Internet Engineering Task Force
INTERNET DRAFT

M. Badra
LIMOS Laboratory

October 10, 2007

Expires: April 2008

# NETCONF over TLS
# <draft-badra-tls-netconf-04.txt>

Status of this Memo



This Internet-Draft will expire on April 2008.

Copyright Notice



Abstract

The NETCONF configuration protocol provides mechanisms to install, manipulate, and delete the configuration of network devices. This document describes how to use TLS to secure NETCONF exchanges.

## 1 Introduction

The NETCONF protocol [NETCONF] defines a simple mechanism through which a network device can be managed. NETCONF is connection-

oriented, requiring a persistent connection between peers. This connection must provide reliable, sequenced data delivery, integrity and confidentiality and peers authentication. This document describes how to use TLS [TLS] to secure NETCONF connections.

### 1.2 Requirements language and Terminologies

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 [KEYWORDS].

### 1.3 Terminology

This document uses the following terms:

manager
It refers to the end initiating the NETCONF connection. It issues the NETCONF RPC commands.

agent
It refers to the end replying to the manager's commands during the NETCONF connection.

## 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 Transport Layer Security (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 IANA-to-be-assigned TCP port <TBC>. It MUST therefore send the TLS ClientHello to begin the TLS handshake. Once the TLS handshake has been finished, the manager and the agent MAY then send their NETCONF exchanges. In particular, the manager will send complete XML documents to the server containing <rpc> elements, and the agent will respond with complete XML documents containing <rpc-reply> elements. The client MAY indicate interest in receiving event notifications from a NETCONF server by creating a subscription to receive event notifications [NETNOT], in which the NETCONF server replies to indicate whether the subscription request was successful and, if it was successful, begins sending the event notifications to the NETCONF client as the events occur within the system. All these elements are encapsulated into TLS records of type "application data". These records are protected using the TLS material keys.

Current NETCONF messages don't include a message's length. This document uses consequently the same delimiter sequence defined in [NETSSH] and therefore the special character sequence, ]]>]]>, to delimit XML documents.

## 2.2. Connection Closure

Either NETCONF peer MAY stop the NETCONF connection at any time and therefore notify the other NETCONF peer that no more data on this channel will be sent and that any data received after a closure request will be ignored. This MAY happen when no data is received from a connection for a long time, where the application decides what "long" means.

TLS has the ability for secure connection closure using the Alert protocol. When the NETCONF peer processes a closure request of the NETCONF connection, it MUST send a TLS close_notify alert before closing the connection. Any data received after a closure alert is ignored.

Unless some other fatal alert has been transmitted, each party is required to send a close_notify alert before closing the write side of the connection. The other party MUST respond with a close_notify alert of its own and close down the connection immediately, discarding any pending writes. It is not required for the initiator of the close to wait for the responding close_notify alert before closing the read side of the connection.

# 3. Endpoint Authentication and Identification

Usually, TLS uses public keys, Kerberos [TLSKERB], or preshared keys [TLSPSK] for authentication.

When public key is used for authentication, TLS supports three authentication modes: authentication of both parties, server authentication with an unauthenticated client, and total anonymity. User authentication in unauthenticated or authenticated client mode is outside the scope of this document. User authentication should be handled by either an extension of TLS (such as the TLS Inner Application Extension [IATLS]) or an authentication extension of NETCONF.

## 3.1. Server Identity

During the TLS negotiation, the client MUST carefully examine the certificate presented by the server to determine if it meets their 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 these rules [RFC4642]:

- The client MUST use the server hostname it used to open the connection (or the hostname specified in TLS "server_name" extension [TLSEXT]) 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 left-most 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 [PKICERT] 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

Typically, the server has no external knowledge of what the client's identity ought to be and so checks (other than that the client has a certificate chain rooted in an appropriate CA) are not possible. If

a server has such knowledge (typically from some source external to NETCONF or TLS) it MUST check the identity as described above.

## 4. Security Considerations

The security considerations described throughout [TLS] apply here as well.

## 5. IANA Considerations

IANA is requested to assign a TCP port number that will be the default port for NETCONF over TLS sessions as defined in this document.

IANA has assigned port <TBD> for this purpose.

## 6. Acknowledgment

The author would like to acknowledge Eric Rescorla and Juergen Schoenwaelder for their detailed reviews of the content of the document. The author appreciates also David Harrington, Miao Fuyou and Dan Romascanu for their effort on issues resolving discussion.

## 7. References

### 7.1. Normative References

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

[TLS] Dierks, T. and E. Rescorla, "The TLS Protocol Version 1.1", RFC 4346, April 2005.

[TLSEXT] Blake-Wilson, S., et. al., "Transport Layer Security (TLS) Extensions", RFC 4346, April 2006.

[TLSPSK] Eronen, P., et. al., "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

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

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

[PKICERT] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280,

April 2002.

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

[NETNOT] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", draft-ietf-netconf-notification-09.txt, (work in progress), September 2007.

## 7.2. Informative References

[TLSKERB] Medvinsky, A. and M. Hur, "Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)", RFC 2712, October 1999.

[IATLS] Funk, P., et. al., "TLS Inner Application Extension (TLS/IA)", draft-funk-tls-inner-application-extension-03.txt (work in progress), June 2006.

Author's Addresses

Mohamad Badra
LIMOS Laboratory - UMR (6158), CNRS
France Email: badra@isima.fr

rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).