Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication
draft-ietf-netconf-rfc5539bis-09

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 with mutual X.509 authentication to secure the exchange of NETCONF messages. This revision of RFC 5539 documents the new message framing used by NETCONF 1.1 and it 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 August 16, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must 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 .............................................................. 2
2. Connection Initiation ...................................................... 3
3. Message Framing ............................................................. 3
4. Connection Closure .......................................................... 3
5. Certificate Validation ...................................................... 4
6. Server Identity .............................................................. 4
7. Client Identity .............................................................. 4
8. Cipher Suites ................................................................. 6
9. Security Considerations .................................................... 6
10. IANA Considerations ....................................................... 7
11. Acknowledgements .......................................................... 7
12. References ................................................................. 7
    12.1. Normative References ................................................ 8
    12.2. Informative References .............................................. 8
Appendix A. Changes from RFC 5539 .......................................... 9
Appendix B. Change Log .......................................................... 9
    B.1. draft-ietf-netconf-rfc5539bis-07 .................................. 9
    B.2. draft-ietf-netconf-rfc5539bis-06 .................................. 9
    B.3. draft-ietf-netconf-rfc5539bis-05 .................................. 10
    B.4. draft-ietf-netconf-rfc5539bis-04 .................................. 10
    B.5. draft-ietf-netconf-rfc5539bis-03 .................................. 10
    B.6. draft-ietf-netconf-rfc5539bis-02 .................................. 10
    B.7. draft-ietf-netconf-rfc5539bis-00 .................................. 11
Authors' Addresses ............................................................ 11

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 how NETCONF messages can be exchanged over Transport Layer Security (TLS) [RFC5246]. Implementations MUST
support mutual TLS certificate-based authentication [RFC5246]. This assures the NETCONF server of the identity of the principal who wishes to manipulate the management information. It also assures the NETCONF client of the identity of the server for which it wishes to manipulate the management information.

2. Connection Initiation

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 connections. The well-known TCP port number 6513 is used by NETCONF servers to listen for TCP connections established by NETCONF over TLS clients. The TLS client MUST 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 is done before the NETCONF <hello> message is sent. This means that the identity verification is completed before the NETCONF session is started.

3. 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 NETCONF <hello> message MUST be followed by the character sequence ]]>]]>. Upon reception of the <hello> message, the peers inspect the announced capabilities. 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.

4. 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 Section 7.2.1 of [RFC5246].

5. Certificate Validation

Both peers MUST use X.509 certificate path validation [RFC5280] to verify the integrity of the certificate presented by the peer. The presented X.509 certificate may also be considered valid if it matches a locally configured certificate fingerprint. If X.509 certificate path validation fails and the presented X.509 certificate does not match a locally configured certificate fingerprint, the connection MUST be terminated as defined in [RFC5246].

6. Server Identity

The NETCONF client MUST check the identity of the server according to Section 6 of [RFC6125].

7. 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 results 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 following algorithm is used by the NETCONF server to derive a NETCONF username from a certificate. (Note that the algorithm below is the same as the one described in the SNMP-TLS-TM-MIB MIB module defined in [RFC6353] and in the ietf-x509-cert-to-
name YANG module defined in [RFC7407].

(a) The server maintains an ordered list of mappings of certificates to NETCONF usernames. Each list entry contains

* a certificate fingerprint (used for matching the presented certificate),

* a map type (indicates how the NETCONF username is derived from the certificate), and

* optional auxiliary data (used to carry a NETCONF username if the map type indicates the user name is explicitly configured).

(b) The NETCONF username is derived by considering each list entry in order. The fingerprint member of the current list entry determines whether the current list entry is a match:

1. If the list entry's fingerprint value matches the fingerprint of the presented certificate, then consider the list entry as a successful match.

2. If the list entry's fingerprint value matches that of a locally held copy of a trusted CA certificate, and that CA certificate was part of the CA certificate chain to the presented certificate, then consider the list entry as a successful match.

(c) Once a matching list entry has been found, the map type of the current list entry is used to determine how the username associated with the certificate should be determined. Possible mapping options are:

A. The username is taken from the auxiliary data of the current list entry. This means the username is explicitly configured (map type 'specified').

B. The subjectAltName's rfc822Name field is mapped to the username (map type 'san-rfc822-name'). The local part of the rfc822Name is used unaltered but the host-part of the
name must be converted to lowercase.

C. The subjectAltName's dNSName is mapped to the username (map type 'san-dns-name'). The characters of the dNSName are converted to lowercase.

D. The subjectAltName's iPAddress is mapped to the username (map type 'san-ip-address'). IPv4 addresses are converted into decimal-dotted quad notation (e.g., '192.0.2.1'). IPv6 addresses are converted into a 32-character all lowercase hexadecimal string without any colon separators.

E. Any of the subjectAltName's rfc822Name, dNSName, iPAddress is mapped to the username (map type 'san-any'). The first matching subjectAltName value found in the certificate of the above types MUST be used when deriving the name.

F. The certificate's CommonName is mapped to the username (map type 'common-name'). The CommonName is converted to UTF-8 encoding. The usage of CommonNames is deprecated and users are encouraged to use subjectAltName mapping methods instead.

(d) If it is impossible to determine a username from the list entry's data combined with the data presented in the certificate, then additional list entries MUST be searched looking for another potential match. Similarly, if the username does not comply to the NETCONF requirements on usernames [RFC6241] (i.e., the username is not representable in XML), then additional list entries MUST be searched looking for another potential match. If there are no further list entries, the TLS session MUST be terminated.

The username provided by the NETCONF over TLS implementation will be made available to the NETCONF message layer as the NETCONF username without modification.

8. Cipher Suites

Implementations MUST support TLS 1.2 [RFC5246] and are REQUIRED to support the mandatory-to-implement cipher suite. Implementations MAY
implement additional TLS cipher suites that provide mutual authentication [RFC5246] and confidentiality as required by NETCONF [RFC6241]. Implementations SHOULD follow the recommendations given in [I-D.ietf-uta-tls-bcp].

9. Security Considerations

NETCONF is used to access configuration and state information and to modify configuration information, so the ability to access this protocol should be limited to users and systems that are authorized to view the NETCONF server's configuration and state or to modify the NETCONF server's configuration.

Configuration or state data may include sensitive information, such as usernames or security keys. So, NETCONF requires communications channels that provide strong encryption for data privacy. This document defines a NETCONF over TLS mapping that provides for support of strong encryption and authentication. The security considerations for TLS [RFC5246] and NETCONF [RFC6241] apply here as well.

NETCONF over TLS requires mutual authentication. Neither side should establish a NETCONF over TLS connection with an unknown, unexpected, or incorrect identity on the opposite side. This document 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, the SSH transport can be used.

RFC 5539 assumes that the end-of-message (EOM) sequence, ]]>]>], cannot appear in any well-formed XML document, which turned out to be mistaken. The EOM sequence can cause operational problems and open space for attacks if sent deliberately in NETCONF messages. It is however believed that the associated threat is not very high. This document still uses the EOM sequence for the initial <hello> message to avoid incompatibility with existing implementations. When both peers implement :base:1.1 capability, a proper framing protocol (chunked framing mechanism; see Section 3) is used for the rest of the NETCONF session, to avoid injection attacks.

10. 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. 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

[[CREF1: RFC Editor: Please replace XXXX above with the allocated RFC number and remove this comment. --JS]]

11. Acknowledgements

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

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

12. References


Internet-Draft NETCONF over TLS February 2015

12.1. Normative References
12.2. Informative References


Appendix A. Changes from RFC 5539

This section summarizes major changes between this document and RFC 5539.

- Documented that NETCONF over TLS uses the new message framing if both peers support the :base:1.1 capability.
- Removed redundant text that can be found in the TLS and NETCONF specifications and restructured the text. Alignment with [RFC6125].
- Added a high-level description how NETCONF usernames are derived from certificates.
- Removed the reference to BEEP.

Appendix B. Change Log

[[CREF2: RFC Editor: Please remove this appendix before publication. --JS]]

B.1. draft-ietf-netconf-rfc5539bis-07

- Limited the scope of the document to TLS with mutual X.509 authentication.
- Added a high-level description how NETCONF usernames are extracted from certificates.
- Editorial changes

B.2. draft-ietf-netconf-rfc5539bis-06

- Removed all call-home related text.
- Removed redundant text as discussed at the Toronto IETF meeting.
B.3.  draft-ietf-netconf-rfc5539bis-05

  o Removed the YANG configuration data model since it became a separate document.

  o Added reference to RFC 3234 plus editorial updates.

B.4.  draft-ietf-netconf-rfc5539bis-04

  o Added the applicability statement proposed by Stephen Hanna.

  o Added call-home configuration objects and a tls-call-home feature.

  o Rewrote the text such that the role swap happens right after the TCP connection has been established.

B.5.  draft-ietf-netconf-rfc5539bis-03

  o 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).

  o Merged sections 2 and 3 into a new section 2 and restructured the text.

  o Extended the IANA considerations section.

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

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

B.6.  draft-ietf-netconf-rfc5539bis-02

  o 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.


Internet-Draft NETCONF over TLS February 2015

- Addressed 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.

B.7. 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
Zayed University

Email: mbadra@gmail.com