SYSLOG YANG Model

draft-ietf-netmod-syslog-model-07

Abstract

This document describes a data model for the Syslog protocol which is used to convey event notification messages.

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 September 21, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

Operating systems, processes and applications generate messages indicating their own status or the occurrence of events. These messages are useful for managing and/or debugging the network and its services. The BSD Syslog protocol is a widely adopted protocol that is used for transmission and processing of the messages.

Since each process, application and operating system was written somewhat independently, there is little uniformity to the content of Syslog messages. For this reason, no assumption is made upon the formatting or contents of the messages. The protocol is simply designed to transport these event messages. No acknowledgement of the receipt is made.

Essentially, a Syslog process receives messages (from the kernel, processes, applications or other Syslog processes) and processes those. The processing involves logging to a local file, displaying on console, user terminal, and/or relaying to syslog processes on other machines. The processing is determined by the "facility" that originated the message and the "severity" assigned to the message by the facility.

We are using definitions of Syslog protocol from [RFC5424] in this draft.
1.1. Requirements Language

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 [RFC2119].

2. Problem Statement

This document defines a YANG [RFC6020] configuration data model that may be used to monitor and control one or more syslog processes running on a system. YANG models can be used with network management agents such as NETCONF [RFC6241] to install, manipulate, and delete the configuration of network devices.

This module makes use of the YANG "feature" construct which allows implementations to support only those Syslog features that lie within their capabilities.

This module can be used to configure the SYSLOG application conceptual layer [RFC5424].

3. Design of the SYSLOG Model

The syslog model was designed by comparing various syslog features implemented by various vendors’ in different implementations.

This draft addresses the common leafs between implementations and creates a common model, which can be augmented with proprietary features, if necessary. The base model is designed to be very simple for maximum flexibility.

Many vendors extend the list of facilities available for logging in their implementation. Here is an example that shows how additional facilities could be added to the list of available facilities (in this case two facilities are added):
module vendor-syslog-types {
    namespace "urn:vendor:params:xml:ns:yang:vendor-syslog-types";
    prefix vendor-syslogtypes;

    import ietf-syslog-types {
        prefix syslogtypes;
    }

    organization "Vendor, Inc.";
    contact
        "Vendor, Inc.
        Customer Service
        E-mail: syslog-yang@vendor.com";

    description
        "This module contains a collection of vendor-specific YANG type
definitions for SYSLOG."
;

    revision 2016-03-20 {
        description
            "Version 1.0";
        reference
            "Vendor SYSLOG Types: SYSLOG YANG Model";
    }

    identity vendor_specific_type_1 {
        base syslogtypes:syslog-facility;
    }

    identity vendor_specific_type_2 {
        base syslogtypes:syslog-facility;
    }
}

Syslog consists of message originators, and message distributors. The following diagram shows syslog messages flowing from a message originator, to message distributors where suppression filtering can take place.
Message Originators

+-------------+  +-------------+  +-------------+  +-------------+
|  Various    |  |     OS      |  |             |  |   Remote    |
| Components  |  |   Kernel    |  | Line Cards  |  |   Servers   |
+-------------+  +-------------+  +-------------+  +-------------+

+-------------+  +-------------+  +-------------+  +-------------+
|    SNMP     |  |  Interface  |  |   Standby   |  |   Syslog    |
|   Events    |  |   Events    |  |  Supervisor |  |   Itself    |
+-------------+  +-------------+  +-------------+  +-------------+

The leaves in the base syslog model log-actions container correspond to each message distributor:

console
log buffer(s)
log file(s)
remote relay(s)/collector(s)
terminal(s)
user session(s).

Optional features are used to specified functionality that is present in specific vendor configurations.
3.1. SYSLOG Module

A simplified graphical representation of the complete data tree is presented here.

Each node is printed as:

<status> <flags> <name> <opts> <type> <if-features>

<status> is one of:
+  for current
 x  for deprecated
 o  for obsolete

<flags> is one of:
rw  for configuration data
ro  for non-configuration data
-x  for rpcs
-n  for notifications

:name> is the name of the node

(<name>) means that the node is a choice node
:(<name>) means that the node is a case node

If the node is augmented into the tree from another module, its name is printed as <prefix>:<name>.

<opts> is one of:

?  for an optional leaf or choice
!  for a presence container
*  for a leaf-list or list
[<keys>] for a list’s keys

<type> is the name of the type for leafs and leaf-lists

If the type is a leafref, the type is printed as "-> TARGET", where TARGET is either the leafref path, with prefixed removed if possible.

<if-features> is the list of features this node depends on, printed within curly brackets and a question mark " {... } "?

module: ietf-syslog
+--rw syslog
   +--rw syslog
   +--rw log-actions
++--rw console! [console-action]?
  ++--rw log-selector
   ++--rw (selector-facility)
    |  ++--:(no-log-facility)
    |  |  ++--rw no-facilities? empty
    |  ++--:(log-facility)
    |  ++--rw log-facility* [facility]
    |     ++--rw facility union
    |     ++--rw severity union
    |     ++--rw severity-operator? enumeration {selector-sevop-config}?
    |  ++--rw pattern-match? string {selector-match-config}?
  ++--rw buffer [buffer-action]?
   ++--rw log-buffer* [name]
    |  ++--rw name string
   ++--rw log-selector
    |  ++--rw (selector-facility)
     |     ++--:(no-log-facility)
     |     |  ++--rw no-facilities? empty
     |     ++--:(log-facility)
     |     ++--rw log-facility* [facility]
     |     |  ++--rw facility union
     |     |  ++--rw severity union
     |     |  ++--rw severity-operator? enumeration {selector-sevop-config}?
     |     ++--rw pattern-match? string {selector-match-config}?
     |     ++--rw buffer-size-bytes? uint64 {buffer-limit-bytes}?
     |     ++--rw buffer-size-messages? uint64 {buffer-limit-messages}?
     |     ++--rw structured-data? boolean {structured-data-config}?
  ++--rw file
   ++--rw log-file* [name]
    |  ++--rw name inet:uri
   ++--rw log-selector
    |  ++--rw (selector-facility)
     |     ++--:(no-log-facility)
     |     |  ++--rw no-facilities? empty
     |     ++--:(log-facility)
     |     ++--rw log-facility* [facility]
     |     |  ++--rw facility union
     |     |  ++--rw severity union
     |     |  ++--rw severity-operator? enumeration {selector-sevop-config}?
     |     ++--rw pattern-match? string {selector-match-config}?
     |     ++--rw structured-data? boolean {structured-data-config}?
    |  ++--rw file-archive
     |     ++--rw number-of-files? uint32 {file-limit-size}?
     |     ++--rw max-file-size? uint64 {file-limit-size}?
     |     ++--rw rollover? uint32 {file-limit-duration}?
     |     ++--rw retention? uint16 {file-limit-duration}?
  ++--rw remote
   |  ++--rw destination* [name]
| +--rw name                      | string          |
| +--rw (transport)              |                |
| | +--:(tcp)                     |                |
| | | +--rw tcp                    |                |
| | | | +--rw address?             | inet:host      |
| | | | +--rw port?                | inet:port-number|
| | +--:(udp)                     |                |
| | | +--rw udp                    |                |
| | | | +--rw address?             | inet:host      |
| | | | +--rw port?                | inet:port-number|
| +--:(tls)                      |                |
| | +--rw tls                     |                |
| | | +--rw address?              | inet:host      |
| | | +--rw port?                 | inet:port-number|
| +--rw log-selector             |                |
| | +--rw (selector-facility)    |                |
| | | +--:(no-log-facility)       |                |
| | | | +--rw no-facilities?       | empty          |
| | | +--:(log-facility)          |                |
| | | | +--rw log-facility*        | [facility]     |
| | | | | +--rw facility            | union          |
| | | | | +--rw severity            | union          |
| | | | +--rw severity-operator?   | enumeration {selector-sevop-config}? |
| | +--rw pattern-match?          | string {selector-match-config}? |
| +--rw destination-facility?    | identityref    |
| +--rw source-interface?        | if:interface-ref|
| +--rw structured-data?         | boolean {structured-data-config}? |
| +--rw syslog-sign!             | (signed-messages-config)? |
| | +--rw cert-initial-repeat      | uint16         |
| | +--rw cert-resend-delay       | uint16         |
| | +--rw cert-resend-count       | uint16         |
| | +--rw sig-max-delay           | uint16         |
| | +--rw sig-number-resends      | uint16         |
| | +--rw sig-resend-delay        | uint16         |
| | +--rw sig-resend-count        | uint16         |
| +--rw terminal*                |                |
| | +--rw (terminal-action)       |                |
| | | +--rw log-selector           |                |
| | | | +--rw (selector-facility)  |                |
| | | | | +--:(no-log-facility)      |                |
| | | | | | +--rw no-facilities?      | empty          |
| | | | | +--:(log-facility)         |                |
| | | | | | +--rw log-facility*       | [facility]     |
| | | | | | | +--rw facility            | union          |
| | | | | | | +--rw severity            | union          |
| | | | | | +--rw severity-operator?  | enumeration {selector-sevop-config}? |
| | | | +--rw pattern-match?        | string {selector-match-config}? |
| | | +--rw terminal*              | string {terminal-facility-user-logging-config}? |
4. SYSLOG YANG Models

4.1. SYSLOG-TYPES Module

<CODE BEGINS> file "ietf-syslog-types.yang"
module ietf-syslog-types {
    namespace "urn:ietf:params:xml:ns:yang:ietf-syslog-types";
    prefix syslogtypes;

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

Wildes & Koushik Expires September 21, 2016 [Page 9]
This module contains a collection of YANG type definitions for SYSLOG.

Copyright (c) 2015 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 (http://tools.ietf.org/html/rfcXXXX); see the RFC itself for full legal notices.

reference

"RFC 5424: The Syslog Protocol";

revision 2016-03-20{

description

"Initial Revision";

reference

"RFC XXXX: SYSLOG YANG Model";
typedef severity {
    type enumeration {
        enum "emergency" {
            value 0;
            description "Emergency Level Msg";
        }
        enum "alert" {
            value 1;
            description "Alert Level Msg";
        }
        enum "critical" {
            value 2;
            description "Critical Level Msg";
        }
        enum "error" {
            value 3;
            description "Error Level Msg";
        }
        enum "warning" {
            value 4;
            description "Warning Level Msg";
        }
        enum "notice" {
            value 5;
            description "Notification Level Msg";
        }
        enum "info" {
            value 6;
            description "Informational Level Msg";
        }
        enum "debug" {
            value 7;
            description "Debugging Level Msg";
        }
    }
    description "The definitions for Syslog message severity as per RFC 5424.";
}
identity syslog-facility {
    description "This identity is used as a base for all syslog facilities as per RFC 5424.";
}

identity kern {
    base syslog-facility;
    description "The facility for kernel messages (0) as defined in RFC 5424.";
}

identity user {
    base syslog-facility;
    description "The facility for user-level messages (1) as defined in RFC 5424.";
}

identity mail {
    base syslog-facility;
    description "The facility for the mail system (2) as defined in RFC 5424.";
}

identity daemon {
    base syslog-facility;
    description "The facility for the system daemons (3) as defined in RFC 5424.";
}

identity auth {
    base syslog-facility;
    description "The facility for security/authorization messages (4) as defined in RFC 5424.";
}

identity syslog {
    base syslog-facility;
    description "The facility for messages generated internally by syslogd facility (5) as defined in RFC 5424.";
}

identity lpr {
    base syslog-facility;
    description "The facility for the line printer subsystem (6) as defined in
RFC 5424.

identity news {
  base syslog-facility;
  description
    "The facility for the network news subsystem (7) as defined in
    RFC 5424.";
}

identity uucp {
  base syslog-facility;
  description
    "The facility for the UUCP subsystem (8) as defined in RFC 5424.";
}

identity cron {
  base syslog-facility;
  description
    "The facility for the clock daemon (9) as defined in RFC 5424.";
}

identity authpriv {
  base syslog-facility;
  description
    "The facility for privileged security/authorization messages (10)
    as defined in RFC 5424.";
}

identity ftp {
  base syslog-facility;
  description
    "The facility for the FTP daemon (11) as defined in RFC 5424.";
}

identity ntp {
  base syslog-facility;
  description
    "The facility for the NTP subsystem (12) as defined in RFC 5424.";
}

identity audit {
  base syslog-facility;
  description
    "The facility for log audit messages (13) as defined in RFC 5424.";
}

identity console {
  base syslog-facility;
  description
    "The facility for log audit messages (13) as defined in RFC 5424.";
}
base syslog-facility;
description
"The facility for log alert messages (14) as defined in RFC 5424.";
}

identity cron2 {
    base syslog-facility;
description
"The facility for the second clock daemon (15) as defined in
RFC 5424.";
}

identity local0 {
    base syslog-facility;
description
"The facility for local use 0 messages (16) as defined in
RFC 5424.";
}

identity local1 {
    base syslog-facility;
description
"The facility for local use 1 messages (17) as defined in
RFC 5424.";
}

identity local2 {
    base syslog-facility;
description
"The facility for local use 2 messages (18) as defined in
RFC 5424.";
}

identity local3 {
    base syslog-facility;
description
"The facility for local use 3 messages (19) as defined in
RFC 5424.";
}

identity local4 {
    base syslog-facility;
description
"The facility for local use 4 messages (20) as defined in
RFC 5424.";
}

identity local5 {
base syslog-facility;
description
  "The facility for local use 5 messages (21) as defined in 
  RFC 5424.";
}

identity local6 {
  base syslog-facility;
description
  "The facility for local use 6 messages (22) as defined in 
  RFC 5424.";
}

identity local7 {
  base syslog-facility;
description
  "The facility for local use 7 messages (23) as defined in 
  RFC 5424.";
}

<CODE ENDS>

4.2. Syslog Module

<CODE BEGINS> file "ietf-syslog.yang"
module ietf-syslog {
  namespace "urn:ietf:params:xml:ns:yang:ietf-syslog";
  prefix syslog;

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-interfaces {
    prefix if;
  }

  import ietf-syslog-types {
    prefix syslogtypes;
  }

  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: Lou Berger

Wildes & Koushik Expires September 21, 2016 [Page 15]
This module contains a collection of YANG definitions for Syslog configuration.

Copyright (c) 2015 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 (http://tools.ietf.org/html/rfcXXXX); see the RFC itself for full legal notices.

reference
"RFC 5424: The Syslog Protocol
RFC 5848: Signed Syslog Messages";

revision 2016-03-20{
description
"Initial Revision";
reference
"RFC XXXX: SYSLOG YANG Model";
}

feature console-action {
description
  "This feature indicates that logging to the console is supported.";
}

feature buffer-action {
  description
  "This feature indicates that logging to an in-memory buffer is supported.";
}

feature buffer-limit-bytes {
  description
  "This feature indicates that local memory logging buffers are limited in size using a limit expressed in bytes.";
}

feature buffer-limit-messages {
  description
  "This feature indicates that local memory logging buffers are limited in size using a limit expressed in number of messages.";
}

feature file-limit-size {
  description
  "This feature indicates that file logging resources are managed using size and number limits.";
}

feature file-limit-duration {
  description
  "This feature indicates that file logging resources are managed using time based limits.";
}

feature terminal-action {
  description
  "This feature indicates that logging to a terminal is supported.";
}

feature terminal-facility-user-logging-config {
  description
  "This feature represents the ability to adjust log message settings for individual terminal devices.";
}
feature session-action {
    description
        "This feature indicates that logging to user
        CLI session is supported.";
}

feature session-facility-user-logging-config {
    description
        "This feature represents the ability to adjust
        log message settings for individual user CLI
        sessions.";
}

feature selector-sevop-config {
    description
        "This feature represents the ability to select messages
        using the additional operators equal to, or not equal to
        when comparing the Syslog message severity.";
}

feature selector-match-config {
    description
        "This feature represents the ability to select messages based
        on a Posix 1003.2 regular expression pattern match.";
}

feature structured-data-config {
    description
        "This feature represents the ability to log messages
        in structured-data format as per RFC 5424.";
}

feature signed-messages-config {
    description
        "This feature represents the ability to configure signed
        syslog messages according to RFC 5848.";
}

grouping syslog-severity {
    description
        "This grouping defines the Syslog severity which is used to
        select log messages.";
    leaf severity {
        type union {
            type syslogtypes:severity;
            type enumeration {
                enum all {
                    value -1;
                }
            }
        }
    }
}
enum none {
    value -2;
    description "This enum describes the case where no severities are selected.";
}

mandatory true;

description "This leaf specifies the Syslog message severity. When severity is specified, the default severity comparison is all messages of the specified severity and greater are selected. 'all' is a special case which means all severities are selected. 'none' is a special case which means that no selection should occur or disable this filter.";

leaf severity-operator {
    when '../severity != "all" and ../severity != "none"' {
        description "The severity-operator is not applicable for severity 'all' or severity 'none'";
    }
    if-feature selector-sevop-config;
    type enumeration {
        enum equals-or-higher {
            description "This enum specifies all messages of the specified severity and higher are logged according to the given log-action";
        }
        enum equals {
            description "This enum specifies all messages that are for the specified severity are logged according to the given log-action";
        }
        enum not-equals {
            description "This enum specifies all messages that are not for the specified severity are logged according to the given log-action";
        }
    }
}
default equals-or-higher;

description
"This leaf describes the option to specify how the severity comparison is performed."
;
}
)

grouping syslog-selector {

description
"This grouping defines a Syslog selector which is used to select log messages for the log-action (buffer, file, etc). Choose one of the following:
no-log-facility
log-facility [<facility> <severity>...]");
}

container log-selector {

description
"This container describes the log selector parameters for Syslog.";

choice selector-facility {

mandatory true;

description
"This choice describes the option to specify no facilities, or a specific facility which can be all for all facilities.";

case no-log-facility {

description
"This case specifies no facilities will match when comparing the Syslog message facility. This is a method that can be used to effectively disable a particular log-action (buffer, file, etc).";

leaf no-facilities {

type empty;

description
"This leaf specifies that no facilities are selected for this log-action.";

}
)

case log-facility {

description
"This case specifies one or more specified facilities will match when comparing the Syslog message facility.";

list log-facility {

key facility;

description
"This list describes a collection of Syslog facilities and severities.";

leaf facility {

}}
type union {
  type identityref {
    base syslogtypes:syslog-facility;
  }
  type enumeration {
    enum all {
      description "This enum describes the case where all facilities are requested.";
    }
  }
}

leaf pattern-match {
  if-feature selector-match-config;
  type string;
  description "This leaf describes a Posix 1003.2 regular expression string that can be used to select a Syslog message for logging. The match is performed on the RFC 5424 SYSLOG-MSG field.";
}

grouping syslog-structured-data {
  description "This grouping defines the Syslog structured data option which is used to select the format used to write log messages.";
  leaf structured-data {
    if-feature structured-data-config;
    type boolean;
    default false;
    description "This leaf describes how log messages are written to the log file. If true, messages will be written with one or more STRUCTURED-DATA elements as per RFC5424; if false, messages will be written with STRUCTURED-DATA = NILVALUE.";
  }
}
container syslog {
    description
    "This container describes the configuration parameters for
    Syslog.";
}
container log-actions {
    description
    "This container describes the log-action parameters
    for Syslog.";
}
container console {
    if-feature console-action;
    presence "Enables logging console configuration";
    description
    "This container describes the configuration parameters for
    console logging.";
    uses syslog-selector;
}
container buffer {
    if-feature buffer-action;
    description
    "This container describes the configuration parameters for
    local memory buffer logging. The buffer is circular in
    nature, so newer messages overwrite older messages after
    the buffer is filled. The method used to read syslog messages
    from the buffer is supplied by the local implementation.";
}
list log-buffer {
    key name;
    description
    "This list describes a collection of local logging
    memory buffers. If buffer size limits are not supplied, it
    is assumed that the local implementation defined
    limits will be used.";
    leaf name {
        type string;
        description
        "This leaf specifies the name of the log buffer.";
    }
    uses syslog-selector;
    leaf buffer-size-bytes {
        if-feature buffer-limit-bytes;
        type uint64;
        units "bytes";
        description
        "This leaf configures the amount of memory (in bytes) that
        will be dedicated to the local memory logging buffer.
        The default value varies by implementation.";
    }
    leaf buffer-size-messages {
        if-feature buffer-limit-messages;
    }
type uint64;
units "log messages";

description
 "This leaf configures the amount number of log messages that can be stored in the local memory logging buffer. The default value varies by implementation."
);

uses syslog-structured-data;
}
}

container file {  
description
 "This container describes the configuration parameters for file logging. If file-archive limits are not supplied, it is assumed that the local implementation defined limits will be used.";

list log-file {
  key "name";

description
 "This list describes a collection of local logging files.";

leaf name {
  type inet:uri {
    pattern 'file:.*';
  }

description
 "This leaf specifies the name of the log file which MUST use the uri scheme file:."
}

uses syslog-selector;
uses syslog-structured-data;
}

container file-archive {

description
 "This container describes the configuration parameters for log file archiving.";

leaf number-of-files {
  if-feature file-limit-size;
  type uint32;

description
 "This leaf specifies the maximum number of log files retained. Specify 1 for implementations that only support one log file.";
}

leaf max-file-size {
  if-feature file-limit-size;
  type uint64;
  units "megabytes";

description
"This leaf specifies the maximum file size for log files. Specify 1 for implementations that only support one log file.";
}

uses syslog-selector;
uses syslog-structured-data;
}
"This leaf specifies the maximum log file size."
}
leaf rollover {
  if-feature file-limit-duration;
  type uint32;
  units "minutes";
  description
  "This leaf specifies the length of time that log
  events should be written to a specific log file.
  Log events that arrive after the rollover period
  cause the current log file to be closed and a new
  log file to be opened.";
}
leaf retention {
  if-feature file-limit-duration;
  type uint16;
  units "hours";
  description
  "This leaf specifies the length of time that
  completed/closed log event files should be stored
  in the file system before they are deleted.";
}
}
}
container remote {
  description
  "This container describes the configuration parameters for
  remote logging.";
list destination {
  key "name";
  description
  "This list describes a collection of remote logging
  destinations.";
leaf name {
  type string;
  description
  "An arbitrary name for the endpoint to connect to.";
}
choice transport {
  mandatory true;
  description
  "This choice describes the transport option.";
  case tcp {
    container tcp {
      description
      "This container describes the TCP transport
      options.";
  }
leaf address {
    type inet:host;
    description "The leaf uniquely specifies the address of
        the remote host. One of the following must be specified: an ipv4 address, an ipv6
        address, or a host name.";
}

leaf port {
    type inet:port-number;
    default 514;
    description "This leaf specifies the port number used to
        deliver messages to the remote server.";
}

case udp {
    container udp {
        description "This container describes the UDP transport
            options.";
        reference "RFC 5426: Transmission of Syslog Messages over UDP";
        leaf address {
            type inet:host;
            description "The leaf uniquely specifies the address of
                the remote host. One of the following must be specified: an ipv4 address, an ipv6
                address, or a host name.";
        }
        leaf port {
            type inet:port-number;
            default 514;
            description "This leaf specifies the port number used to
                deliver messages to the remote server.";
        }
    }
}

case tls {
    container tls {
        description "This container describes the TLS transport options.";
        reference "RFC 5878: Transmission of Syslog Messages over TLS";
    }
}
"RFC 5425: Transport Layer Security (TLS) Transport Mapping for Syslog ";
leaf address {
  type inet:host;
  description
  "The leaf uniquely specifies the address of
  the remote host. One of the following must be
  specified: an ipv4 address, an ipv6 address,
  or a host name.";
}
leaf port {
  type inet:port-number;
  default 6514;
  description
  "This leaf specifies the port number used to
  deliver messages to the remote server.";
}
}
}
}
uses syslog-selector;
leaf destination-facility {
  type identityref {
    base syslogtypes:syslog-facility;
  }
  default syslogtypes:local7;
  description
  "This leaf specifies the facility used in messages
  delivered to the remote server.";
}
leaf source-interface {
  type if:interface-ref;
  description
  "This leaf sets the source interface for the remote
  Syslog server. Either the interface name or the
  interface IP address can be specified. If not set,
  messages sent to a remote syslog server will
  contain the IP address of the interface the syslog
  message uses to exit the network element";
}
uses syslog-structured-data;
container syslog-sign {
  if-feature signed-messages-config;
  presence
  "If present, syslog-sign is activated.";
  description
  "This container describes the configuration
  parameters for signed syslog messages as described
by RFC 5848.

reference

"RFC 5848: Signed Syslog Messages"

leaf cert-initial-repeat {
  type uint16;
  mandatory true;
  description
  "This leaf specifies the number of times each
  Certificate Block should be sent before the first
  message is sent."
}

leaf cert-resend-delay {
  type uint16;
  mandatory true;
  description
  "This leaf specifies the maximum time delay in
  seconds until resending the Certificate Block."
}

leaf cert-resend-count {
  type uint16;
  mandatory true;
  description
  "This leaf specifies the maximum number of other
  syslog messages to send until resending the
  Certificate Block."
}

leaf sig-max-delay {
  type uint16;
  mandatory true;
  description
  "This leaf specifies when to generate a new
  Signature Block. If this many seconds have
  elapsed since the message with the first message
  number of the Signature Block was sent, a new
  Signature Block should be generated."
}

leaf sig-number-resends {
  type uint16;
  mandatory true;
  description
  "This leaf specifies the number of times a
  Signature Block is resent. (It is recommended to
  select a value of greater than 0 in particular
  when the UDP transport [RFC5426] is used.)"
}

leaf sig-resend-delay {
  type uint16;
  mandatory true;
description
"This leaf specifies when to send the next Signature Block transmission based on time. If this many seconds have elapsed since the previous sending of this Signature Block, resend it."
}
leaf sig-resend-count {
  type uint16;
  mandatory true;
  description
  "This leaf specifies when to send the next Signature Block transmission based on a count. If this many other syslog messages have been sent since the previous sending of this Signature Block, resend it."
}
}
}
}
}
container terminal {
  if-feature terminal-action;
  description
  "This container describes the configuration parameters for the terminal logging configuration."
}
container all-terminals {
  presence "Enables logging to all terminals."
  description
  "This container describes the configuration parameters for all terminals."
  uses syslog-selector;
}
}
list terminal {
  if-feature terminal-facility-user-logging-config;
  key "name";
  description
  "This list describes a collection of "
  leaf name {
    type string;
    description
    "This leaf uniquely describes a terminal which will receive log messages."
  }
  uses syslog-selector;
}
}
}
container session {
  if-feature session-action;
  description
"This container describes the configuration parameters for user CLI session logging configuration.";
container all-users {
    presence "Enables logging to all user sessions.";
    description "This container describes the configuration parameters for all users.";
    uses syslog-selector;
}
list user {
    if-feature session-facility-user-logging-config;
    key "name";
    description "This list describes a collection of user names.";
    leaf name {
        type string;
        description "This leaf uniquely describes a user name which is the login name of the user whose session is to receive log messages.";
    }
    uses syslog-selector;
}
<CODE ENDS>

4.3. A Syslog Example

Requirement:
Enable console logging of syslogs of severity critical

Here is the example syslog configuration xml:
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <edit-config>
        <target>
            <candidate/>
        </target>
        <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
                <log-actions>
                    <console>
                        <log-selector>
                            <log-facility>
                                <facility>all</facility>
                            </log-facility>
                        </log-selector>
                    </console>
                </log-actions>
            </syslog>
        </config>
    </edit-config>
</rpc>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>

Enable remote logging of syslogs to udp destination 1.1.1.1 for facility auth, severity error

<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <candidate/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
        <log-actions>
          <remote>
            <destination>
              <name>remote1</name>
              <udp>
                <address>1.1.1.1</address>
              </udp>
            </destination>
            <log-selector>
              <log-facility>
                <severity>error</severity>
              </log-facility>
            </log-selector>
          </remote>
        </log-actions>
      </syslog>
    </config>
  </edit-config>
</rpc>
5. Acknowledgements

The authors wish to thank the following who commented on versions 01 through 06 of this proposal:

Martin Bjorklund <mbjorklu@cisco.com>
Jim Gibson <gibson@cisco.com>
Jeffrey Haas <jhaas@pfrc.org>
John Heasley <heas@shrubbery.net>
Giles Heron <giheron@cisco.com>
Lisa Huang <yihuan@cisco.com>
Jeffrey K Lange <jeffrey.K.lange@ge.com>
Jan Lindblad <j.lindblad@cisco.com>
Chris Lonvick <lonvick@gmail.com>
Tom Petch <ietf@btconnect.com>
Juergen Schoenwaelder <j.schoenwaelder@jacobs-university.de>
Jason Sterne <jason.sterne@alcatel-lucent.com>
Peter Van Horne <petervh@cisco.com>
Bert Wijnen <bertietf@bwijnen.net>
Aleksandr Zhdankin <azhdanki@cisco.com>

6. IANA Considerations

This document registers two URIs in the IETF XML registry [RFC3688].

Following the format in RFC 3688, the following registration is requested to be made:


Registrant Contact: The IESG.

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

This document registers a YANG module in the YANG Module Names registry [RFC6020].


prefix: ietf-syslog-types reference: RFC XXXX
Following the format in RFC 3688, the following registration is requested to be made:


Registrant Contact: The IESG.

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

This document registers a YANG module in the YANG Module Names registry [RFC6020].

prefix: ietf-syslog reference: RFC XXXX

7. 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 [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the 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 (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

8. References

8.1. Normative References


Informative References


8.2. Informative References


Authors’ Addresses

Clyde Wildes (editor)
Cisco Systems Inc.
170 West Tasman Drive
San Jose, CA  95134
US

Phone: +1 408 527-2672
Email: cwildes@cisco.com
Kiran Koushik (editor)
Cisco Systems Inc.
12515 Research Blvd., Building 4
Austin, TX 78759
US

Phone: +1 512 378-1482
Email: kkoushik@cisco.com