Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 15, 2014

A. Bierman
YumaWorks
M. Bjorklund
Tail-f Systems

May 14, 2014

A YANG Data Model for System Management
draft-ietf-netmod-system-mgmt-16

Abstract

This document defines a YANG data model for the configuration and identification of some common system properties within a device containing a NETCONF server. This includes data node definitions for system identification, time-of-day management, user management, DNS resolver configuration, and some protocol operations for system management.

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 November 15, 2014.

Table of Contents

1. Introduction

This document defines a YANG [RFC6020] data model for the configuration and identification of some common properties within a device containing a NETCONF server.

Devices that are managed by NETCONF and perhaps other mechanisms have common properties that need to be configured and monitored in a standard way.

The "ietf-system" YANG module defined in this document provides the following features:

- system identification configuration and monitoring
- system time-of-day configuration and monitoring
- user authentication configuration
- local users configuration
- DNS resolver configuration
- system control operations (shutdown, restart, setting time)

1.1. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, [RFC2119].

The following terms are defined in [RFC6241] and are not redefined here:

- client
- configuration data
- server
- state data

1.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.
- Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Objectives

2.1. System Identification

There are many common properties used to identify devices, operating systems, software versions, etc. that need to be supported in the system data module. These objects are defined as operational state data and the information returned by the server is intended to be specific to the device vendor.

Some user-configurable administrative strings are also provided, such as the system location and description.

2.2. System Time Management

The management of the date and time used by the system need to be supported. Use of one or more NTP servers to automatically set the system date and time need to be possible. Utilization of the Timezone database [RFC6557] also need to be supported. It should be possible to configure the system to use NTP.

2.3. User Authentication

The authentication mechanism needs to support password authentication over RADIUS, to support deployment scenarios with centralized authentication servers. Additionally, local users need to be supported, for scenarios when no centralized authentication server exists, or for situations where the centralized authentication server cannot be reached from the device.

Since the mandatory transport protocol for NETCONF is SSH [RFC6242] the authentication model needs to support SSH's "publickey" and "password" authentication methods [RFC4252].

The model for authentication configuration should be flexible enough to support authentication methods defined by other standard documents or by vendors. It should be possible to configure the system authentication properties.

2.4. DNS Resolver

The configuration of the DNS resolver within the system containing the NETCONF server is required in order to control how domain names are resolved.

2.5. System Control

A few operations are needed to support common tasks such as restarting the device or setting the system date and time.

3. System Data Model

3.1. System Identification

The data model for system identification has the following structure:

```
+--rw system
|  +--rw contact?          string
|  +--rw hostname?         inet:domain-name
|  +--rw location?         string
+--ro system-state
   +--ro platform
      +--ro os-name?       string
      +--ro os-release?    string
      +--ro os-version?    string
      +--ro machine?       string
```

3.2. System Time Management

The data model for system time management has the following structure:

```
+--rw system
|  +--rw clock
|  |  +--rw (timezone)?
|  |     +--:(timezone-name)
|  |     |  +--rw timezone-name?     timezone-name
|  |     +--:(timezone-utc-offset)
|  |        +--rw timezone-utc-offset?   int16
|  +--rw ntp!
|     +--rw enabled?   boolean
|     +--rw server* [name]
|        +--rw name                string
|        +--rw (transport)
|        |  +--:(udp)
|        |     +--rw udp
|        |        +--rw address    inet:host
|        |        +--rw port?      inet:port-number
|        +--rw association-type?   enumeration
|        +--rw iburst?             boolean
|        +--rw prefer?             boolean
+--ro system-state
   +--ro clock
      +--ro current-datetime?      yang:date-and-time
      +--ro boot-datetime?         yang:date-and-time
```

New "case" statements can be added over time or augmented to the "transport" choice to support other transport protocols.

3.3. DNS Resolver Model

The data model for configuration of the DNS resolver has the following structure:

```
   +--rw system
      +--rw dns-resolver
         +--rw search*    inet:domain-name
         +--rw server* [name]
         |  +--rw name    string
         |  +--rw (transport)
         |     +--:(udp-and-tcp)
         |        +--udp-and-tcp
         |           +--rw address    inet:ip-address
         |           +--rw port?      inet:port-number
         +--rw options
            +--rw timeout?    uint8
            +--rw attempts?   uint8
```

New "case" statements can be added over time or augmented to the "transport" choice to support other transport protocols.

3.4. RADIUS Client Model

The data model for configuration of the RADIUS client has the following structure:

```
   +--rw system
      +--rw radius
         +--rw server* [name]
         |  +--rw name                    string
         |  +--rw (transport)
         |  |  +--:(udp)
         |  |     +--rw udp
         |  |        +--rw address                inet:host
         |  |        +--rw authentication-port?   inet:port-number
         |  |        +--rw shared-secret          string
         |  +--rw authentication-type?   identityref
         +--rw options
            +--rw timeout?    uint8
            +--rw attempts?   uint8
```

New "case" statements can be added over time or augmented to the "transport" choice to support other transport protocols.

3.5. User Authentication Model

This document defines three authentication methods for use with NETCONF:

- publickey for local users over SSH
- password for local users over any secure transport
- password for RADIUS users over any secure transport

Additional methods can be defined by other standard documents or by vendors.

This document defines two optional YANG features, "local-users" and "radius-authentication", which the server advertises to indicate support for configuring local users on the device, and support for using RADIUS for authentication, respectively.

The authentication parameters defined in this document are primarily used to configure authentication of NETCONF users, but MAY also be used by other interfaces, e.g., a Command Line Interface or a Web-based User Interface.

The data model for user authentication has the following structure:

```
   +--rw system
      +--rw authentication
         +--rw user-authentication-order*   identityref
         +--rw user* [name]
            +--rw name        string
            +--rw password?   ianach:crypt-hash
            +--rw authorized-key* [name]
               +--rw name         string
               +--rw algorithm    string
               +--rw key-data     binary
```

3.5.1. SSH Public Key Authentication

If the NETCONF server advertises the "local-users" feature, configuration of local users and their SSH public keys is supported in the /system/authentication/user list.

Public key authentication is requested by the SSH client. If the "local-users" feature is supported, then when a NETCONF client starts an SSH session towards the server using the "publickey" authentication "method name" [RFC4252], the SSH server looks up the user name given in the SSH authentication request in the /system/

authentication/user list, and verifies the key as described in [RFC4253].

3.5.2. Local User Password Authentication

If the NETCONF server advertises the "local-users" feature, configuration of local users and their passwords is supported in the /system/authentication/user list.

For NETCONF transport protocols that support password authentication, the leaf-list "user-authentication-order" is used to control if local user password authentication should be used.

In SSH, password authentication is requested by the client. Other NETCONF transport protocols MAY also support password authentication.

When local user password authentication is requested, the NETCONF transport looks up the user name provided by the client in the /system/authentication/user list, and verifies the password.

3.5.3. RADIUS Password Authentication

If the NETCONF server advertises the "radius-authentication" feature, the device supports user authentication using RADIUS.

For NETCONF transport protocols that support password authentication, the leaf-list "user-authentication-order" is used to control if RADIUS password authentication should be used.

In SSH, password authentication is requested by the client. Other NETCONF transport protocols MAY also support password authentication.

3.6. System Control

The following operations are defined:

```
set-current-datetime
system-restart
system-shutdown
```

Two protocol operations are included to restart or shutdown the system. The 'system-restart' operation can be used to restart the entire system (not just the NETCONF server). The 'system-shutdown' operation can be used to power off the entire system.

4. Relationship to the SNMPv2-MIB

If a device implements the SNMPv2-MIB [RFC3418], there are two objects that MAY be mapped by the implementation. See the YANG module definition in Section 6 for details. The following table lists the YANG data nodes with corresponding objects in the SNMPv2-MIB.

YANG data node	SNMPv2-MIB object
contact	sysContact
location	sysLocation

YANG interface configuration data nodes and related SNMPv2-MIB objects

5. IANA Crypt Hash YANG module

This YANG module references [RFC1321], [IEEE-1003.1-2008], and [FIPS.180-3.2008].

RFC Ed.: update the date below with the date of RFC publication and remove this note.

<CODE BEGINS> file "iana-crypt-hash@2014-04-04.yang"

```
module iana-crypt-hash {
  namespace "urn:ietf:params:xml:ns:yang:iana-crypt-hash";
  prefix ianach;

  organization "IANA";
  contact
    "        Internet Assigned Numbers Authority

     Postal: ICANN
             4676 Admiralty Way, Suite 330
             Marina del Rey, CA 90292

     Tel:    +1 310 823 9358
     E-Mail: iana&iana.org";
  description
    "This YANG module defines a typedef for storing passwords
     using a hash function, and features to indicate which hash
     functions are supported by an implementation.

     The latest revision of this YANG module can be obtained from
     the IANA web site.

     Requests for new values should be made to IANA via
     email (iana&iana.org).

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  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).

     The initial version of this YANG module is part of RFC XXXX;
     see the RFC itself for full legal notices.";
  // RFC Ed.: replace XXXX with actual RFC number and remove this
```

```
    // note.

    // RFC Ed.: update the date below with the date of RFC publication
    // and remove this note.
    revision 2014-04-04 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: A YANG Data Model for System Management";
    }

    typedef crypt-hash {
      type string {
        pattern
          '$0$.*'
        + '|$1$[a-zA-Z0-9./]{1,8}$[a-zA-Z0-9./]{22}'
        + '|$5$(rounds=\d+$)?[a-zA-Z0-9./]{1,16}$[a-zA-Z0-9./]{43}'
        + '|$6$(rounds=\d+$)?[a-zA-Z0-9./]{1,16}$[a-zA-Z0-9./]{86}';
      }
      description
        "The crypt-hash type is used to store passwords using
         a hash function.  The algorithms for applying the hash
         function and encoding the result are implemented in
         various UNIX systems as the function crypt(3).

         A value of this type matches one of the forms:

           $0$<clear text password>
           $<id>$<salt>$<password hash>
           $<id>$<parameter>$<salt>$<password hash>

         The '$0$' prefix signals that the value is clear text.  When
         such a value is received by the server, a hash value is
         calculated, and the string '$<id>$<salt>$' or
         $<id>$<parameter>$<salt>$ is prepended to the result.  This
         value is stored in the configuration data store.

         If a value starting with '$<id>$', where <id> is not '0', is
         received, the server knows that the value already represents a
         hashed value, and stores it as is in the data store.

         When a server needs to verify a password given by a user, it
         finds the stored password hash string for that user, extracts
         the salt, and calculates the hash with the salt and given
         password as input.  If the calculated hash value is the same
         as the stored value, the password given by the client is
         accepted.
```

```
          This type defines the following hash functions:

            id | hash function | feature
            ---+---------------+-------------------
             1 | MD5           | crypt-hash-md5
             5 | SHA-256       | crypt-hash-sha-256
             6 | SHA-512       | crypt-hash-sha-512

          The server indicates support for the different hash functions
          by advertising the corresponding feature.";
        reference
          "IEEE Std 1003.1-2008 - crypt() function
           RFC 1321: The MD5 Message-Digest Algorithm
           FIPS.180-3.2008: Secure Hash Standard";
      }

      feature crypt-hash-md5 {
        description
          "Indicates that the device supports the MD5
           hash function in 'crypt-hash' values";
        reference "RFC 1321: The MD5 Message-Digest Algorithm";
      }

      feature crypt-hash-sha-256 {
        description
          "Indicates that the device supports the SHA-256
           hash function in 'crypt-hash' values";
        reference "FIPS.180-3.2008: Secure Hash Standard";
      }

      feature crypt-hash-sha-512 {
        description
          "Indicates that the device supports the SHA-512
           hash function in 'crypt-hash' values";
        reference "FIPS.180-3.2008: Secure Hash Standard";
      }

    }

    <CODE ENDS>
```

6. System YANG module

This YANG module imports YANG extensions from [RFC6536], and imports YANG types from [RFC6991]. It also references [RFC1035], [RFC2865], [RFC3418], [RFC5607], [RFC5966], [RFC6557].

RFC Ed.: update the date below with the date of RFC publication and remove this note.

<CODE BEGINS> file "ietf-system@2014-05-14.yang"

```
module ietf-system {
  namespace "urn:ietf:params:xml:ns:yang:ietf-system";
  prefix "sys";

  import ietf-yang-types {
    prefix yang;
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-netconf-acm {
    prefix nacm;
  }

  import iana-crypt-hash {
    prefix ianach;
  }

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

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

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Andy Bierman
               <mailto:andy@yumaworks.com>

     Editor:   Martin Bjorklund
```

```
                <mailto:mbj@tail-f.com>";

    description
      "This module contains a collection of YANG definitions for the
       configuration and identification of some common system
       properties within a device containing a NETCONF server.  This
       includes data node definitions for system identification,
       time-of-day management, user management, DNS resolver
       configuration, and some protocol operations for system
       management.

       Copyright (c) 2014 IETF Trust and the persons identified as
       authors of the code.  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 XXXX; see
       the RFC itself for full legal notices.";

    // RFC Ed.: replace XXXX with actual RFC number and remove this
    // note.

    // RFC Ed.: remove this note
    // Note: extracted from draft-ietf-netmod-system-mgmt-07.txt

    // RFC Ed.: update the date below with the date of RFC publication
    // and remove this note.
    revision "2014-05-14" {
      description
        "Initial revision.";
      reference
        "RFC XXXX: A YANG Data Model for System Management";
    }

    /*
     * Typedefs
     */

    typedef timezone-name {
      type string;
      description
        "A timezone name as used by the Time Zone Database, sometimes
         referred to as the 'Olson Database'.
```

```
          The exact set of valid values is an implementation-specific
          matter.  Client discovery of the exact set of time zone names
          for a particular server is out of scope.";
        reference
          "RFC 6557: Procedures for Maintaining the Time Zone Database";
       }


     /*
      * Features
      */

     feature radius {
       description
         "Indicates that the device can be configured as a RADIUS
          client.";
       reference
         "RFC 2865: Remote Authentication Dial In User Service "
       + "(RADIUS)";
     }

     feature authentication {
       description
         "Indicates that the device supports configuration
          for user authentication.";
     }

     feature local-users {
       if-feature authentication;
       description
         "Indicates that the device supports configuration of
          local user authentication.";
     }

     feature radius-authentication {
       if-feature radius;
       if-feature authentication;
       description
         "Indicates that the device supports configuration of user
          authentication over RADIUS.";
       reference
         "RFC 2865: Remote Authentication Dial In User Service (RADIUS)
          RFC 5607: Remote Authentication Dial-In User Service (RADIUS)
                    Authorization for Network Access Server (NAS)
                    Management";
     }

     feature ntp {
```

```
      description
        "Indicates that the device can be configured
         to use one or more NTP servers to set the
         system date and time.";
    }

    feature ntp-udp-port {
      if-feature ntp;
      description
        "Indicates that the device supports the configuration of
         the UDP port for NTP servers.

         This is a 'feature' since many implementations do not support
         any other port than the default port.";
    }

    feature timezone-name {
      description
        "Indicates that the local timezone on the device
         can be configured to use the TZ database
         to set the timezone and manage daylight savings time.";
      reference
        "RFC 6557: Procedures for Maintaining the Time Zone Database";
    }

    feature dns-udp-tcp-port {
      description
        "Indicates that the device supports the configuration of
         the UDP and TCP port for DNS servers.

         This is a 'feature' since many implementations do not support
         any other port than the default port.";
    }

    /*
     * Identities
     */

    identity authentication-method {
      description
        "Base identity for user authentication methods.";
    }

    identity radius {
      base authentication-method;
      description
        "Indicates user authentication using RADIUS.";
      reference
```

```
         "RFC 2865: Remote Authentication Dial In User Service (RADIUS)
          RFC 5607: Remote Authentication Dial-In User Service (RADIUS)
                    Authorization for Network Access Server (NAS)
                    Management";
     }

     identity local-users {
       base authentication-method;
       description
         "Indicates password-based authentication of locally
          configured users.";
     }

     identity radius-authentication-type {
       description
         "Base identity for RADIUS authentication types.";
     }

     identity radius-pap {
       base radius-authentication-type;
       description
         "The device requests PAP authentication from the RADIUS
          server.";
       reference
         "RFC 2865: Remote Authentication Dial In User Service";
     }

     identity radius-chap {
       base radius-authentication-type;
       description
         "The device requests CHAP authentication from the RADIUS
          server.";
       reference
         "RFC 2865: Remote Authentication Dial In User Service";
     }

     /*
      * Configuration data nodes
      */

     container system {
       description
         "System group configuration.";

       leaf contact {
         type string;
         description
           "The administrator contact information for the system.
```

```
          A server implementation MAY map this leaf to the sysContact
          MIB object.  Such an implementation needs to use some
          mechanism to handle the differences in size and characters
          allowed between this leaf and sysContact.  The definition of
          such a mechanism is outside the scope of this document.";
        reference
          "RFC 3418: Management Information Base (MIB) for the
                     Simple Network Management Protocol (SNMP)
                     SNMPv2-MIB.sysContact";
      }
      leaf hostname {
        type inet:domain-name;
       description
         "The name of the host.  This name can be a single domain
          label, or the fully qualified domain name of the host.";
      }
      leaf location {
        type string;
        description
          "The system location.

          A server implementation MAY map this leaf to the sysLocation
          MIB object.  Such an implementation needs to use some
          mechanism to handle the differences in size and characters
          allowed between this leaf and sysLocation.  The definition
          of such a mechanism is outside the scope of this document.";
        reference
          "RFC 3418: Management Information Base (MIB) for the
                     Simple Network Management Protocol (SNMP)
                     SNMPv2-MIB.sysLocation";
      }

      container clock {
        description
          "Configuration of the system date and time properties.";

        choice timezone {
          description
            "The system timezone information.";

          case timezone-name {
            if-feature timezone-name;
            leaf timezone-name {
              type timezone-name;
              description
                "The TZ database name to use for the system, such
                 as 'Europe/Stockholm'.";
            }
```

```
          }
          case timezone-utc-offset {
            leaf timezone-utc-offset {
              type int16 {
                range "-1500 .. 1500";
              }
              units "minutes";
              description
                "The number of minutes to add to UTC time to
                 identify the timezone for this system.  For example,
                 'UTC - 8:00 hours' would be represented as '-480'.
                 Note that automatic daylight savings time adjustment
                 is not provided, if this object is used.";
            }
          }
        }
      }

      container ntp {
        if-feature ntp;
        presence
          "Enables the NTP client unless the 'enabled' leaf
           (which defaults to 'true') is set to 'false'";
        description
          "Configuration of the NTP client.";

        leaf enabled {
          type boolean;
          default true;
          description
            "Indicates that the system should attempt
             to synchronize the system clock with an
             NTP server from the 'ntp/server' list.";
        }
        list server {
          key name;
          description
            "List of NTP servers to use for
             system clock synchronization.  If '/system/ntp/enabled'
             is 'true', then the system will attempt to
             contact and utilize the specified NTP servers.";

          leaf name {
            type string;
            description
              "An arbitrary name for the NTP server.";
          }
          choice transport {
```

```
            mandatory true;
            description
              "The transport protocol specific parameters for this
               server.";

            case udp {
              container udp {
                description
                  "Contains UDP specific configuration parameters
                   for NTP.";
                leaf address {
                  type inet:host;
                  mandatory true;
                  description
                    "The address of the NTP server.";
                }
                leaf port {
                  if-feature ntp-udp-port;
                  type inet:port-number;
                  default 123;
                  description
                    "The port number of the NTP server.";
                }
              }
            }
          }
          leaf association-type {
            type enumeration {
              enum server {
                description
                  "Use client association mode.  This device
                   will not provide synchronization to the
                   configured NTP server.";
              }
              enum peer {
                description
                  "Use symmetric active association mode.
                   This device may provide synchronization
                   to the configured NTP server.";
              }
              enum pool {
                description
                  "Use client association mode with one or
                   more of the NTP servers found by DNS
                   resolution of the domain name given by
                   the 'address' leaf.  This device will not
                   provide synchronization to the servers.";
              }
```

```
            }
            default server;
            description
              "The desired association type for this NTP server.";
          }
          leaf iburst {
            type boolean;
            default false;
            description
              "Indicates whether this server should enable burst
               synchronization or not.";
          }
          leaf prefer {
            type boolean;
            default false;
            description
              "Indicates whether this server should be preferred
               or not.";
          }
        }
      }

      container dns-resolver {
        description
          "Configuration of the DNS resolver.";

        leaf-list search {
          type inet:domain-name;
          ordered-by user;
          description
            "An ordered list of domains to search when resolving
             a host name.";
        }
        list server {
          key name;
          ordered-by user;
          description
            "List of the DNS servers that the resolver should query.

             When the resolver is invoked by a calling application, it
             sends the query to the first name server in this list.  If
             no response has been received within 'timeout' seconds,
             the resolver continues with the next server in the list.
             If no response is received from any server, the resolver
             continues with the first server again.  When the resolver
             has traversed the list 'attempts' times without receiving
             any response, it gives up and returns an error to the
             calling application.
```

```
           Implementations MAY limit the number of entries in this
           list.";

         leaf name {
           type string;
           description
             "An arbitrary name for the DNS server.";
         }
         choice transport {
           mandatory true;
           description
             "The transport protocol specific parameters for this
              server.";

           case udp-and-tcp {
             container udp-and-tcp {
               description
                 "Contains UDP and TCP specific configuration
                  parameters for DNS.";
               reference
                 "RFC 1035: Domain Implementation and Specification
                  RFC 5966: DNS over TCP";

               leaf address {
                 type inet:ip-address;
                 mandatory true;
                 description
                   "The address of the DNS server.";
               }
               leaf port {
                 if-feature dns-udp-tcp-port;
                 type inet:port-number;
                 default 53;
                 description
                   "The UDP and TCP port number of the DNS server.";
               }
             }
           }
         }
       }
       container options {
         description
           "Resolver options.  The set of available options has been
            limited to those that are generally available across
            different resolver implementations, and generally
            useful.";
         leaf timeout {
           type uint8 {
```

```
            range "1..max";
          }
          units "seconds";
          default "5";
          description
            "The amount of time the resolver will wait for a
             response from each remote name server before
             retrying the query via a different name server.";
        }
        leaf attempts {
          type uint8 {
            range "1..max";
          }
          default "2";
          description
            "The number of times the resolver will send a query to
             all its name servers before giving up and returning an
             error to the calling application.";
        }
      }
    }

    container radius {
      if-feature radius;

      description
        "Configuration of the RADIUS client.";

      list server {
        key name;
        ordered-by user;
        description
          "List of RADIUS servers used by the device.

           When the RADIUS client is invoked by a calling
           application, it sends the query to the first server in
           this list.  If no response has been received within
           'timeout' seconds, the client continues with the next
           server in the list.  If no response is received from any
           server, the client continues with the first server again.
           When the client has traversed the list 'attempts' times
           without receiving any response, it gives up and returns an
           error to the calling application.";

        leaf name {
          type string;
          description
            "An arbitrary name for the RADIUS server.";
```

```
        }
        choice transport {
          mandatory true;
          description
            "The transport protocol specific parameters for this
             server.";

          case udp {
            container udp {
              description
                "Contains UDP specific configuration parameters
                 for RADIUS.";
              leaf address {
                type inet:host;
                mandatory true;
                description
                  "The address of the RADIUS server.";
              }
              leaf authentication-port {
                type inet:port-number;
                default "1812";
                description
                  "The port number of the RADIUS server.";
              }
              leaf shared-secret {
                type string;
                mandatory true;
                nacm:default-deny-all;
                description
                  "The shared secret which is known to both the
                   RADIUS client and server.";
                reference
                  "RFC 2865: Remote Authentication Dial In User
                   Service";
              }
            }
          }
        }
        leaf authentication-type {
          type identityref {
            base radius-authentication-type;
          }
          default radius-pap;
          description
            "The authentication type requested from the RADIUS
             server.";
        }
      }
```

```
        container options {
          description
            "RADIUS client options.";

          leaf timeout {
            type uint8 {
              range "1..max";
            }
            units "seconds";
            default "5";
            description
              "The number of seconds the device will wait for a
               response from each RADIUS server before trying with a
               different server.";
          }
          leaf attempts {
            type uint8 {
              range "1..max";
            }
            default "2";
            description
              "The number of times the device will send a query to
               all its RADIUS servers before giving up.";
          }
        }
      }

      container authentication {
        nacm:default-deny-write;
        if-feature authentication;

         description
           "The authentication configuration subtree.";

         leaf-list user-authentication-order {
           type identityref {
             base authentication-method;
           }
           must '(. != "sys:radius" or ../../radius/server)' {
             error-message
               "When 'radius' is used, a RADIUS server"
             + " must be configured.";
             description
               "When 'radius' is used as an authentication method,
                a RADIUS server must be configured.";
           }
           ordered-by user;
```

```
      description
        "When the device authenticates a user with a password,
         it tries the authentication methods in this leaf-list in
         order.  If authentication with one method fails, the next
         method is used.  If no method succeeds, the user is
         denied access.

         An empty user-authentication-order leaf-list still allows
         authentication of users using mechanisms that do not
         involve a password.

         If the 'radius-authentication' feature is advertised by
         the NETCONF server, the 'radius' identity can be added to
         this list.

         If the 'local-users' feature is advertised by the
         NETCONF server, the 'local-users' identity can be
         added to this list.";
    }

    list user {
      if-feature local-users;
      key name;
      description
        "The list of local users configured on this device.";

      leaf name {
        type string;
        description
         "The user name string identifying this entry.";
      }
      leaf password {
        type ianach:crypt-hash;
        description
          "The password for this entry.";
      }
      list authorized-key {
        key name;
        description
          "A list of public SSH keys for this user.  These keys
           are allowed for SSH authentication, as described in
           RFC 4253.";
        reference
          "RFC 4253: The Secure Shell (SSH) Transport Layer
                     Protocol";

        leaf name {
          type string;
```

```
              description
                "An arbitrary name for the SSH key.";
            }
            leaf algorithm {
              type string;
              mandatory true;
              description
                "The public key algorithm name for this SSH key.

                 Valid values are the values in the IANA Secure Shell
                 (SSH) Protocol Parameters registry, Public Key
                 Algorithm Names";
              reference
                "IANA Secure Shell (SSH) Protocol Parameters registry,
                 Public Key Algorithm Names";
            }
            leaf key-data {
              type binary;
              mandatory true;
              description
                "The binary public key data for this SSH key, as
                 specified by RFC 4253, Section 6.6, i.e.,:

                   string    certificate or public key format
                             identifier
                   byte[n]   key/certificate data
                 ";
              reference
                "RFC 4253: The Secure Shell (SSH) Transport Layer
                           Protocol";
            }
          }
        }
      }
    }

    /*
     * Operational state data nodes
     */

    container system-state {
      config false;
      description
        "System group operational state.";

      container platform {
        description
          "Contains vendor-specific information for
```

```
           identifying the system platform and operating system.";
        reference
          "IEEE Std 1003.1-2008 - sys/utsname.h";

        leaf os-name {
          type string;
          description
            "The name of the operating system in use,
             for example 'Linux'";
          reference
            "IEEE Std 1003.1-2008 - utsname.sysname";
        }
        leaf os-release {
          type string;
          description
            "The current release level of the operating
             system in use.  This string MAY indicate
             the OS source code revision.";
          reference
            "IEEE Std 1003.1-2008 - utsname.release";
        }
        leaf os-version {
          type string;
          description
            "The current version level of the operating
             system in use.  This string MAY indicate
             the specific OS build date and target variant
             information.";
          reference
            "IEEE Std 1003.1-2008 - utsname.version";
        }
        leaf machine {
          type string;
          description
            "A vendor-specific identifier string representing
             the hardware in use.";
          reference
            "IEEE Std 1003.1-2008 - utsname.machine";
        }
      }

      container clock {
        description
          "Monitoring of the system
           date and time properties.";

        leaf current-datetime {
          type yang:date-and-time;
```

```
          description
            "The current system date and time.";
        }
        leaf boot-datetime {
          type yang:date-and-time;
          description
            "The system date and time when the system last restarted.";
        }
      }
    }

    rpc set-current-datetime {
      nacm:default-deny-all;
      description
        "Set the /system-state/clock/current-datetime leaf
         to the specified value.

         If the system is using NTP (i.e., /system/ntp/enabled
         is set to 'true'), then this operation will
         fail with error-tag 'operation-failed',
         and error-app-tag value of 'ntp-active'";
      input {
        leaf current-datetime {
          type yang:date-and-time;
          mandatory true;
          description
            "The current system date and time.";
        }
      }
    }

    rpc system-restart {
      nacm:default-deny-all;
      description
        "Request that the entire system be restarted immediately.
         A server SHOULD send an rpc reply to the client before
         restarting the system.";
    }

    rpc system-shutdown {
      nacm:default-deny-all;
      description
        "Request that the entire system be shut down immediately.
         A server SHOULD send an rpc reply to the client before
         shutting down the system.";
    }

  }
```

<CODE ENDS>

7. IANA Considerations

IANA is requested to create an IANA-maintained YANG Module called "iana-crypt-hash", based on the contents of Section 5, which will allow for new hash algorithms to be added to the type "crypt-hash". The registration procedure will be Expert Review, as defined by [RFC5226].

This document registers two URIs in the IETF XML registry [RFC3688]. Following the format in RFC 3688, the following registrations are requested to be made.

```
URI: urn:ietf:params:xml:ns:yang:iana-crypt-hash
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-system
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
```

This document registers two YANG modules in the YANG Module Names registry [RFC6020].

```
name:         iana-crypt-hash
namespace:    urn:ietf:params:xml:ns:yang:iana-crypt-hash
prefix:       ianach
reference:    RFC XXXX

name:         ietf-system
namespace:    urn:ietf:params:xml:ns:yang:ietf-system
prefix:       sys
reference:    RFC XXXX
```

8. Security Considerations

The YANG modules defined in this memo are 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 [RFC6242]. Authorization for access to specific portions of conceptual data and operations within this module is provided by the NETCONF access control model (NACM) [RFC6536].

There are a number of data nodes defined in the "ietf-system" YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations to these data nodes can have a negative effect on network operations. It is thus important to control write access (e.g., via edit-config) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

- /system/clock/timezone: This choice contains the objects used to control the timezone used by the device.

- /system/ntp: This container contains the objects used to control the Network Time Protocol servers used by the device.

- /system/dns-resolver: This container contains the objects used to control the Domain Name System servers used by the device.

- /system/radius: This container contains the objects used to control the Remote Authentication Dial-In User Service servers used by the device.

- /system/authentication/user-authentication-order: This leaf controls how user login attempts are authenticated by the device.

- /system/authentication/user: This list contains the local users enabled on the system.

Some of the readable data nodes in the "ietf-system" 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:

- /system/platform: This container has objects which may help identify the specific NETCONF server and/or operating system implementation used on the device.

- /system/authentication/user: This list has objects that may help identify the specific user names and password information in use on the device.

Some of the remote procedure call (RPC) operations in the "ietf-system" YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

- set-current-datetime: Changes the current date and time on the device.

- system-restart: Reboots the device.

- system-shutdown: Shuts down the device.

Since this document describes the use of RADIUS for purposes of authentication, it is vulnerable to all of the threats that are present in other RADIUS applications. For a discussion of such threats, see [RFC2865] and [RFC3162], and section 4 of [RFC3579].

This document provides configuration parameters for SSH's "publickey" and "password" authentication mechanisms. Section 9.4 of [RFC4251] and section 11 of [RFC4252] discuss security considerations for these mechanisms.

The "iana-crypt-hash" YANG module defines a type "crypt-hash" that can be used to store MD5 hashes. [RFC6151] discusses security considerations for MD5. The usage of MD5 is NOT RECOMMENDED.

9. Change Log

-- RFC Ed.: remove this section before publication.

9.1. 00-01

- added configuration-source identities
- added configuration-source leaf to ntp and dns (via grouping) to choose configuration source
- added association-type, iburst, prefer, and true leafs to the ntp-server list
- extended the ssh keys for a user to a list of keys. support all defined key algorithms, not just dsa and rsa
- clarified timezone-utc-offset description-stmt
- removed '/system/ntp/server/true' leaf from data model

9.2. 01-02

- added default-stmts to ntp-server/iburst and ntp-server/prefer leafs
- changed timezone-location leaf to use iana-timezone typedef instead of a string

9.3. 02-03

- removed configuration-source identities and leafs

9.4. 03-04

- removed ndots dns resolver option
- added radius-authentication-type identity, and identities for pap and chap, and a leaf to control which authentication type to use when communicating with the radius server
- made 0 an invalid value for timeouts and attempts

9.5. 04-05

- updated tree diagram explanation text

9.6. 05-06

- changed ntp/use-ntp to ntp/enabled
- changed ntp/ntp-server to ntp/server
- removed /system/platform/nodename leaf
- changed /system/name to /system/hostname
- simplified must expression in user-authentication-order
- added optional rounds to sha hash definition
- clarified the crypt-hash description
- clarified ntp descriptions
- clarified YANG module description to indicate that some system properties are supported, not the entire system
- clarified that system identification values are vendor specific, not the data node objects
- clarified sec. 2.2 and 2.3 to indicate that the server should also be capable of configuring these properties
- changed /system/dns/search from inet:host to inet:domain-name
- changed RFC6021 reference to 6021-bis
- changed /system/platform/nodename to /system/platform/hostname
- changed /system/radius/server/{leafs} to be within a choice and 'udp' case statement so other transport specific parameters can augment this list or they can be added by the WG to a future version of this module. {leafs} are authentication-port and shared-secret.
- updated YANG tree diagrams for objects added in -05 and -06

9.7. 06-07

- updated the Abstract and Introduction
- updated Tree diagram notation

- identify all external servers (dns, ntp, radius) by name instead of address, in order to make the data model extensible for additional transport protocol.

- updated the Security Considerations section with a reference to NACM.

9.8. 07-08

- renamed the DNS transport to 'udp-and-tcp' and added references.

- moved the operational state nodes into /system-state.

9.9. 08-09

- made "ntp" node a presence container

- added reference to RFC 6151

- updated reference from 6021-bis to RFC 6991

- cleaned up usage of config false in the YANG module

9.10. 09-10

- clarified relationship with SNMPv2-MIB

9.11. 11-12

- added typedef "timezone-name", and removed reference to draft-ietf-netmod-iana-timezones

9.12. 13-14

- moved the "crypt-hash" typedef to an IANA maintained module.

- updated security considerations to mention RADIUS threats.

9.13. 14-15

- updated security considerations to mention SSH authentication method threats.

10. References

10.1. Normative References

[FIPS.180-3.2008]
National Institute of Standards and Technology, "Secure Hash Standard", FIPS PUB 180-3, October 2008, <http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf>.

[IEEE-1003.1-2008]
Institute of Electrical and Electronics Engineers, "POSIX.1-2008", IEEE Standard 1003.1, March 2008.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

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

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC3162] Aboba, B., Zorn, G., and D. Mitton, "RADIUS and IPv6", RFC 3162, August 2001.

[RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC4251] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

[RFC4252] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Authentication Protocol", RFC 4252, January 2006.

[RFC4253] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5607] Nelson, D. and G. Weber, "Remote Authentication Dial-In

User Service (RADIUS) Authorization for Network Access Server (NAS) Management", RFC 5607, July 2009.

[RFC5966] Bellis, R., "DNS Transport over TCP - Implementation Requirements", RFC 5966, August 2010.

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

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, March 2011.

[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.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, March 2012.

[RFC6991] Schoenwaelder, J., "Common YANG Data Types", RFC 6991, July 2013.

10.2. Informative References

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, September 2003.

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

[RFC6557] Lear, E. and P. Eggert, "Procedures for Maintaining the Time Zone Database", BCP 175, RFC 6557, February 2012.

Authors' Addresses

Andy Bierman
YumaWorks

Email: andy@yumaworks.com

Martin Bjorklund
Tail-f Systems

Email: mbj@tail-f.com