NETCONF Working Group
Internet-Draft
Obsoletes: 5539 (if approved)
Intended status: Standards Track
Expires: November 11, 2013

M. Badra
LIMOS Laboratory
A. Luchuk
SNMP Research, Inc.
J. Schoenwaelder
Jacobs University Bremen
May 10, 2013

Using the NETCONF Protocol over Transport Layer Security (TLS)
draft-ietf-netconf-rfc5539bis-03

Abstract

The Network Configuration Protocol (NETCONF) provides mechanisms to install, manipulate, and delete the configuration of network devices. This document describes how to use the Transport Layer Security (TLS) protocol to secure the exchange of NETCONF messages. This document obsoletes RFC 5539.

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 November 11, 2013.

Copyright Notice

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 NETCONF protocol [RFC6241] defines a mechanism through which a network device can be managed. NETCONF is connection-oriented, requiring a persistent connection between peers. This connection must provide integrity, confidentiality, peer authentication, and reliable, sequenced data delivery.

This document defines "NETCONF over TLS", which includes support for certificate and pre-shared key (PSK)-based authentication and key derivation, utilizing the protected ciphersuite negotiation, mutual authentication, and key management capabilities of the TLS (Transport Layer Security) protocol, described in [RFC5246]. A YANG data model is provided for configuring the policy used to map X.509 certificates into NETCONF usernames and to provision TLS pre-shared keys and to associate these keys with NETCONF usernames.

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].

2. NETCONF over TLS

Since TLS is application-protocol-independent, NETCONF can operate on top of the TLS protocol transparently. This document defines how NETCONF can be used within a TLS session.

2.1. Connection Initiation

In many deployments, the NETCONF client will initiate the connection to a NETCONF server as described in Section 2.1.1. However, in order to use NETCONF in environments where middleboxes prevent the client from establishing the connection, the server may initiate the connection as described in Section 2.1.2 (call home).

2.1.1. Client to Server

The peer acting as the NETCONF client MUST act as the TLS client. The TLS client actively opens the TLS connection and the TLS server passively listens for the incoming TLS connection on the TCP port 6513. The TLS client MUST therefore send the TLS ClientHello message to begin the TLS handshake. Once the TLS handshake has finished, the client and the server MAY begin to exchange NETCONF messages. Client and server identity verification (as described in Section 2.4 and Section 2.5) is done before the <hello> message is sent; for the server, this means the identity verification is completed before the NETCONF session has started.

2.1.2. Server to Client (Call Home)

The peer acting as the NETCONF server MUST act as the TLS client. The TLS client actively opens the TLS connection and the TLS server passively listens for the incoming TLS connection on the TCP port YYYY. The TLS client MUST therefore send the TLS ClientHello message to begin the TLS handshake. Once the TLS handshake has finished, the client and the server MAY begin to exchange NETCONF messages. Client and server identity verification (as described in Section 2.4 and Section 2.5) is done before the <hello> message is sent; for the server, this means the identity verification is completed before the NETCONF session has started.

2.2. Message Framing

All NETCONF messages MUST be sent as TLS "application data". It is possible that multiple NETCONF messages be contained in one TLS record, or that a NETCONF message be transferred in multiple TLS records.

The previous version [RFC5539] of this document used the framing sequence defined in [RFC4742], under the assumption that it could not be found in well-formed XML documents. However, this assumption is not correct [RFC6242]. In order to solve this problem, this document adopts the framing protocol defined in [RFC6242] as follows:

The <hello> message MUST be followed by the character sequence]]>]]>. Upon reception of the <hello> message, the receiving peer's TLS Transport layer conceptually passes the <hello> message to the Messages layer. If the :base:1.1 capability is advertised by both peers, the chunked framing mechanism defined in Section 4.2 of [RFC6242] is used for the remainder of the NETCONF session. Otherwise, the old end-of-message-based mechanism (see Section 4.3 of [RFC6242]) is used.

2.3. Connection Closure

A NETCONF server will process NETCONF messages from the NETCONF client in the order in which they are received. A NETCONF session is closed using the <close-session> operation. When the NETCONF server processes a <close-session> operation, the NETCONF server SHALL respond and close the TLS session as described in [RFC5246] Section 7.2.1. The NETCONF server MUST NOT process any NETCONF messages received after the <close-session> operation.

2.4. X.509-based Authentication, Identification and Authorization

Implementations MAY optionally support TLS certificate-based authentication [RFC5246]. If the implementation supports TLS certificate-based authentication, then the following sections apply.

2.4.1. Server Identity

If the certificate presented by a NETCONF server has passed certification path validation [RFC5280] to a configured trust anchor, the NETCONF client MUST carefully examine the certificate presented by the server to determine if it meets the client's expectations. Particularly, the NETCONF client MUST check its understanding of the NETCONF server hostname against the server's identity as presented in the server Certificate message, in order to prevent man-in-the-middle attacks.

Matching is performed according to the rules and guidelines defined in [RFC6125]. If the match fails, the NETCONF client MUST either ask for explicit user confirmation or terminate the connection and indicate the NETCONF server's identity is suspect.

Additionally, NETCONF clients MUST verify the binding between the identity of the NETCONF servers to which they connect and the public keys presented by those servers. NETCONF clients SHOULD implement the algorithm in Section 6 of [RFC5280] for general certificate validation, but MAY supplement that algorithm with other validation methods that achieve equivalent levels of verification (such as comparing the NETCONF server certificate against a local store of already-verified certificates and identity bindings).

If the NETCONF client has external information as to the expected identity of the NETCONF server, the hostname check MAY be omitted.

2.4.2. Client Identity

The NETCONF server MUST verify the identity of the NETCONF client to ensure that the incoming request to establish a NETCONF session is legitimate before the NETCONF session is started.

The NETCONF protocol [RFC6241] requires that the transport protocol's authentication process MUST result in an authenticated NETCONF client identity whose permissions are known to the server. The authenticated identity of a client is commonly referred to as the NETCONF username.

The username provided by the NETCONF over TLS implementation will be made available to the NETCONF message layer as the NETCONF username

without modification. If the username does not comply to the NETCONF requirements on usernames [RFC6241], i.e., the username is not representable in XML, the TLS session MUST be dropped.

2.4.2.1. Deriving NETCONF Usernames from X.509 Certificates

After completing the TLS handshake, the NETCONF server attempts to derive a NETCONF username from the X.509 certificate presented by the NETCONF client. If the NETCONF server cannot derive a valid NETCONF username from the presented certificate, then the NETCONF server MUST close the TLS connection, and MUST NOT accept NETCONF messages over it. The NETCONF server uses the algorithm defined in [I-D.ietf-netmod-snmp-cfg] to extract a NETCONF username from the X.509 certificate presented by the NETCONF client. The cert-map list in the ietf-netconf-tls YANG submodule specifies how a NETCONF server transforms a certificate into a NETCONF username.

2.5. Pre-Shared-Key-based Authentication, Identification and Authorization

Implementations MAY optionally support TLS Pre-Shared Key (PSK) authentication [RFC4279]. RFC4279 describes pre-shared key ciphersuites for TLS. The description of the psk-maps container in the ietf-netconf-tls YANG submodule, defined in Section 4.3, specifies how a NETCONF server associates a TLS pre-shared key with a NETCONF username.

2.6. Cipher Suites

Implementations of the protocol specified in this document MAY implement any TLS cipher suite that provides mutual authentication [RFC5246]. However, implementations MUST support TLS 1.2 [RFC5246] and are REQUIRED to support the mandatory-to-implement cipher suite, which is TLS_RSA_WITH_AES_128_CBC_SHA. This document is assumed to apply to future versions of TLS; in which case, the mandatory-to-implement cipher suite for the implemented version MUST be supported.

3. Data Model Overview

In order to support future extensibility of the NETCONF configuration data model, the YANG definitions have been organized in a set of YANG submodules, all sharing the same module namespace.

- ietf-netconf-config: The module importing the submodules and defining the module namespace.

- ietf-netconf-common: The submodule providing common definitions shared by all submodules.

- ietf-netconf-tls: The submodule defining configuration objects for the NETCONF over TLS transport.

This organizations allows to add configuration support for additional NETCONF features while keeping the number of namespaces that have to be dealt with down to a minimum. If new definitions need to be added to the NETCONF configuration data model, either an existing YANG submodule can be updated or a new YANG submodule can be written. In both cases, the new document will carry an updated version of the "ietf-netconf-config" module importing the submodules.

The YANG submodule "ietf-netconf-tls" defines (i) how to configure the policy used to map X.509 certificates into NETCONF usernames and (ii) the mechanisms used to provision pre-shared keys and to associate them with NETCONF usernames. The mapping of X.509 certificates to NETCONF usernames imports definitions from [RFC6536] and [I-D.ietf-netmod-snmp-cfg].

```
   +--rw netconf
      +--rw tls
         +--rw enabled?     boolean
         +--rw cert-maps
         |  +--rw cert-to-name* [id]
         |     +--rw id              uint32
         |     +--rw fingerprint     x509c2n:tls-fingerprint
         |     +--rw map-type        identityref
         |     +--rw name            string
         +--rw psk-maps
            +--rw psk-map* [psk-identity]
               +--rw psk-identity         string
               +--rw user-name            nacm:user-name-type
               +--rw not-valid-before?    yang:date-and-time
               +--rw not-valid-after?     yang:date-and-time
               +--rw key                  string
```

The meaning of the symbols in this diagrams is as follows:

- 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 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. Definitions

The YANG modules and submodules import type definitions and groupings from [I-D.ietf-netmod-rfc6021-bis], [RFC6536], and [I-D.ietf-netmod-snmp-cfg].

4.1. Module 'ietf-netconf-config'

```
<CODE BEGINS> file "ietf-netconf-config@2013-05-07.yang"

module ietf-netconf-config {

  namespace "urn:ietf:params:xml:ns:yang:ietf-netconf-config";
  prefix "ncconf";

  include ietf-netconf-common {
    revision-date 2013-05-07;
  }

  include ietf-netconf-tls {
    revision-date 2013-05-07;
  }

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

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

    WG Chair: Mehmet Ersue
              <mailto:mehmet.ersue@nsn.com>

    WG Chair: Bert Wijnen
              <mailto:bertietf@bwijnen.net>

    Editor:   Mohamad Badra
```

```
                <mailto:mbadra@gmail.com>

                Alan Luchuk
                <mailto:luchuk@snmp.com>

                Juergen Schoenwaelder
                <mailto:j.schoenwaelder@jacobs-university.de>";

    description
     "This module contains a collection of YANG definitions for
      configuring NETCONF servers.

      Copyright (c) 2013 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.";
    // RFC Ed.: replace XXXX with actual RFC number and
    // remove this note

    // RFC Ed.: please update the date to the date of publication

    revision "2013-05-07" {
      description
       "Initial version";
      reference
       "RFC XXXX: NETCONF over Transport Layer Security (TLS)";
    }
  }
  <CODE ENDS>
```

4.2. Submodule 'ietf-netconf-common'

```
  <CODE BEGINS> file "ietf-netconf-common@2013-05-07.yang"

  submodule ietf-netconf-common {

    belongs-to ietf-netconf-config {
      prefix ncconf;
```

```
  }

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

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

    WG Chair: Mehmet Ersue
              <mailto:mehmet.ersue@nsn.com>

    WG Chair: Bert Wijnen
              <mailto:bertietf@bwijnen.net>

    Editor:   Mohamad Badra
              <mailto:mbadra@gmail.com>

              Alan Luchuk
              <mailto:luchuk@snmp.com>

              Juergen Schoenwaelder
              <mailto:j.schoenwaelder@jacobs-university.de>";

  description
   "This submodule contains a collection of common YANG definitions
    for configuring NETCONF servers.

    Copyright (c) 2013 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.";
  // RFC Ed.: replace XXXX with actual RFC number and
  // remove this note

  // RFC Ed.: please update the date to the date of publication

  revision "2013-05-07" {
    description
     "Initial version";
```

```
      reference
       "RFC XXXX: NETCONF over Transport Layer Security (TLS)";
    }

    container netconf {
      description
       "Top-level container for NETCONF related configuration
        objects.";
    }
  }
  <CODE ENDS>
```

4.3. Submodule 'ietf-netconf-tls'

```
  <CODE BEGINS> file "ietf-netconf-tls@2013-05-07.yang"

  submodule ietf-netconf-tls {

    belongs-to ietf-netconf-config {
      prefix ncconf;
    }

    import ietf-yang-types {
      prefix yang;
    }
    import ietf-netconf-acm {
      prefix nacm;
    }
    import ietf-x509-cert-to-name {
      prefix x509c2n;
    }

    include ietf-netconf-common;

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

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

      WG Chair: Mehmet Ersue
                <mailto:mehmet.ersue@nsn.com>

      WG Chair: Bert Wijnen
                <mailto:bertietf@bwijnen.net>
```

```
       Editor:   Mohamad Badra
                 <mailto:mbadra@gmail.com>

                 Alan Luchuk
                 <mailto:luchuk@snmp.com>

                 Juergen Schoenwaelder
                 <mailto:j.schoenwaelder@jacobs-university.de>";

     description

      "This submodule applies to NETCONF over TLS.  It specifies how
       NETCONF servers transform X.509 certificates presented by
       NETCONF clients into NETCONF usernames.  It also specifies
       how NETCONF servers transform pre-shared TLS keys into NETCONF
       usernames.

       Copyright (c) 2013 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.";
     // RFC Ed.: replace XXXX with actual RFC number and
     // remove this note

     // RFC Ed.: please update the date to the date of publication

     revision "2013-05-07" {
       description
        "Initial version";
       reference
        "RFC XXXX: NETCONF over Transport Layer Security (TLS)";
     }

     feature tls {
       description
        "A server implements this feature if it supports NETCONF
         over Transport Layer Security (TLS).";
       reference
        "RFC XXXX: NETCONF over Transport Layer Security (TLS)";
     }
```

```
feature tls-map-certificates {
  description
   "The tls-map-certificates feature indicates that the
    server implements mapping X.509 certificates to NETCONF
    usernames.";
}

feature tls-map-pre-shared-keys {
  description
   "The tls-map-pre-shared-keys feature indicates that the
    server implements mapping TLS pre-shared keys to NETCONF
    usernames.";
}

augment /ncconf:netconf {
  if-feature tls;

  container tls {

    leaf enabled {
      type boolean;
      default "false";
      description
       "Enables NETCONF over Transport Layer Security (TLS).";
    }

    // Objects for deriving NETCONF usernames from X.509
    // certificates.

    container cert-maps {
      if-feature tls-map-certificates;
      uses x509c2n:cert-to-name;
      description
       "The cert-maps container is used by a NETCONF server to
        map the NETCONF client's presented X.509 certificate to
        a NETCONF username.

        If no matching and valid cert-to-name list entry can be
        found, then the NETCONF server MUST close the connection,
        and MUST NOT accept NETCONF messages over it.";
    }

    //  Objects for deriving NETCONF usernames from TLS
    //  pre-shared keys.

    container psk-maps {
      if-feature tls-map-pre-shared-keys;
      description
```

```
          "During the TLS Handshake, the client indicates which
           key to use by including a PSK identity in the TLS
           ClientKeyExchange message. On the server side, this
           PSK identity is used to look up an entry in the psk-map
           list. If such an entry is found, and the pre-shared keys
           match, then the client is authenticated. The server uses
           the value from the user-name leaf in the psk-map list as
           the NETCONF username.  If the server cannot find an entry
           in the psk-map list, or if the pre-shared keys do not
           match, then the server terminates the connection.";
         reference
          "RFC 4279: Pre-Shared Key Ciphersuites for Transport Layer
                     Security (TLS)";

         list psk-map {
           key psk-identity;

           leaf psk-identity {
             type string;
             description
              "The PSK identity encoded as a UTF-8 string. For
               details on how the PSK identity MAY be encoded in
               UTF-8, see section 5.1. of RFC 4279.";
             reference
              "RFC 4279: Pre-Shared Key Ciphersuites for Transport
                         Layer Security (TLS)";
           }
           leaf user-name {
             type nacm:user-name-type;
             mandatory true;
             description
              "The NETCONF username associated with this PSK
               identity.";
           }
           leaf not-valid-before {
             type yang:date-and-time;
             description
              "This PSK identity is not valid before the given date
               and time.";
           }
           leaf not-valid-after {
             type yang:date-and-time;
             description
              "This PSK identity is not valid before the given date
               and time.";
           }
           leaf key {
             type yang:hex-string;
```

```
              mandatory true;
              nacm:default-deny-all;
              description
               "The key associated with the PSK identity";
              reference
               "RFC 4279: Pre-Shared Key Ciphersuites for Transport
                          Layer Security (TLS)";
            }
          }  // list psk-map
        }  // container psk-maps
      }
    }
  }

  <CODE ENDS>
```

5. Usage Examples

5.1. Certificate Mapping Configuration Example

The following XML shows an example configuration mapping a specific X.509 certificate to a NETCONF username:

```
<netconf xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-config">
  <tls>
    <enabled>true</enabled>
    <cert-maps>
      <!-- Use a subject alt name field of a specific
           certificate as the NC username. -->
      <cert-to-name>
        <id>1</id>
        <fingerprint>11:0A:05:11:00</fingerprint>
        <map-type>x509c2n:san-any</map-type>
      </cert-to-name>
      <!-- Map a specific certificate to the NC username
           'Joe Cool'. -->
      <cert-to-name>
        <id>2</id>
        <fingerprint>11:0A:05:11:00</fingerprint>
        <map-type>x509c2n:specified</map-type>
        <name>Joe Cool</name>
      </cert-to-name>
    </cert-maps>
  </tls>
</netconf>
```

5.2. PSK Mapping Configuration Example

The following XML shows an example configuration mapping a pre-shared key to a NETCONF username:

```
<netconf xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-config">
  <tls>
    <enabled>true</enabled>
    <psk-maps>
      <psk-map>
        <psk-identity>a8gc8]klh59</psk-identity>
        <user-name>admin</user-name>
        <not-valid-before>2013-01-01T00:00:00-00:00</not-valid-before>
        <not-valid-after>2014-01-01T00:00:00-00:00</not-valid-after>
      </psk-map>
    </psk-maps>
  </tls>
</netconf>
```

6. Security Considerations

The security considerations described throughout [RFC5246] and [RFC6241] apply here as well.

This document in its current version does not support third-party authentication (e.g., backend Authentication, Authorization, and Accounting (AAA) servers) due to the fact that TLS does not specify this way of authentication and that NETCONF depends on the transport protocol for the authentication service. If third-party authentication is needed, SSH transport can be used.

An attacker might be able to inject arbitrary NETCONF messages via some application that does not carefully check exchanged messages. When the :base:1.1 capability is not advertised by both peers, an attacker might be able to deliberately insert the delimiter sequence]]>]]> in a NETCONF message to create a DoS attack. If the :base:1.1 capability is not advertised by both peers, applications and NETCONF APIs MUST ensure that the delimiter sequence]]>]]> never appears in NETCONF messages; otherwise, those messages can be dropped, garbled, or misinterpreted. More specifically, if the delimiter sequence is found in a NETCONF message by the sender side, a robust implementation of this document SHOULD warn the user that illegal characters have been discovered. If the delimiter sequence is found in a NETCONF message by the receiver side (including any XML attribute values, XML comments, or processing instructions), a robust

implementation of this document MUST silently discard the message without further processing and then stop the NETCONF session.

Finally, this document does not introduce any new security considerations compared to [RFC6242].

7. IANA Considerations

Based on the previous version of this document, RFC 5539, IANA has assigned a TCP port number (6513) in the "Registered Port Numbers" range with the service name "netconf-tls". This port will be the default port for NETCONF over TLS, as defined in Section 2.1.1. Below is the registration template following the rules in [RFC6335].

```
Service Name:           netconf-tls
Transport Protocol(s):  TCP
Assignee:               IESG <iesg@ietf.org>
Contact:                IETF Chair <chair@ietf.org>
Description:            NETCONF over TLS
Reference:              RFC XXXX
Port Number:            6513
```

This document requests that IANA assigns a TCP port number in the "Registered Port Numbers" range with the service name "netconf-tls-ch". This port will be the default port for NETCONF over TLS when the NETCONF server calls home, as defined in Section 2.1.2. Below is the registration template following the rules in [RFC6335].

```
Service Name:           netconf-tls-ch
Transport Protocol(s):  TCP
Assignee:               IESG <iesg@ietf.org>
Contact:                IETF Chair <chair@ietf.org>
Description:            NETCONF over TLS (call home)
Reference:              RFC XXXX
Port Number:            YYYY
```

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made.

```
URI: urn:ietf:params:xml:ns:yang:ietf-netconf-config
```

```
   Registrant Contact: The NETMOD WG of the IETF.
   XML: N/A, the requested URI is an XML namespace.
```

This document registers a YANG module in the YANG Module Names registry [RFC6020].

```
   name:         ietf-netconf-config
   namespace:    urn:ietf:params:xml:ns:yang:ietf-netconf-config
   prefix:       ncconf
   reference:    RFC XXXX
```

The document registers the following YANG submodules in the YANG Module Names registry [RFC6020].

```
   name:         ietf-netconf-common
   parent:       ietf-netconf-config
   reference:    RFC XXXX

   name:         ietf-netconf-tls
   parent:       ietf-netconf-config
   reference:    RFC XXXX
```

8. Acknowledgements

A significant amount of the text in Section 2.4 was lifted from [RFC4642].

The authors like to acknowledge Martin Bjorklund, Olivier Coupelon, Mehmet Ersue, Miao Fuyou, David Harrington, Alfred Hoenes, Simon Josefsson, Eric Rescorla, Dan Romascanu, Bert Wijnen and the NETCONF mailing list members for their comments on this document. Charlie Kaufman, Pasi Eronen, and Tim Polk provided a the thorough review of previous versions of this document.

Juergen Schoenwaelder and was partly funded by Flamingo, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

9. Contributor's Address

Ibrahim Hajjeh
Ineovation
France

EMail: ibrahim.hajjeh@ineovation.fr

10. References

10.1. Normative References

[I-D.ietf-netmod-rfc6021-bis]
Schoenwaelder, J., "Common YANG Data Types", draft-ietf-netmod-rfc6021-bis-02 (work in progress), May 2013.

[I-D.ietf-netmod-snmp-cfg]
Bjorklund, M. and J. Schoenwaelder, "A YANG Data Model for SNMP Configuration", draft-ietf-netmod-snmp-cfg-02 (work in progress), April 2013.

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

[RFC4279] Eronen, P. and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

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

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

[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.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, August 2011.

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

10.2. Informative References

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC4642] Murchison, K., Vinocur, J., and C. Newman, "Using Transport Layer Security (TLS) with Network News Transfer Protocol (NNTP)", RFC 4642, October 2006.

[RFC4742] Wasserman, M. and T. Goddard, "Using the NETCONF Configuration Protocol over Secure SHell (SSH)", RFC 4742, December 2006.

[RFC5539] Badra, M., "NETCONF over Transport Layer Security (TLS)", RFC 5539, May 2009.

Appendix A. Change Log (to be removed by RFC Editor before publication)

A.1. draft-ietf-netconf-rfc5539bis-03

- Added support for call home (allocation of a new port number, rewrote text to allow a NETCONF client to be a TLS server and a NETCONF server to be a TLS client).
- Merged sections 2 and 3 into a new section 2 and restructured the text.
- Extended the IANA considerations section.

- Using the cert-to-name mapping grouping from the SNMP configuration data model and updated the examples.

- Creating an extensible set of YANG (sub)modules for NETCONF following the (sub)module structure of the SNMP configuration model.

A.2. draft-ietf-netconf-rfc5539bis-02

- Addressed remaining issues identified at IETF 85

  * Harmonized the cert-maps container of the YANG module in this draft with the tlstm container in the ietf-snmp-tls sub-module specified in draft-ietf-netmod-snmp-cfg. Replaced the children of the cert-maps container with the children copied from the tlstm container of the ietf-snmp-tls sub-module.

  * Added an overview of data model in the ietf-netconf-tls YANG module.

  * Added example configurations.

- Addessed issues posted on NETCONF WG E-mail list.

- Deleted the superfluous tls container that was directly below the netconf-config container.

- Added a statement to the text indicating that support for mapping X.509 certificates to NETCONF usernames is optional. This is analogous to existing text indicating that support for mapping pre-shared keys to NETCONF usernames is optional. Resource-constrained systems now can omit support for mapping X.509 certificates to NETCONF usernames and still comply with this specification.

- Clarified the document structure by promoting the sections of the document related to the data model.

- Updated author's addresses.

A.3. draft-ietf-netconf-rfc5539bis-00

- Remove the reference to BEEP.

- Rename host-part to domain-part in the description of RFC822.

Authors' Addresses

Mohamad Badra
LIMOS Laboratory

Email: mbadra@gmail.com

Alan Luchuk
SNMP Research, Inc.
3001 Kimberlin Heights Road
Knoxville, TN 37920
US

Phone: +1 865 573 1434
Email: luchuk@snmp.com
URI: http://www.snmp.com/

Juergen Schoenwaelder
Jacobs University Bremen
Campus Ring 1
28759 Bremen
Germany

Phone: +49 421 200 3587
Email: j.schoenwaelder@jacobs-university.de
URI: http://www.jacobs-university.de/