Network Working Group
Internet-Draft
Expires: January 30, 2004

E. Lear
K. Crozier
Cisco Systems
R. Enns
Juniper Networks
August 2003

NETCONF Transport over BEEP
draft-lear-netconfbeep-00

Status of this Memo

Copyright Notice

Abstract

This document specifies a transport mapping for the NETCONF protocol over the Blocks Extensible Exchange Protocol (BEEP).

Table of Contents

1. Introduction

The NETCONF protocol [1] defines a simple mechanism through which a network device can be managed. NETCONF is designed to be usable over a variety of transport protocols. This document specifies a transport mapping over the Blocks Extensible Exchange Protocol (BEEP) [2] .

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 RFC 2119 [3].

2. BEEP Transport Mapping

All NETCONF over BEEP implementations MUST implement the profile and functional mapping between NETCONF and BEEP as described below.

2.1 NETCONF Session Initiation

Managers may be either BEEP listeners or initiators. Similarly, agents may be either listeners or initiators. Thus the initial exchange takes place without regard to whether a manager or the agent is the initiator. After the transport connection is established, as greetings are exchanged, they should each announce their support for TLS [5] and optionally SASL [4] (see below), as well as for the SYSLOG profile [6]. Once greetings are exchanged, if TLS is to be used and available by both parties, the listener STARTs a channel with the TLS profile.

Once TLS has been started, a new greeting is sent by both initiator and listener, as required by the BEEP RFC.

At this point, if SASL is desired, the initiator starts BEEP channel 1 to perform a SASL exchange to authenticate itself. When SASL is completed, the channel MUST be closed.

Once authentication has occurred, there is no need to distinguish between initiator and listener. We now distinguish between manager and agent.

The manager now establishes an NETCONF management channel for the purpose of exchanging capabilities, monitoring progress, and aborting remote procedure calls. As initiators assign odd channels and listeners assign even channels, the management channel is BEEP channel 1 or 2, depending on whether the manager is the initiator or the listener.

The manager next establishes the NETCONF operational channel for the purpose of issuing RPC requests. This channel is BEEP channel 3 or 4.

Finally, if either manager or agent wishes to send or receive notifications, it may issue a start on the next available channel if the other side has sent the send or receive NETCONF capability.

At this point, the NETCONF session is established.

2.2 NETCONF RPC Execution

To issue an RPC, the manager transmits on the operational channel a

BEEP MSG containing the RPC and its arguments. In accordance with the BEEP standard, RPC requests may be split across multiple BEEP frames.

Once received and processed, the agent responds with BEEP RPYs on the same channel with the response to the RPC. In accordance with the BEEP standard, responses may be split across multiple BEEP frames.

2.3 NETCONF <rpc-abort> and <rpc-progress>

<rpc-abort> and <rpc-progress> requests are issued by the manager on the NETCONF management channel, and the agent responds with BEEP RPYs on that same channel.

2.4 NETCONF Session Teardown

Either side may initiate the termination of an NETCONF session. In This is done by issuing a BEEP close on the operational channel after the current RPC has completed. The same is done with any notification channels by the end that transmits notifications. Finally, BEEP channel 0 is closed.

2.5 BEEP Profiles for NETCONF Channels

There are two profiles, the management channel profile and the operations channel profile. These are not to be confused with the BEEP control channel.

The operations channel will have two commands, <rpc> and <rpc-reply>. The management channel will have one additional operation with <rpc-progress>.

2.5.1 Management Channel Profile

```
<!-- DTD for netconf management over BEEP

  Refer to this DTD as:

    <!ENTITY % NETCONF PUBLIC "netconf/management/1.0" "">
    %NETCONF;
  -->

<!--   Contents

    Overview

    Includes
    Profile Summaries
```

```
          Entity Definitions

          Operations
           rpc
           rpc-reply
           rpc-progress
        -->

      <!--  Overview   NETCONF Management channel  -->

      <!-- Includes -->

             <!ENTITY % BEEP PUBLIC "-//Blocks//DTD BEEP//EN"
                        "">
             %BEEP;

      <!--  Profile summaries


          BEEP profile NETCONF-MANAGEMENT

          role        MSG                       RPY        ERR
          ====        ===                       ===        ===
          I or L      rpc                       ok         error
          I or L      rpc-reply         ok         error
          I or L      rpc-progress              ok         error

      -->

      <!--
        Entity Definitions

              entity        syntax/reference     example
              ======        ================     =======

        a PRC
           RPC-DATA       Alpha
        a RPC reply number
              RPC-REPLY      1*3DIGIT
        a RPC progress number
              RPC-PROGRESS   1*3DIGIT

      -->

      <!ENTITY % RPC-REPLY    "CDATA">
      <!ENTITY % RPC-DATA     "CDATA">
      <!ENTITY % RPC-PROGRESS "CDATA">
       -->
```

```
      <!--
        RPC command
        -->

      <!ELEMENT rpc        (#PCDATA)>
      <!ATTLIST rpc
                rpc-data       %RPC_DATA;                #REQUIRED>


   <!--
        Result of RPC.
        -->

      <!ELEMENT rpc-reply    (#PCDATA)>
      <!ATTLIST rpc-reply
                rpc-reply      %RPC-REPLY;               #REQUIRED
             rpc-data    %rpc-data                #REQUIRED>

   <!--
        Progress of RPC operation.
        -->

      <!ELEMENT rpc-progress   (#PCDATA)>
      <!ATTLIST rpc-progress
                rpc-progress %RPC-PROGRESS;               #REQUIRED>

      <!-- End of DTD -->
```

2.5.2 Operations Channel Profile

```
      <!-- DTD for netconf operations over BEEP

        Refer to this DTD as:

          <!ENTITY % NETCONF PUBLIC "netconf/Operation/1.0" "">
          %NETCONF;
        -->

      <!--   Contents

          Overview

          Includes
          Profile Summaries
          Entity Definitions

          Operations
```

```
          rpc
          rpc-reply
        -->

      <!--  Overview   NETCONF operation channel  -->

      <!-- Includes -->

            <!ENTITY % BEEP PUBLIC "-//Blocks//DTD BEEP//EN"
                       "">
            %BEEP;

      <!--  Profile summaries


          BEEP profile NETCONF-MANAGEMENT

          role        MSG                       RPY        ERR
          ====        ===                       ===        ===
          I or L      rpc                               ok         error
          I or L      rpc-reply                 ok         error

      -->

      <!--
        Entity Definitions

              entity        syntax/reference     example
              ======        ================     =======

        a PRC
           RPC-DATA       Alpha
        a RPC reply number
              RPC-REPLY      1*3DIGIT

      -->

      <!ENTITY % RPC-REPLY    "CDATA">
      <!ENTITY % RPC-DATA     "CDATA">

       -->

      <!--
        RPC command
        -->

      <!ELEMENT RPC        (#PCDATA)>
      <!ATTLIST RPC
```

```
               RPC-DATA        %RPC_DATA;                #REQUIRED>


   <!--
        Result of RPC.
        -->

      <!ELEMENT RPC-REPLY    (#PCDATA)>
      <!ATTLIST RPC-REPLY
                RPC-REPLY       %RPC-REPLY;               #REQUIRED
                RPC-DATA    %RPC-DATA             #REQUIRED>


      <!-- End of DTD -->
```

2.5.3 Notification Channel Profile

The NETCONF notification channel profile is defined in RFC 3195 [6].

3. Security Considerations

Configuration information is by its very nature sensitive. Its transmission in the clear and without integrity checking leaves devices open to classic so-called "person in the middle" attacks. Configuration information often times contains passwords, user names, service descriptions, and topological information, all of which are sensitive. A NETCONF transport protocol, therefore, must minimally support options for both confidentiality and authentication.

BEEP makes use of both transport layer security and SASL. We require that TLS be used in BEEP as described by the BEEP standard. Client-side certificates are strongly desirable, but an SASL authentication is the bare minimum. SASL allows for the use of protocols such as RADIUS [9], so that authentication can occur off the box.

SASL authentication will occur on the first channel creation. No further authentication may occur during the same session. This avoids a situation where rights are different between different channels. If an implementation wishes to support multiple accesses by different individuals with different rights, then multiple sessions are required.

Different environments may well allow different rights prior to and then after authentication. Thus, an authorization model is not specified in this document. When an operation is not properly authorized then a simple "permission denied" is sufficient. Note that authorization information may be exchanged in the form of configuration information, which is all the more reason to ensure the security of the connection.

Normative References

[1] Enns, R., "NETCONF Configuration Protocol", draft-ietf-netconf-prot-00 (work in progress), August 2003.

[2] Rose, M., "The Blocks Extensible Exchange Protocol Core", RFC 3080, March 2001.

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

[4] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997.

[5] Dierks, T., Allen, C., Treese, W., Karlton, P., Freier, A. and P. Kocher, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[6] New, D. and M. Rose, "Reliable Delivery for syslog", RFC 3195, November 2001.

Informative References

[7] Bray, T., Paoli, J., Sperberg-McQueen, C. and E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C REC REC-xml-20001006, October 2000.

[8] Hollenbeck, S., Rose, M. and L. Masinter, "Guidelines for the Use of Extensible Markup Language (XML) within IETF Protocols", BCP 70, RFC 3470, January 2003.

[9] Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

Authors' Addresses

Eliot Lear
Cisco Systems
170 W. Tasman Dr.
San Jose, CA 95134-1706
US

EMail: lear@cisco.com

Ken Crozier
Cisco Systems
170 W. Tasman Dr.
San Jose, CA 95134-1706
US

EMail: kcrozier@cisco.com

Rob Enns
Juniper Networks
1194 North Mathilda Ave
Sunnyvale, CA 94089
US

EMail: rpe@juniper.net

Intellectual Property Statement

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.