Abstract

This document describes a data model for the configuration of syslog.

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 May 17, 2017.

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 RFC.

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].

1.2. Terminology

The term "originator" is defined in [RFC5424]: an "originator" generates syslog content to be carried in a message.

The terms "relay" and "collectors" are as defined in [RFC5424].

2. Problem Statement

This document defines a YANG [RFC6020] configuration data model that may be used to configure the syslog feature running on a system. YANG models can be used with network management protocols such as NETCONF [RFC6241] to install, manipulate, and delete the configuration of network devices.

The data model 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 layers as implemented on the target system [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 leaves 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.

Syslog consists of originators, and collectors. The following digram shows syslog messages flowing from an originator, to collectors where suppression filtering can take place.

Many vendors extend the list of facilities available for logging in their implementation. An example is included in Extending Facilities (Appendix A.1).
The leaves in the base syslog model actions container correspond to each message collector:

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

Within each action, a selector is used to filter syslog messages. A selector consists of two parts: one or more facility-severity
matches, and if supported via the select-match feature, an optional regular expression pattern match that is performed on the SYSLOG-MSG field.

The facility is one of a specific syslogtypes:syslog-facility, none, or all facilities. None is a special case that can be used to disable an action.

The severity is one of syslogtypes:severity, all severities, or none. None is a special case that can be used to disable a facility. When filtering severity, the default comparison is that all messages of the specified severity and higher are logged. This is shown in the model as ?default equals-or-higher?. This behavior can be altered if the select-sev-compare feature is enabled to specify: ?equals? to specify only this single severity; ?not-equals? to ignore that severity; ?equals-or-higher? to specify all messages of the specified severity and higher.

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 actions
      +--rw console!
        +--rw selector
          +--rw (selector-facility)
            +--:(facility)
            |  +--rw no-facilities? empty
            +--:(name)
            |  +--rw facility-list* [facility]
              +--rw facility union
              +--rw severity union
              |  +--rw compare? enumeration {select-sev-compare}?
            +--rw pattern-match? string {select-match}?
        +--rw buffer (buffer-action)?
          +--rw selector
            +--rw (selector-facility)
            |  +--:(facility)
            |  |  +--rw no-facilities? empty
            |  +--:(name)
            |  |  +--rw facility-list* [facility]
            |  |  |  +--rw facility union
            |  |  +--rw severity union
            |  |  |  +--rw compare? enumeration {select-sev-compare}?
            |  +--rw pattern-match? string {select-match}?
            +--rw structured-data? boolean {structured-data}?
            +--rw buffer-limit-bytes? uint64 {buffer-limit-bytes}?
            +--rw buffer-limit-messages? uint64 {buffer-limit-messages}?
        +--rw file
          +--rw log-file* [name]
+++rw name     inet:uri
+++rw selector
    +++rw (selector-facility)
    |      ++---(facility)
    |          |      +++rw no-facilities?     empty
    |          |      ++---(name)
    |          |          +++rw facility-list*     [facility]
    |          |          |      +++rw facility     union
    |          |          |      +++rw severity     union
    |          |          |      +++rw compare?      enumeration {select-sev-compare}?
    |          |      +++rw pattern-match?      string {select-match}?
    |      +++rw structured-data?    boolean {structured-data}?
    |      +++rw file-rotation
    |          |      +++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
    |          |          +++rw udp
    |          |          |      +++rw address?     inet:host
    |          |          |      +++rw port?     inet:port-number
    +++rw selector
    |      +++rw (selector-facility)
    |          |      ++---(facility)
    |          |          |      +++rw no-facilities?     empty
    |          |          |      ++---(name)
    |          |          |          +++rw facility-list*     [facility]
    |          |          |          |      +++rw facility     union
    |          |          |          |      +++rw severity     union
    |          |          |          |      +++rw compare?      enumeration {select-sev-compare}?
    |          |          |      +++rw pattern-match?      string {select-match}?
    |      +++rw structured-data?    boolean {structured-data}?
    |      +++rw facility-override?   identityref
    |      +++rw source-interface?    if:interface-ref
    +++rw signing-options!     {signed-messages}?
      |      +++rw cert-initial-repeat    uint16
      |      +++rw cert-resend-delay      uint16
      |      +++rw cert-resend-count      uint16
      |      +++rw max-delay      uint16
      |      +++rw number-resends      uint16
Figure 2. ietf-syslog Module Tree

4. Syslog YANG Modules

4.1. The ietf-syslog-types Module

This module references [RFC5424].

<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
      "WG Web:  <http://tools.ietf.org/wg/netmod/>
      WG List:  <mailto:netmod@ietf.org>
      WG Chair: Lou Berger
      <mailto:lberger@labn.net>
This module contains a collection of YANG type definitions for SYSLOG.

Copyright (c) 2016 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.

typedef severity {
    type enumeration {
        enum "emergency" {
            value 0;
            description
                "The severity level ‘Emergency’ indicating that the system is unusable.";
        }
    }
}
enum "alert" {
    value 1;
    description
    "The severity level 'Alert' indicating that an action must be taken immediately.";
}
enum "critical" {
    value 2;
    description
    "The severity level 'Critical' indicating a critical condition.";
}
enum "error" {
    value 3;
    description
    "The severity level 'Error' indicating an error condition.";
}
enum "warning" {
    value 4;
    description
    "The severity level 'Warning' indicating a warning condition.";
}
enum "notice" {
    value 5;
    description
    "The severity level 'Notice' indicating a normal but significant condition.";
}
enum "info" {
    value 6;
    description
    "The severity level 'Info' indicating an informational message.";
}
enum "debug" {
    value 7;
    description
    "The severity level 'Debug' indicating a debug-level message.";
}

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 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>

Figure 3. ietf-syslog-types Module

4.2. The ietf-syslog Module

This module imports typedefs from [RFC6021] and [RFC7223], and it
references [RFC5424], [RFC5425], [RFC5426], [RFC6587], and [RFC5848].

<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
          <mailto:lberger@labn.net>
This module contains a collection of YANG definitions for syslog configuration.

Copyright (c) 2016 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 key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and 'OPTIONAL' in the module text are to be interpreted as described in RFC 2119 (http://tools.ietf.org/html/rfc2119).

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 5426: Transmission of Syslog Messages over UDP
 RFC 6587: Transmission of Syslog Messages over TCP
 RFC 5848: Signed Syslog Messages";

revision 2016-11-13 {
  description
    "Initial Revision";
  reference
    "RFC XXXX: Syslog YANG Model";
}

feature buffer-action {
  description
    "This feature indicates that the local memory logging buffer action is supported.";
}
feature buffer-limit-bytes {
    description
    "This feature indicates that the local memory logging buffer
    is limited in size using a limit expressed in bytes.";
}

feature buffer-limit-messages {
    description
    "This feature indicates that the local memory logging buffer
    is limited in size using a limit expressed in number of log
    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 select-sev-compare {
    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 select-match {
    description
    "This feature represents the ability to select messages based
    on a Posix 1003.2 regular expression pattern match.";
}

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

feature signed-messages {
    description

"This feature represents the ability to configure signed syslog messages according to RFC 5848."


grouping log-severity {
  description "This grouping defines the severity value that is used to select log messages.";
  leaf severity {
    type union {
      type syslogtypes:severity;
      type enumeration {
        enum none {
          value -2;
          description "This enum describes the case where no severities are selected.";
        }
        enum all {
          value -1;
          description "This enum describes the case where all 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 compare {
    when '../severity != "all" and ../severity != "none"' {
      description "The compare leaf is not applicable for severity ‘all’ or severity ‘none’";
    }
  }
  if-feature select-sev-compare;
  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 selector {
  description
  "This grouping defines a syslog selector which is used to select log messages for the log-action (console, file, remote, etc). Choose one of the following:
  no-facility
  facility [<facility> <severity>...]";
container 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 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.

This case specifies one or more specified facilities will match when comparing the syslog message facility.

This list describes a collection of syslog facilities and severities.

The leaf uniquely identifies a syslog facility.

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.

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;
    type boolean;
    default false;
    description
        "This leaf describes how log messages are written. 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 {
    presence "Enables logging.";
    description
        "This container describes the configuration parameters for syslog.";
    container actions {
        description
            "This container describes the log-action parameters for syslog.";
    }
    container console {
        presence "Enables logging console configuration";
        description
            "This container describes the configuration parameters for console logging.";
        uses 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.";
        uses selector;
        uses structured-data;
        leaf buffer-limit-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.";
        }
    }
}
leaf buffer-limit-messages {
  if-feature buffer-limit-messages;
  type uint64;
  units "log messages";
  description
    "This leaf configures the number of log messages that
    will be dedicated to the local memory logging buffer.
    The default value varies by implementation.";
}
}

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 selector;
    uses structured-data;
  }
  container file-rotation {
    description
      "This container describes the configuration
      parameters for log file rotation.";
    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;
      description
        "This leaf configures the number of log messages that
        will be dedicated to the local memory logging buffer.
        The default value varies by implementation.";
    }
  }
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 forwarding syslog messages to remote relays or collectors.";
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 leaf specifies the maximum log file size.";
            }
        }
    }
}

Wildes & Koushik Expires May 17, 2017 [Page 22]
"This container describes the TCP transport options."

reference
"RFC 6587: Transmission of Syslog Messages over TCP"

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.";
    }
  }
}
uses selector;
uses structured-data;
leaf facility-override {
    type identityref {
        base syslogtypes:syslog-facility;
    }
    description
        "If specified, this leaf specifies the facility used
        to override the facility in messages delivered to the
        remote server."
}
leaf source-interface {
    type if:interface-ref;
    description
        "This leaf sets the source interface to be used to send
        message to the remote syslog server. 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";
}
container signing-options {
    if-feature signed-messages;
    presence
        "If present, syslog-signing options 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;
        units "seconds";
        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 number of times each Certificate
            Block should be sent before the first message is sent.";
    }
}
"This leaf specifies the maximum number of other syslog messages to send until resending the Certificate Block."

leaf max-delay {
  type uint16;
  units "seconds";
  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 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 resend-delay {
  type uint16;
  units "seconds";
  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 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 session {

}
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 selector;
}
list user {
    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 selector;
}
}
</CODE ENDS>

Figure 4. ietf-syslog Module

5. Usage Examples
Requirement:
Enable console logging of syslogs of severity critical

Here is the example syslog configuration xml:
<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <syslog xmlns="urn:ietf:params:xml:ns:yang:ietf-syslog"
    <actions>
      <console>
        <selector>
          <facility-list>
            <facility>all</facility>
            <severity>critical</severity>
          </facility>
        </selector>
      </console>
    </actions>
  </syslog>
</config>

Enable remote logging of syslogs to udp destination 2001:db8:a0b:12f0::1 for facility auth, severity error

<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <syslog xmlns="urn:ietf:params:xml:ns:yang:ietf-syslog"
    <actions>
      <remote>
        <destination>
          <name>remote1</name>
          <udp>
            <address>2001:db8:a0b:12f0::1</address>
          </udp>
        </destination>
      </remote>
    </actions>
  </syslog>
</config>

Figure 5. ietf-syslog Examples
6. Acknowledgements

The authors wish to thank the following who commented on this proposal:

Martin Bjorklund
Jim Gibson
Jeffrey Haas
John Heasley
Giles Heron
Lisa Huang
Mahesh Jethanandani
Jeffrey K Lange
Jan Lindblad
Chris Lonvick
Tom Petch
Juergen Schoenwaelder
Phil Shafer
Jason Sterne
Peter Van Horne
Bert Wijnen
Aleksandr Zhdankin

7. 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

8. 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.

8.1. Resource Constraints

Network administrators must take the time to estimate the appropriate memory limits caused by the configuration of actions/buffer using buffer-limit-bytes and/or buffer-limit-messages where necessary to limit the amount of memory used.

Network administrators must take the time to estimate the appropriate storage capacity caused by the configuration of actions/file using file-archive attributes to limit storage used.

It is the responsibility of the network administrator to ensure that the configured message flow does not overwhelm system resources.
8.2. Inappropriate Configuration

It is the responsibility of the network administrator to ensure that the messages are actually going to the intended recipients.

9. References

9.1. Normative References


9.2. Informative References

Appendix A.  Implementor Guidelines

A.1.  Extending Facilities

Many vendors extend the list of facilities available for logging in their implementation. Additional facilities may not work with the syslog protocol as defined in [RFC5424] and hence such facilities apply for local syslog-like logging functionality.

The following is an example that shows how additional facilities could be added to the list of available facilities (in this example two facilities are added):


module vendor-syslog-types-example {
    namespace "urn:vendor:params:xml:ns:yang:vendor-syslog-types";
    prefix vendor-syslogtypes;

    import ietf-syslog-types {
        prefix syslogtypes;
    }

    organization "Example, Inc.";
    contact
        "Example, Inc.
        Customer Service
        E-mail: syslog-yang@example.com"
    description
        "This module contains a collection of vendor-specific YANG type
definitions for SYSLOG."

    revision 2016-11-13 {
        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;
    }
}

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