Internet Engineering Task Force (IETF)
Request for Comments: 6470
Category: Standards Track
ISSN: 2070-1721

A. Bierman
Brocade

February 2012

Network Configuration Protocol (NETCONF) Base Notifications

Abstract

The Network Configuration Protocol (NETCONF) provides mechanisms to manipulate configuration datastores. However, client applications often need to be aware of common events, such as a change in NETCONF server capabilities, that may impact management applications. Standard mechanisms are needed to support the monitoring of the base system events within the NETCONF server. This document defines a YANG module that allows a NETCONF client to receive notifications for some common system events.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6470.

Table of Contents

1. Introduction

The NETCONF protocol [RFC6241] provides mechanisms to manipulate configuration datastores. However, client applications often need to be aware of common events, such as a change in NETCONF server capabilities, that may impact management applications. Standard mechanisms are needed to support the monitoring of the base system events within the NETCONF server. This document defines a YANG module [RFC6020] that allows a NETCONF client to receive notifications for some common system events.

1.1. Terminology

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].

The following terms are defined in [RFC6241]:

- client
- datastore
- protocol operation
- server

The following terms are defined in [RFC5277]:

- event
- stream
- subscription

The following term is defined in [RFC6020]:

- data node

2. YANG Module for NETCONF Base Notifications

2.1. Overview

The YANG module defined within this document specifies a small number of event notification messages for use within the 'NETCONF' stream, and accessible to clients via the subscription mechanism described in [RFC5277]. This module imports data types from the 'ietf-netconf' module defined in [RFC6241] and 'ietf-inet-types' module defined in [RFC6021].

These notifications pertain to configuration and monitoring portions of the managed system, not the entire system. A server MUST report events that are directly related to the NETCONF protocol. A server MAY report events for non-NETCONF management sessions, using the 'session-id' value of zero.

This module defines the following notifications for the 'NETCONF' stream to notify a client application that the NETCONF server state has changed:

netconf-config-change:
: Generated when the NETCONF server detects that the <running> or <startup> configuration datastore has been changed by a management session. The notification summarizes the edits that have been detected.

netconf-capability-change:
: Generated when the NETCONF server detects that the server capabilities have changed. Indicates which capabilities have been added, deleted, and/or modified. The manner in which a server capability is changed is outside the scope of this document.

netconf-session-start:
: Generated when a NETCONF server detects that a NETCONF session has started. A server MAY generate this event for non-NETCONF management sessions. Indicates the identity of the user that started the session.

netconf-session-end:
: Generated when a NETCONF server detects that a NETCONF session has terminated. A server MAY optionally generate this event for non-NETCONF management sessions. Indicates the identity of the user that owned the session, and why the session was terminated.

netconf-confirmed-commit:
Generated when a NETCONF server detects that a confirmed-commit event has occurred. Indicates the event and the current state of the confirmed-commit procedure in progress.

2.2. Definitions

<CODE BEGINS> file="ietf-netconf-notifications@2011-12-09.yang"

```
module ietf-netconf-notifications {

   namespace
     "urn:ietf:params:xml:ns:yang:ietf-netconf-notifications";

   prefix ncn;

   import ietf-inet-types { prefix inet; }
   import ietf-netconf { prefix nc; }

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

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

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

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

      Editor:   Andy Bierman
                <mailto:andy@netconfcentral.org>";

   description
     "This module defines a YANG data model for use with the
      NETCONF protocol that allows the NETCONF client to
      receive common NETCONF base event notifications.

      Copyright (c) 2012 IETF Trust and the persons identified as
      the document authors.  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 6470; see
        the RFC itself for full legal notices.";

      revision "2012-02-06" {
        description
          "Initial version.";
        reference
          "RFC 6470: NETCONF Base Notifications";
      }

    grouping common-session-parms {
      description
        "Common session parameters to identify a
         management session.";

      leaf username {
        type string;
        mandatory true;
        description
          "Name of the user for the session.";
      }

      leaf session-id {
        type nc:session-id-or-zero-type;
        mandatory true;
        description
          "Identifier of the session.
           A NETCONF session MUST be identified by a non-zero value.
           A non-NETCONF session MAY be identified by the value zero.";
      }

      leaf source-host {
        type inet:ip-address;
        description
          "Address of the remote host for the session.";
      }
    }
```

```
 grouping changed-by-parms {
  description
    "Common parameters to identify the source
     of a change event, such as a configuration
     or capability change.";

  container changed-by {
    description
      "Indicates the source of the change.
       If caused by internal action, then the
       empty leaf 'server' will be present.
       If caused by a management session, then
       the name, remote host address, and session ID
       of the session that made the change will be reported.";
    choice server-or-user {
      mandatory true;
      leaf server {
        type empty;
        description
          "If present, the change was caused
           by the server.";
      }

      case by-user {
        uses common-session-parms;
      }
    } // choice server-or-user
  } // container changed-by-parms
}


notification netconf-config-change {
  description
    "Generated when the NETCONF server detects that the
     <running> or <startup> configuration datastore
     has been changed by a management session.
     The notification summarizes the edits that
     have been detected.

     The server MAY choose to also generate this
     notification while loading a datastore during the
     boot process for the device.";

  uses changed-by-parms;
```

```
      leaf datastore {
        type enumeration {
          enum running {
            description "The <running> datastore has changed.";
          }
          enum startup {
            description "The <startup> datastore has changed";
          }
        }
        default "running";
        description
          "Indicates which configuration datastore has changed.";
      }

      list edit {
        description
          "An edit record SHOULD be present for each distinct
           edit operation that the server has detected on
           the target datastore.  This list MAY be omitted
           if the detailed edit operations are not known.
           The server MAY report entries in this list for
           changes not made by a NETCONF session (e.g., CLI).";

        leaf target {
          type instance-identifier;
          description
            "Topmost node associated with the configuration change.
             A server SHOULD set this object to the node within
             the datastore that is being altered.  A server MAY
             set this object to one of the ancestors of the actual
             node that was changed, or omit this object, if the
             exact node is not known.";
        }

        leaf operation {
          type nc:edit-operation-type;
          description
            "Type of edit operation performed.
             A server MUST set this object to the NETCONF edit
             operation performed on the target datastore.";
        }
      } // list edit
    } // notification netconf-config-change
```

```
    notification netconf-capability-change {
      description
        "Generated when the NETCONF server detects that
         the server capabilities have changed.
         Indicates which capabilities have been added, deleted,
         and/or modified.  The manner in which a server
         capability is changed is outside the scope of this
         document.";

      uses changed-by-parms;

      leaf-list added-capability {
        type inet:uri;
        description
          "List of capabilities that have just been added.";
      }

      leaf-list deleted-capability {
        type inet:uri;
        description
          "List of capabilities that have just been deleted.";
      }

      leaf-list modified-capability {
        type inet:uri;
        description
          "List of capabilities that have just been modified.
           A capability is considered to be modified if the
           base URI for the capability has not changed, but
           one or more of the parameters encoded at the end of
           the capability URI have changed.
           The new modified value of the complete URI is returned.";
      }
    } // notification netconf-capability-change


    notification netconf-session-start {
      description
        "Generated when a NETCONF server detects that a
         NETCONF session has started.  A server MAY generate
         this event for non-NETCONF management sessions.
         Indicates the identity of the user that started
         the session.";
      uses common-session-parms;
    } // notification netconf-session-start
```

```
     notification netconf-session-end {
       description
         "Generated when a NETCONF server detects that a
          NETCONF session has terminated.
          A server MAY optionally generate this event for
          non-NETCONF management sessions.  Indicates the
          identity of the user that owned the session,
          and why the session was terminated.";

       uses common-session-parms;

       leaf killed-by {
         when "../termination-reason = 'killed'";
         type nc:session-id-type;
         description
           "The ID of the session that directly caused this session
            to be abnormally terminated.  If this session was abnormally
            terminated by a non-NETCONF session unknown to the server,
            then this leaf will not be present.";
       }

       leaf termination-reason {
         type enumeration {
           enum "closed" {
             description
               "The session was terminated by the client in normal
                fashion, e.g., by the NETCONF <close-session>
                protocol operation.";
           }
           enum "killed" {
             description
               "The session was terminated in abnormal
                fashion, e.g., by the NETCONF <kill-session>
                protocol operation.";
           }
           enum "dropped" {
             description
               "The session was terminated because the transport layer
                connection was unexpectedly closed.";
           }
           enum "timeout" {
             description
               "The session was terminated because of inactivity,
                e.g., waiting for the <hello> message or <rpc>
                messages.";
           }
```

```
          enum "bad-hello" {
            description
              "The client's <hello> message was invalid.";
          }
          enum "other" {
            description
              "The session was terminated for some other reason.";
          }
        }
        mandatory true;
        description
          "Reason the session was terminated.";
      }
    } // notification netconf-session-end


    notification netconf-confirmed-commit {
      description
        "Generated when a NETCONF server detects that a
         confirmed-commit event has occurred.  Indicates the event
         and the current state of the confirmed-commit procedure
         in progress.";
      reference
        "RFC 6241, Section 8.4";

      uses common-session-parms {
        when "../confirm-event != 'timeout'";
      }

      leaf confirm-event {
        type enumeration {
          enum "start" {
            description
              "The confirmed-commit procedure has started.";
          }
          enum "cancel" {
            description
              "The confirmed-commit procedure has been canceled,
               e.g., due to the session being terminated, or an
               explicit <cancel-commit> operation.";
          }
          enum "timeout" {
            description
              "The confirmed-commit procedure has been canceled
               due to the confirm-timeout interval expiring.
               The common session parameters will not be present
               in this sub-mode.";
          }
```

```
            enum "extend" {
              description
                "The confirmed-commit timeout has been extended,
                 e.g., by a new <confirmed-commit> operation.";
            }
            enum "complete" {
              description
                "The confirmed-commit procedure has been completed.";
            }
          }
          mandatory true;
          description
            "Indicates the event that caused the notification.";
        }

        leaf timeout {
          when
            "../confirm-event = 'start' or ../confirm-event = 'extend'";
          type uint32;
          units "seconds";
          description
            "The configured timeout value if the event type
             is 'start' or 'extend'.  This value represents
             the approximate number of seconds from the event
             time when the 'timeout' event might occur.";
        }
      } // notification netconf-confirmed-commit

    }

    <CODE ENDS>
```

3. IANA Considerations

This document registers one XML namespace URN in the 'IETF XML registry', following the format defined in [RFC3688].

URI: urn:ietf:params:xml:ns:yang:ietf-netconf-notifications

Registrant Contact: The IESG.

XML: N/A; the requested URI is an XML namespace.

This document registers one module name in the 'YANG Module Names' registry, defined in [RFC6020].

```
name: ietf-netconf-notifications
prefix: ncn
namespace: urn:ietf:params:xml:ns:yang:ietf-netconf-notifications
RFC: 6470
```

4. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH, defined in [RFC6242].

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/netconf-config-change:
: Event type itself indicates that the system configuration has changed. This event could alert an attacker that specific configuration data nodes have been altered.

/netconf-config-change/changed-by:
: Indicates whether the server or a specific user management session made the configuration change. Identifies the user name, session-id, and source host address associated with the configuration change, if any.

/netconf-config-change/datastore:
: Indicates which datastore has been changed. This data can be used to determine if the non-volatile startup configuration data has been changed.

/netconf-config-change/edit:
: Identifies the specific edit operations and specific datastore subtree(s) that have changed. This data could be used to determine if specific server vulnerabilities may now be present.

/netconf-capability-change:
Event type itself indicates that the system capabilities have changed, and may now be vulnerable to unspecified attacks. An attacker will likely need to understand the content represented by specific capability URI strings. For example, knowing that a packet capture monitoring capability has been added to the system might help an attacker identify the device for possible unauthorized eavesdropping.
/netconf-capability-change/changed-by:
Indicates whether the server or a specific user management session made the capability change. Identifies the user name, session-id, and source host address associated with the capability change, if any.
/netconf-capability-change/added-capability:
Indicates the specific capability URIs that have been added. This data could be used to determine if specific server vulnerabilities may now be present.
/netconf-capability-change/deleted-capability:
Indicates the specific capability URIs that have been deleted. This data could be used to determine if specific server vulnerabilities may now be present.
/netconf-capability-change/modified-capability:
Indicates the specific capability URIs that have been modified. This data could be used to determine if specific server vulnerabilities may now be present.

/netconf-session-start:
Event type itself indicates that a NETCONF or other management session may start altering the device configuration and/or state. It may be possible for an attacker to alter the configuration by somehow taking advantage of another session concurrently editing an unlocked datastore.
/netconf-session-start/username:
Indicates the user name associated with the session.
/netconf-session-start/source-host:
Indicates the source host address associated with the session.

/netconf-session-end:
Event type itself indicates that a NETCONF or other management session may be finished altering the device configuration. This event could alert an attacker that a datastore may have been altered.
/netconf-session-end/username:
Indicates the user name associated with the session.
/netconf-session-end/source-host:
Indicates the source host address associated with the session.

/netconf-confirmed-commit:
Event type itself indicates that the <running> datastore may have changed. This event could alert an attacker that the device behavior has changed.
/netconf-confirmed-commit/username:
Indicates the user name associated with the session.
/netconf-confirmed-commit/source-host:
Indicates the source host address associated with the session.
/netconf-confirmed-commit/confirm-event:
Indicates the specific confirmed-commit state change that occurred. A value of 'complete' probably indicates that the <running> datastore has changed.
/netconf-confirmed-commit/timeout:
Indicates the number of seconds in the future when the <running> datastore may change, due to the server reverting to an older configuration.

5. Acknowledgements

Thanks to Martin Bjorklund, Juergen Schoenwaelder, Kent Watsen, and many other members of the NETCONF WG for providing important input to this document.

6. Normative References

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

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

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

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

[RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6021, October 2010.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

Author’s Address

Andy Bierman
Brocade

EMail: andy@netconfcentral.org