Network Working Group                                           M. Badra
Request for Comments: 5539                         CNRS/LIMOS Laboratory
Category: Standards Track                                       May 2009

# NETCONF over Transport Layer Security (TLS)

## Status of This Memo



## Copyright Notice



## 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 NETCONF exchanges.

Table of Contents

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.1. Conventions Used in This Document . . . . . . . . . . . . . 2
2. NETCONF over TLS . . . . . . . . . . . . . . . . . . . . . . . 3
2.1. Connection Initiation . . . . . . . . . . . . . . . . . . . 3
2.2. Connection Closure . . . . . . . . . . . . . . . . . . . . 4
3. Endpoint Authentication and Identification . . . . . . . . . . 4
3.1. Server Identity . . . . . . . . . . . . . . . . . . . . . . 4
3.2. Client Identity . . . . . . . . . . . . . . . . . . . . . . 5
4. Security Considerations . . . . . . . . . . . . . . . . . . . 5
5. IANA Considerations . . . . . . . . . . . . . . . . . . . . . 6
6. Acknowledgements . . . . . . . . . . . . . . . . . . . . . . . 6
7. Contributor's Address . . . . . . . . . . . . . . . . . . . . 7
8. References . . . . . . . . . . . . . . . . . . . . . . . . . 7
8.1. Normative References . . . . . . . . . . . . . . . . . . . 7
8.2. Informative References . . . . . . . . . . . . . . . . . . 7

## 1. Introduction

The NETCONF protocol [RFC4741] 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-based mutual 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].

Throughout this document, the terms "client" and "server" are used to refer to the two ends of the TLS connection. The client actively opens the TLS connection, and the server passively listens for the incoming TLS connection. The terms "manager" and "agent" are used to refer to the two ends of the NETCONF protocol session. The manager issues NETCONF remote procedure call (RPC) commands, and the agent replies to those commands. When NETCONF is run over TLS using the mapping defined in this document, the client is always the manager, and the server is always the agent.

### 1.1. Conventions Used in This Document

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

The peer acting as the NETCONF manager MUST also act as the TLS client. It MUST connect to the server that passively listens for the incoming TLS connection on the TCP port 6513. It 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 data. In particular, the client will send complete XML documents to the server containing <rpc> elements, and the server will respond with complete XML documents containing <rpc-reply> elements. The client MAY indicate interest in receiving event notifications from a server by creating a subscription to receive event notifications [RFC5277]. In this case, the server replies to indicate whether the subscription request was successful and, if it was successful, the server begins sending the event notifications to the client as the events occur within the system.

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.

This document uses the same delimiter sequence ("]]>]]>") defined in [RFC4742], which MUST be sent by both the client and the server after each XML document in the NETCONF exchange. Since this character sequence can legally appear in plain XML in attribute values, comments, and processing instructions, implementations of this document MUST ensure that this character sequence is never part of a NETCONF message.

Implementation of the protocol specified in this document MAY implement any TLS cipher suite that provides certificate-based mutual authentication [RFC5246]. The server MUST support certificate-based client authentication.

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.

## 2.2. Connection Closure

A TLS client (NETCONF manager) MUST close the associated TLS connection if the connection is not expected to issue any NETCONF RPC commands later. It MUST send a TLS close_notify alert before closing the connection. The TLS client MAY choose to not wait for the TLS server (NETCONF agent) close_notify alert and simply close the connection, thus generating an incomplete close on the TLS server side. Once the TLS server gets a close_notify from the TLS client, it MUST reply with a close_notify unless it becomes aware that the connection has already been closed by the TLS client (e.g., the closure was indicated by TCP).

When no data is received from a connection for a long time (where the application decides what "long" means), a NETCONF peer MAY close the connection. The NETCONF peer MUST attempt to initiate an exchange of close_notify alerts with the other NETCONF peer before closing the connection. The close_notify's sender that is unprepared to receive any more data MAY close the connection after sending the close_notify alert, thus generating an incomplete close on the close_notify's receiver side.

# 3. Endpoint Authentication and Identification

## 3.1. Server Identity

During the TLS negotiation, the client MUST carefully examine the certificate presented by the server to determine if it meets the client's expectations. Particularly, the client MUST check its understanding of the 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 below (following the example of [RFC4642]):

- The client MUST use the server hostname it used to open the connection (or the hostname specified in the TLS "server_name" extension [RFC5246]) as the value to compare against the server name as expressed in the server certificate. The client MUST NOT use any form of the server hostname derived from an insecure remote source (e.g., insecure DNS lookup). CNAME canonicalization is not done.

- If a subjectAltName extension of type dNSName is present in the certificate, it MUST be used as the source of the server's identity.

- Matching is case-insensitive.

- A "*" wildcard character MAY be used as the leftmost name component in the certificate. For example, *.example.com would match a.example.com, foo.example.com, etc., but would not match example.com.

- If the certificate contains multiple names (e.g., more than one dNSName field), then a match with any one of the fields is considered acceptable.

If the match fails, the client MUST either ask for explicit user confirmation or terminate the connection and indicate the server's identity is suspect.

Additionally, clients MUST verify the binding between the identity of the servers to which they connect and the public keys presented by those servers. 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 server certificate against a local store of already-verified certificates and identity bindings).

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

### 3.2. Client Identity

The server MUST verify the identity of the client with certificate-based authentication according to local policy to ensure that the incoming client request is legitimate before any configuration or state data is sent to or received from the client.

## 4. Security Considerations

The security considerations described throughout [RFC5246] and [RFC4741] 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, BEEP or 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 or deliberately insert the delimiter sequence in a NETCONF message to create a DoS attack. Hence, applications and NETCONF APIs MUST ensure that the delimiter sequence defined in Section 2.1 never appears in NETCONF messages; otherwise, those messages can be dropped, garbled, or misinterpreted. 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 [RFC4742].

## 5. IANA Considerations

IANA has assigned a TCP port number (6513) in the "Registered Port Numbers" range with the name "netconf-tls". This port will be the default port for NETCONF over TLS, as defined in this document.

Registration Contact: Mohamad Badra, badra@isima.fr.
Transport Protocol: TCP.
Port Number: 6513
Broadcast, Multicast or Anycast: No.
Port Name: netconf-tls.
Service Name: netconf.
Reference: RFC 5539

## 6. Acknowledgements

A significant amount of the text in Section 3 was lifted from [RFC4642].

The author would like to acknowledge David Harrington, Miao Fuyou, Eric Rescorla, Juergen Schoenwaelder, Simon Josefsson, Olivier Coupelon, Alfred Hoenes, and the NETCONF mailing list members for their comments on the document. The author also appreciates Bert Wijnen, Mehmet Ersue, and Dan Romascanu for their efforts on issues resolving discussion; and Charlie Kaufman, Pasi Eronen, and Tim Polk for the thorough review of this document.

## 7. Contributor's Address

Ibrahim Hajjeh
Ineovation
France

EMail: ibrahim.hajjeh@ineovation.fr

## 8. References

### 8.1. Normative References

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

[RFC4741] Enns, R., "NETCONF Configuration Protocol", RFC 4741, December 2006.

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

[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.

### 8.2. Informative References

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

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, July 2008.

## Author's Address

Mohamad Badra
CNRS/LIMOS Laboratory
Campus de cezeaux, Bat. ISIMA
Aubiere 63170
France

EMail: badra@isima.fr