore)
          +--:(local) {local-definitions-supported}?
          |  +-- local-definition
          |     +-- host-key*   ct:ssh-host-key
          +--:(truststore) {truststore-supported,ssh-host-keys}?
             +-- truststore-reference?   ts:host-keys-ref
     grouping local-or-truststore-psks-grouping
       +-- (local-or-truststore)
          +--:(local) {local-definitions-supported}?
          |  +-- local-definition
          |     +-- psk*   ct:psk
```

```
        +--:(truststore) {truststore-supported,psks}?
           +-- truststore-reference?   ts:psks-ref
     grouping local-or-truststore-raw-pub-keys-grouping
       +-- (local-or-truststore)
          +--:(local) {local-definitions-supported}?
          |  +-- local-definition
          |     +-- raw-public-key*   ct:raw-public-key
          +--:(truststore) {truststore-supported,raw-public-keys}?
             +-- truststore-reference?   ts:raw-public-keys-ref
     grouping truststore-grouping
       +-- certificates* [name] {x509-certificates}?
       |  +-- name?          string
       |  +-- description?   string
       |  +-- certificate* [name]
       |     +-- name?                    string
       |     +-- cert                     trust-anchor-cert-cms
       |     +---n certificate-expiration
       |        +-- expiration-date    yang:date-and-time
       +-- host-keys* [name] {ssh-host-keys}?
       |  +-- name?          string
       |  +-- description?   string
       |  +-- host-key* [name]
       |     +-- name?       string
       |     +-- host-key    ct:ssh-host-key
       +-- psks* [name] {psks}?
       |  +-- name?          string
       |  +-- description?   string
       |  +-- psk* [name]
       |     +-- name?   string
       |     +-- psk     ct:psk
       +-- raw-public-keys* [name] {raw-public-keys}?
          +-- name?             string
          +-- description?      string
          +-- raw-public-key* [name]
             +-- name?             string
             +-- raw-public-key    ct:raw-public-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 six sets of pinned certificates and one set of pinned host keys.

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

```
    <!-- Manufacturer's trusted root CA certs -->
    <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>
      <certificate>
        <name>Manufacturer Root CA cert 1</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
      <certificate>
        <name>Manufacturer Root CA cert 2</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
    </certificates>

    <!-- specific end-entity certs for authenticating servers -->
    <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 CA.
      </description>
      <certificate>
        <name>Fred Flintstone</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
    </certificates>

    <!-- trusted CA certs for authenticating servers -->
    <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>
      <certificate>
        <name>ca.example.com</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
    </certificates>
```

```
    <!-- specific end-entity certs for authenticating clients -->
    <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 CA.
      </description>
      <certificate>
        <name>George Jetson</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
    </certificates>

    <!-- trusted CA certs for authenticating clients -->
    <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>
      <certificate>
        <name>ca.example.com</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
    </certificates>

    <!-- trusted CA certs for random HTTPS servers on Internet -->
    <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>
      <certificate>
        <name>ex-certificate-authority</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
    </certificates>

    <!-- specific SSH host keys for authenticating clients -->
    <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>
      <host-key>
        <name>corp-fw1</name>
        <host-key>base64encodedvalue==</host-key>
      </host-key>
    </host-keys>

  </truststore>

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

  ========== NOTE: '\' line wrapping per BCP XXX (RFC XXXX) ===========

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

2.3. YANG Module

This YANG module imports modules from [RFC8341] and [I-D.ietf-netconf-crypto-types].

```
<CODE BEGINS> file "ietf-truststore@2019-11-02.yang"


<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) [RFC8341] 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. For this reason, the NACM extension "default-deny-write" has been set for the entire data tree.

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-truststore
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-truststore
namespace:    urn:ietf:params:xml:ns:yang:ietf-truststore
prefix:       ta
reference:    RFC XXXX
```

5. References

5.1. Normative References

[I-D.ietf-netconf-crypto-types]
Watsen, K. and H. Wang, "Common YANG Data Types for Cryptography", draft-ietf-netconf-crypto-types-11 (work in progress), October 2019.

[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>.

[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>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

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. 00 to 01

- Added features "x509-certificates" and "ssh-host-keys".
- Added nacm:default-deny-write to "trust-anchors" container.

A.2. 01 to 02

- Switched "list pinned-certificate" to use the "trust-anchor-cert-grouping" from crypto-types. Effectively the same definition as before.

A.3. 02 to 03

- Updated copyright date, boilerplate template, affiliation, folding algorithm, and reformatted the YANG module.

A.4. 03 to 04

- Added groupings 'local-or-truststore-certs-grouping' and 'local-or-truststore-host-keys-grouping', matching similar definitions in the keystore draft. Note new (and incomplete) "truststore" usage!
- Related to above, also added features 'truststore-supported' and 'local-trust-anchors-supported'.

A.5. 04 to 05

- Renamed "trust-anchors" to "truststore"
- Removed "pinned." prefix everywhere, to match truststore rename
- Moved everything under a top-level 'grouping' to enable use in other contexts.
- Renamed feature from 'local-trust-anchors-supported' to 'local-definitions-supported' (same name used in keystore)
- Removed the "require-instance false" statement from the "*-ref" typedefs.
- Added missing "ssh-host-keys" and "x509-certificates" if-feature statements

A.6. 05 to 06

- Editorial changes only.

A.7. 06 to 07

- Added Henk Birkholz as a co-author (thanks Henk!)

- Added PSKs and raw public keys to Truststore.

Acknowledgements

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

Authors' Addresses

Kent Watsen
Watsen Networks

EMail: kent+ietf@watsen.net

Henk Birkholz
Fraunhofer SIT

EMail: henk.birkholz@sit.fraunhofer.de