NETCONF
Internet-Draft
Intended status: Standards Track
Expires: 4 August 2024

M. Jethanandani
Kloud Services
K. Watsen
Watsen Networks
1 February 2024

An HTTPS-based Transport for YANG Notifications
draft-ietf-netconf-https-notif-15

Abstract

This document defines a protocol for sending asynchronous event notifications similar to notifications defined in RFC 5277, but over HTTPS. YANG modules for configuring publishers are also defined. Examples are provided illustrating how to configure various publishers.

This document requires that the publisher is a "server" (e.g., a NETCONF or RESTCONF server), but does not assume that the receiver is a server.

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 https://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 4 August 2024.

Copyright Notice

Table of Contents

1. Introduction

This document defines a protocol for sending asynchronous event notifications similar to notifications defined in NETCONF Event Notifications [RFC5277], but over HTTPS. Using HTTPS, which is a secure form of HTTP Semantics [RFC9110], maximizes transport-level interoperability, while allowing for a variety of encoding options. The protocol supports HTTP/1.1: Message Syntax and Routing [RFC9112] and, HTTP/2 [RFC9113]. While the payload does not change between these versions of HTTP and HTTP/3 [RFC9114], the underlying transport does. Since NETCONF does not support QUIC: A UDP-Based Multiplexed and Secure Transport [RFC9000], support for HTTP/3 [RFC9114], is considered out of scope of this document.

This document defines support for JSON and XML; future efforts may define support for other encodings (e.g., binary). This document requires that the publisher is a "server" (e.g., a NETCONF or RESTCONF server), but does not assume that the receiver is a NETCONF or RESTCONF server. It does expect the receiver to be an HTTPS server to receive the notifications.

This document also defines two YANG 1.1 [RFC7950] modules that extend the data model defined in Subscription to YANG Notifications [RFC8639], enabling the configuration of HTTPS-based receivers.

An example module illustrating the configuration of a publisher not using the data model defined in RFC 8639 is also provided.

Configured subscriptions enable a server (e.g., a NETCONF or RESTCONF server), acting as a publisher of notifications, to proactively push notifications to external receivers without the receivers needing to first connect to the server, as is the case with dynamic subscriptions.

1.1. Applicability Statement

While the YANG modules have been defined as an augmentation of Subscription to YANG Notifications [RFC8639], the notification method defined in this document MAY be used outside of Subscription to YANG Notifications [RFC8639] by using some of the definitions from this module along with the grouping defined in Groupings for HTTP Clients and Servers [I-D.ietf-netconf-http-client-server]. For an example on how that can be done, see Section A.2.

1.2. Note to RFC Editor

This document uses several placeholder values throughout the document. Please replace them as follows and remove this section before publication.

RFC XXXX, where XXXX is the number assigned to this document at the time of publication.

RFC YYYY, where YYYY is the number assigned to [I-D.ietf-netconf-http-client-server].

2024-02-01 with the actual date of the publication of this document.

1.3. Abbreviations

Acronym	Expansion
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
SSE	Server-Sent Events
TCP	Transmission Control Protocol
TLS	Transport Layer Security

Table 1

1.4. Terminology

The key words "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] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.4.1. Terms Imported from other RFCs

The following terms are defined in Subscription to YANG Notifications [RFC8639].

* Publisher

* Receiver

* Subscribed Notifications

The following term is defined in RESTCONF Protocol [RFC8040].

* target resource

1.5. Tree Diagram

The tree diagram for the YANG modules defined in this document use annotations defined in YANG Tree Diagrams. [RFC8340].

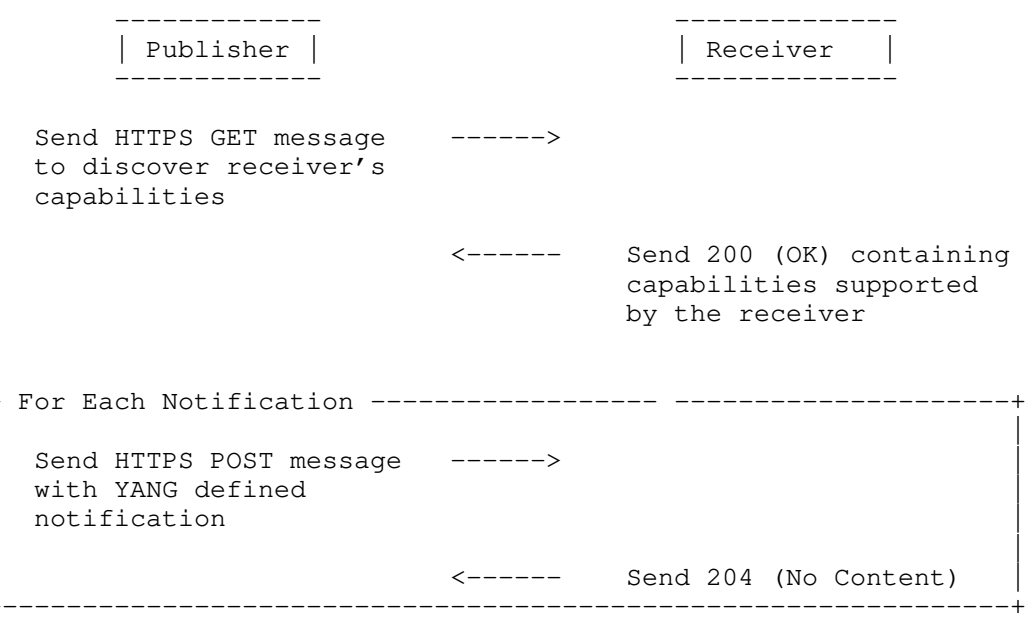
2. Overview of Publisher to Receiver Interaction

The protocol consists of two HTTP-based target resources presented by the receiver. These two resources share a common prefix that the publisher learns from a request it issues, as defined in section 3.2. If the data model in section 6.2 is used, this common prefix is defined by the "path" leaf in the "http-client-parameters" container.

* "capabilities": A target resource enabling the publisher to discover what optional capabilities a receiver supports. Publishers SHOULD query this target before sending any notifications or if ever an error occurs.

* "relay-notification": A target resource enabling the publisher to send one or more notification to a receiver. This document defines support for sending only one notification per message; a future effort MAY extend the protocol to send multiple notifications per message.

The protocol is illustrated in the diagram below:



Note that, for RFC 8639 configured subscriptions, the very first notification must be the "subscription-started" notification.

3. Discovering a Receiver's Capabilities

3.1. Applicability

For publishers using Subscription to YANG Notifications [RFC8639], dynamic discovery of a receiver's supported encoding is necessary only when the "/subscriptions/subscription/encoding" leaf is not configured, per the "encoding" leaf's description statement in the "ietf-subscribed-notification" module.

If the "encoding" leaf is not configured, and the publisher wants to send a notification in a particular format, without going through the setup operation of learning the receiver capabilities, it can do so, but has to be prepared for the case when it receives an error response, because the receiver does not support the format sent by the publisher.

3.2. Request

To learn the capabilities of a receiver, a publisher can issue an HTTPS GET request to the "capabilities" resource (see Section 2) on the receiver with "Accept" header set using the "application/xml" as defined in XML Media Types [RFC7303], and/or "application/json" as defined in JSON [RFC8259] media-types.

3.3. Response

The receiver responds with a "200 (OK)" message, having the "Content-Type" header set to either "application/xml" or "application/json" (which ever was selected), and containing in the response body a list of the receiver's capabilities encoded in the selected format.

Even though a YANG module is not defined for this interaction, the response body MUST conform to the following YANG-modeled format:

```
container receiver-capabilities {
  description
    "A container for a list of capabilities supported by
     the receiver.";
  leaf-list receiver-capability {
    type "inet:uri";
    description
      "A capability supported by the receiver.  A partial list of
       capabilities is defined in the 'Capabilities for HTTPS
       Notification Receivers' registry (see RFC XXXX). Additional
       custom capabilities MAY be defined.";
  }
}
```

As it is possible that the receiver may return custom capability URIs, the publisher MUST ignore any capabilities that it does not recognize.

3.4. Example

The publisher can send the following request to learn the receiver capabilities. In this example, the "Accept" states that the publisher wants to receive the capabilities response in XML but, if not supported, then in JSON.

```
GET /some/path/capabilities HTTP/1.1
Host: example.com
Accept: application/xml, application/json;q=0.5
```

If the receiver is able to reply using "application/xml", and assuming it is able to receive JSON and XML encoded notifications, and it is able to process the RFC 8639 state machine, the response might look like this:

```
HTTP/1.1 200 OK
Date: Wed, 26 Feb 2020 20:33:30 GMT
Server: example-server
Cache-Control: no-cache
Content-Type: application/xml

<receiver-capabilities>
  <receiver-capability>\
    urn:ietf:capability:https-notif-receiver:encoding:json\
  </receiver-capability>
  <receiver-capability>\
    urn:ietf:capability:https-notif-receiver:encoding:xml\
  </receiver-capability>
  <receiver-capability>\
    urn:ietf:capability:https-notif-receiver:sub-notif\
  </receiver-capability>
</receiver-capabilities>
```

If the receiver is unable to reply using "application/xml", the response might look like this:

```
HTTP/1.1 200 OK
Date: Wed, 26 Feb 2020 20:33:30 GMT
Server: example-server
Cache-Control: no-cache
Content-Type: application/json
Content-Length: nnn

{
   "receiver-capabilities": {
     "receiver-capability": [
       "urn:ietf:capability:https-notif-receiver:encoding:json",
       "urn:ietf:capability:https-notif-receiver:encoding:xml",
       "urn:ietf:capability:https-notif-receiver:sub-notif"
     ]
   }
}
```

4. Sending Event Notifications

4.1. Request

The publisher sends an HTTP POST request to the "relay-notification" resource (see Section 2) on the receiver with the "Content-Type" header set to either "application/json" or "application/xml" and a body containing the notification encoded using the specified format.

XML-encoded notifications are encoded using the format defined by NETCONF Event Notifications [RFC5277] for XML.

JSON-encoded notifications are encoded the same as specified in Section 6.4 in RESTCONF [RFC8040] with the following deviations:

* The notifications do not contain the "data:" prefix used by Server-Sent Events (SSE).

* Instead of saying that, for JSON-encoding purposes, the module name for the "notification" element is "ietf-restconf", the module name will instead be "ietf-https-notif".

4.2. Response

The response on success SHOULD be "204 (No Content)". In case of corrupted or malformed event, the response SHOULD be an appropriate HTTP error response.

4.3. Example

An XML-encoded notification might be sent as follows:

```
POST /some/path/relay-notification HTTP/1.1
Host: example.com
Content-Type: application/xml

<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2019-03-22T12:35:00Z</eventTime>
  <event xmlns="https://example.com/example-mod">
    <event-class>fault</event-class>
    <reporting-entity>
      <card>Ethernet0</card>
    </reporting-entity>
    <severity>major</severity>
  </event>
</notification>
```

A JSON-encoded notification might be sent as follows:

```
POST /some/path/relay-notification HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "ietf-https-notif:notification": {
    "eventTime": "2013-12-21T00:01:00Z",
    "example-mod:event" : {
      "event-class" : "fault",
      "reporting-entity" : { "card" : "Ethernet0" },
      "severity" : "major"
    }
  }
}
```

And, in either case, the response on success might be as follows:

```
HTTP/1.1 204 No Content
Date: Wed, 26 Feb 2020 20:33:30 GMT
Server: example-server
```

5. The "ietf-subscribed-notif-receivers" Module

5.1. Data Model Overview

This YANG module augments the "ietf-subscribed-notifications" module to define a choice of transport types that other modules such as the "ietf-https-notif-transport" module can use to define a transport specific receiver.

module: ietf-subscribed-notif-receivers

```
  augment /sn:subscriptions:
    +--rw receiver-instances
       +--rw receiver-instance* [name]
          +--rw name    string
          +--rw (transport-type)
  augment /sn:subscriptions/sn:subscription/sn:receivers/sn:receiver:
    +--rw receiver-instance-ref?   leafref
```

5.2. YANG Module

The YANG module imports Subscription to YANG Notifications [RFC8639].

```
<CODE BEGINS> file "ietf-subscribed-notif-receivers@2024-02-01.yang"
module ietf-subscribed-notif-receivers {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers";
  prefix "snr";

  import ietf-subscribed-notifications {
    prefix sn;
    reference
      "RFC 8639: Subscription to YANG Notifications";
  }

  organization
    "IETF NETCONF Working Group";

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

     Authors: Mahesh Jethanandani (mjethanandani at gmail dot com)
              Kent Watsen (kent plus ietf at watsen dot net)";

  description
    "This YANG module is implemented by Publishers implementing
     the 'ietf-subscribed-notifications' module defined in RFC 8639.

     While this module is defined in RFC XXXX, which primarily
     defines an HTTPS-based transport for notifications, this module
     is not HTTP-specific.  It is a generic extension that can be
     used by any 'notif' transport.

     This module defines two 'augment' statements.  One statement
     augments a 'container' statement called 'receiver-instances'
     into the top-level 'subscriptions' container.  The other
     statement, called 'receiver-instance-ref', augments a 'leaf'
     statement into each 'receiver' that references one of the
     afore mentioned receiver instances.  This indirection enables
     multiple configured subscriptions to send notifications to
     the same receiver instance.

     Copyright (c) 2024 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 Revised 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.

        The key words '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 (RFC 2119) (RFC 8174) when, and only when,
        they appear in all capitals, as shown here.";

    revision "2024-02-01" {
      description
        "Initial Version.";
      reference
        "RFC XXXX: An HTTPS-based Transport for YANG Notifications.";
    }

    augment "/sn:subscriptions" {
      container receiver-instances {
        description
          "A container for all instances of receivers.";

        list receiver-instance {
          key "name";

          leaf name {
            type string;
            description
              "An arbitrary but unique name for this receiver
               instance.";
          }

          choice transport-type {
            mandatory true;
            description
              "Choice of different types of transports used to
               send notifications.  The 'case' statements must
               be augmented in by other modules.";
          }
          description
            "A list of all receiver instances.";
        }
      }
      description
        "Augment the subscriptions container to define the
         transport type.";
    }
```

```
    augment
      "/sn:subscriptions/sn:subscription/sn:receivers/sn:receiver" {
      leaf receiver-instance-ref {
        type leafref {
          path "/sn:subscriptions/snr:receiver-instances/" +
               "snr:receiver-instance/snr:name";
        }
        description
          "Reference to a receiver instance.";
      }
      description
        "Augment the subscriptions container to define an optional
         reference to a receiver instance.";
    }
  }
  <CODE ENDS>
```

6. The "ietf-https-notif-transport" Module

6.1. Data Model Overview

This YANG module is a definition of a set of receivers that are interested in the notifications published by the publisher. The module contains the TCP, TLS and HTTPS parameters that are needed to communicate with the receiver. The module augments the "ietf-subscribed-notif-receivers" module to define a transport specific receiver.

As mentioned earlier, it uses a POST method to deliver the notification. The "http-receiver/tls/http-client-parameters/path" leaf defines the path for the resource on the receiver, as defined by "path-absolute" in URI Generic Syntax [RFC3986]. The user-id used by Network Configuration Access Control Model [RFC8341], is that of the receiver and is derived from the certificate presented by the receiver as part of "receiver-identity".

An abridged tree diagram representing the module is shown below.

```
   module: ietf-https-notif-transport

     augment /sn:subscriptions/snr:receiver-instances
               /snr:receiver-instance/snr:transport-type:
       +--:(https)
          +--rw https-receiver
             +--rw (transport)
             |  +--:(tls) {tls-supported}?
             |     +--rw tls
             |        +--rw tcp-client-parameters
             |        |  +--rw remote-address    inet:host
             |        |  +--rw remote-port?      inet:port-number
             |        |  +--rw local-address?    inet:ip-address
             |        |  |       {local-binding-supported}?
             |        |  +--rw local-port?       inet:port-number
             |        |  |       {local-binding-supported}?
             |        |  +--rw proxy-server! {proxy-connect}?
             |        |  |     ...
             |        |  +--rw keepalives! {keepalives-supported}?
             |        |        ...
             |        +--rw tls-client-parameters
             |        |  +--rw client-identity!
             |        |  |     ...
             |        |  +--rw server-authentication
             |        |  |     ...
             |        |  +--rw hello-params {tlscmn:hello-params}?
             |        |  |     ...
             |        |  +--rw keepalives {tls-client-keepalives}?
             |        |        ...
             |        +--rw http-client-parameters
             |           +--rw client-identity!
             |           |     ...
             |           +--rw proxy-connect! {proxy-connect}?
             |           |     ...
             |           +--rw path               string
             +--rw receiver-identity {receiver-identity}?
                +--rw cert-maps
                   +--rw cert-to-name* [id]
                      +--rw id             uint32
                      +--rw fingerprint    x509c2n:tls-fingerprint
                      +--rw map-type       identityref
                      +--rw name           string
```

6.2. YANG module

The YANG module imports A YANG Data Model for SNMP Configuration [RFC7407], Subscription to YANG Notifications [RFC8639], and YANG Groupings for HTTP Clients and HTTP Servers [I-D.ietf-netconf-http-client-server].

The YANG module is shown below.

```
<CODE BEGINS> file "ietf-https-notif-transport@2024-02-01.yang"
module ietf-https-notif-transport {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-https-notif-transport";
  prefix "hnt";

  import ietf-x509-cert-to-name {
    prefix x509c2n;
    reference
      "RFC 7407: YANG Data Model for SNMP Configuration.";
  }

  import ietf-subscribed-notifications {
    prefix sn;
    reference
      "RFC 8639: Subscription to YANG Notifications";
  }

  import ietf-subscribed-notif-receivers {
    prefix snr;
    reference
      "RFC XXXX: An HTTPS-based Transport for YANG Notifications.";
  }

  import ietf-http-client {
    prefix httpc;
    reference
      "RFC YYYY: YANG Groupings for HTTP Clients and HTTP Servers.";
  }

  organization
    "IETF NETCONF Working Group";

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

     Authors: Mahesh Jethanandani (mjethanandani at gmail dot com)
              Kent Watsen (kent plus ietf at watsen dot net)";
```

```
    description
      "This YANG module is implemented by Publishers that implement
       the 'ietf-subscribed-notifications' module defined in RFC 8639.

       This module augments a 'case' statement called 'https' into
       the 'choice' statement called 'transport-type' defined
       by the 'ietf-https-notif-transport' module defined in RFC XXXX.

       Copyright (c) 2024 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 Revised 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.

       The key words '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 (RFC 2119) (RFC 8174) when, and only when,
       they appear in all capitals, as shown here.";

    revision "2024-02-01" {
      description
        "Initial Version.";
      reference
        "RFC XXXX: An HTTPS-based Transport for YANG Notifications.";
    }

    feature receiver-identity {
      description
        "Indicates that the server supports filtering notifications
         based on the receiver's identity derived from its TLS
         certificate.";
    }

    identity https {
      base sn:transport;
      description
        "HTTPS transport for notifications.";
    }

    grouping https-receiver-grouping {
      description
```

```
        "A grouping that may be used by other modules wishing to
         configure HTTPS-based notifications without using RFC 8639.";
      uses httpc:http-client-stack-grouping {
        refine "transport/tcp" {
          // create the logical impossibility of enabling the
          // "tcp" transport (i.e., "HTTP" without the 'S').
          if-feature "not httpc:tcp-supported";
        }
        augment "transport/tls/tls/http-client-parameters" {
          leaf path {
            type string;
            mandatory true;
            description
              "A path to the target resources.  Under this
               path the receiver must support both the 'capabilities'
               and 'relay-notification' resource targets, as described
               in RFC XXXX.";
          }
          description
            "Augmentation to add a receiver-specific path for the
             'capabilities' and 'relay-notification' resources.";
        }
      }
      container receiver-identity {
        if-feature receiver-identity;
        description
          "Maps the receiver's TLS certificate to a local identity
           enabling access control to be applied to filter out
           notifications that the receiver may not be authorized
           to view.";
        container cert-maps {
          uses x509c2n:cert-to-name;
          description
            "The cert-maps container is used by a TLS-based HTTP
             server to map the HTTPS client's presented X.509
             certificate to a 'local' username. Specifically, the
             'name' field within the module is used along with
             'specified' identity to perform the match. If no
             matching and valid cert-to-name list entry is found,
             the publisher MUST close the connection, and MUST
             NOT send any notifications over it.";
          reference
            "RFC 7407: A YANG Data Model for SNMP Configuration.";
        }
      }
    }

    augment "/sn:subscriptions/snr:receiver-instances/" +
```

```
            "snr:receiver-instance/snr:transport-type" {
      case https {
        container https-receiver {
          description
            "The HTTPS receiver to send notifications to.";
          uses https-receiver-grouping;
        }
      }
      description
        "Augment the transport-type choice to include the 'https'
         transport.";
    }
  }
  <CODE ENDS>
```

7. Security Considerations

The YANG modules specified in this document define a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446]. The NETCONF Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

The YANG modules in this document make use of groupings that are defined in YANG Groupings for HTTP Clients and HTTP Servers [I-D.ietf-netconf-http-client-server], YANG Groupings for TLS Clients and TLS Servers [I-D.ietf-netconf-tls-client-server], and A YANG Data Model for SNMP Configuration [RFC7407]. Please see the Security Considerations section of those documents for considerations related to sensitivity and vulnerability of the data nodes defined in them. Additionally, the parameters defined in the tls-client-grouping in the ietf-tls-client module should follow the recommendations specified in Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). [RFC9325]

There are a number of data nodes defined in the YANG modules that 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. These are the subtrees and data nodes and their sensitivity/vulnerability:

* The "path" node in "ietf-subscribed-notif-receivers" module can be modified by a malicious user to point to an invalid URI. Worse still, it could point the URI of their choosing, exploit the vulnerable client, and if redirects are followed to the same URI, track its usage.

  The container "receiver-identity" contains nodes like "cert-maps" that are used by the HTTP server to map to the HTTPS client's certificate to a 'local' username. An unintended modification of these nodes will result in new connection requests be denied.

Some of the readable data nodes in the YANG modules 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. The model does not define any readable subtrees and data nodes that are particularly sensitive or vulnerable.

Some of the RPC operations in the YANG modules may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. The model does not define any RPC operations.

8. IANA Considerations

8.1. The "IETF XML" Registry

This document registers two URIs in the "ns" subregistry of the "IETF XML" registry [RFC3688]. Following the format in [RFC3688], the following registrations are requested:

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

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

8.2. The "YANG Module Names" Registry

This document registers two YANG modules in the "YANG Module Names" registry [RFC6020]. Following the format in [RFC6020], the following registrations are requested:

```
name:      ietf-subscribed-notif-receivers
namespace: urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers
prefix:    snr
reference: RFC XXXX

name:      ietf-https-notif-transport
namespace: urn:ietf:params:xml:ns:yang:ietf-https-notif-transport
prefix:    hnt
reference: RFC XXXX
```

8.3. Registration of 'yang-notif' URN Sub-namespace

This document requests that IANA register a new URN Sub-namespace within the "IETF URN Sub-namespace for Registered Protocol Parameter Identifiers" registry defined in [RFC3553].

Registry Name: yang-notif
Specification: RFC XXXX
Repository: "YANG Notifications" registry

8.4. Registration of 'https' URN Sub-namespace

This document requests that IANA register a new URN Sub-namespace within the "YANG Notifications" registry group defined in [RFC3553].

Registry Name: https-capability
Specification: RFC XXXX
Repository: "Capabilities for HTTPS Notification Receivers" registry

The following note shall be at the top of the registry:

This registry defines capabilities that can be supported by HTTPS-based notification receivers.

The fields for each registry are:

* URN
  - The name of the URN (required).
  - The URN must conform to the syntax described by [RFC8141].
  - The URN must begin with the string "urn:ietf:params:yang-notif:https-capability".
* Reference
  - The RFC that defined the URN.

- The RFC must be in the form "RFC <Number>: <Title>.

* Description

  - An arbitrary description of the capability.

  - The description should be no more than a few sentences.

  - The description is to be in English, but may contain UTF-8 characters as may be needed in some cases.

The update policy is "RFC Required".

Following is the initial assignment for this registry:

```
Record:
   URN:         urn:ietf:params:yang-notif:https-capability:encoding:json
   Reference:   RFC XXXX:An HTTPS-based Transport for YANG Notifications
   Description: Identifies support for JSON-encoded notifications.

Record:
   URN:         urn:ietf:params:yang-notif:https-capability:encoding:xml
   Reference:   RFC XXXX:An HTTPS-based Transport for YANG Notifications
   Description: Identifies support for XML-encoded notifications.

Record:
   URN:         urn:ietf:params:yang-notif:https-capability:sub-notif
   Reference:   RFC XXXX:An HTTPS-based Transport for YANG Notifications
   Description: Identifies support for state machine described in
                RFC 8639, enabling the publisher to send, e.g., the
                "subscription-started" notification.
```

9. References

9.1. Normative references

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", BCP 73, RFC 3553, DOI 10.17487/RFC3553, June 2003, <https://www.rfc-editor.org/info/rfc3553>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <https://www.rfc-editor.org/info/rfc5277>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC7303] Thompson, H. and C. Lilley, "XML Media Types", RFC 7303, DOI 10.17487/RFC7303, July 2014, <https://www.rfc-editor.org/info/rfc7303>.

[RFC7407] Bjorklund, M. and J. Schoenwaelder, "A YANG Data Model for SNMP Configuration", RFC 7407, DOI 10.17487/RFC7407, December 2014, <https://www.rfc-editor.org/info/rfc7407>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[RFC9112] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <https://www.rfc-editor.org/info/rfc9112>.

[RFC9113] Thomson, M., Ed. and C. Benfield, Ed., "HTTP/2", RFC 9113, DOI 10.17487/RFC9113, June 2022, <https://www.rfc-editor.org/info/rfc9113>.

[RFC9114] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <https://www.rfc-editor.org/info/rfc9114>.

[RFC9325] Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/info/rfc9325>.

[I-D.ietf-netconf-http-client-server] Watsen, K., "YANG Groupings for HTTP 1.1/2.0 Clients and HTTP Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-http-client-server-15, 26 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-http-client-server-15>.

[I-D.ietf-netconf-tls-client-server] Watsen, K., "YANG Groupings for TLS Clients and TLS Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-tls-client-server-36, 29 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-tls-client-server-36>.

9.2. Informative references

[RFC8141] Saint-Andre, P. and J. Klensin, "Uniform Resource Names (URNs)", RFC 8141, DOI 10.17487/RFC8141, April 2017, <https://www.rfc-editor.org/info/rfc8141>.

Appendix A. Configuration Examples

This non-normative section shows two examples for how the "ietf-https-notif-transport" module can be used to configure a publisher to send notifications to a receiver.

In both examples, the publisher, being an HTTPS client, is configured to send notifications to a receiver.

A.1. Using Subscribed Notifications (RFC 8639)

This example shows how an RFC 8639 [RFC8639] based publisher can be configured to send notifications to a receiver.

```
============== NOTE: '\' line wrapping per RFC 8792 ================

<?xml version="1.0" encoding="UTF-8"?>
<subscriptions
    xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications\
">
  <receiver-instances
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-recei\
```

```
   vers">
       <receiver-instance>
         <name>global-receiver-def</name>
         <https-receiver
             xmlns="urn:ietf:params:xml:ns:yang:ietf-https-notif-transp\
   ort"
             xmlns:x509c2n="urn:ietf:params:xml:ns:yang:ietf-x509-cert-\
   to-name">
           <tls>
             <tcp-client-parameters>
               <remote-address>receiver.example.com</remote-address>
               <remote-port>443</remote-port>
             </tcp-client-parameters>
             <tls-client-parameters>
               <server-authentication>
                 <ca-certs>
                   <local-definition>
                     <certificate>
                       <name>Server Cert Issuer #1</name>
                       <cert-data>base64encodedvalue==</cert-data>
                     </certificate>
                   </local-definition>
                 </ca-certs>
               </server-authentication>
             </tls-client-parameters>
             <http-client-parameters>
               <client-identity>
                 <basic>
                   <user-id>my-name</user-id>
                   <cleartext-password>my-password</cleartext-password>
                 </basic>
               </client-identity>
               <path
                   xmlns="urn:ietf:params:xml:ns:yang:ietf-https-notif-\
   transport">/some/path</path>
             </http-client-parameters>
           </tls>
           <receiver-identity>
             <cert-maps>
               <cert-to-name>
                 <id>1</id>
                 <fingerprint>11:0A:05:11:00</fingerprint>
                 <map-type>x509c2n:san-any</map-type>
               </cert-to-name>
             </cert-maps>
           </receiver-identity>
         </https-receiver>
       </receiver-instance>
```

```
    </receiver-instances>
    <subscription>
      <id>6666</id>
      <transport xmlns:ph="urn:ietf:params:xml:ns:yang:ietf-https-noti\
  f-transport">ph:https</transport>
      <stream-subtree-filter>
        <some-subtree-filter/>
      </stream-subtree-filter>
      <stream>some-stream</stream>
      <receivers>
        <receiver>
          <name>subscription-specific-receiver-def</name>
          <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ie\
  tf-subscribed-notif-receivers">global-receiver-def</receiver-instanc\
  e-ref>
        </receiver>
      </receivers>
    </subscription>
  </subscriptions>
  <truststore xmlns="urn:ietf:params:xml:ns:yang:ietf-truststore">
    <certificate-bags>
      <certificate-bag>
        <name>explicitly-trusted-server-ca-certs</name>
        <description>
          Trust anchors (i.e. CA certs) that are used to
          authenticate connections to receivers.  Receivers
          are authenticated if their certificate has a chain
          of trust to one of these CA certificates.
          certificates.
        </description>
        <certificate>
          <name>ca.example.com</name>
          <cert-data>base64encodedvalue==</cert-data>
        </certificate>
        <certificate>
          <name>Fred Flintstone</name>
          <cert-data>base64encodedvalue==</cert-data>
        </certificate>
      </certificate-bag>
    </certificate-bags>
  </truststore>
```

A.2. Not Using Subscribed Notifications

In the case that it is desired to use HTTPS-based notifications outside of Subscribed Notifications, an application-specific module would need to define the configuration for sending the notification.

Following is an example module. Note that the module "uses" the "https-receiver-grouping" grouping from the "ietf-https-notif-transport" module.

```
module example-custom-module {
  yang-version 1.1;
  namespace "http://example.com/example-custom-module";
  prefix "custom";

  import ietf-https-notif-transport {
    prefix "hnt";
    reference
      "RFC XXXX:
        An HTTPS-based Transport for Configured Subscriptions";
  }

  organization
    "Example, Inc.";

  contact
    "Support at example.com";

  description
    "Example of module not using Subscribed Notifications module.";

  revision "2024-02-01" {
    description
      "Initial Version.";
    reference
      "RFC XXXX: An HTTPS-based Transport for YANG Notifications.";
  }

  container example-module {
    description
      "Example of using HTTPS notif without having to
       implement Subscribed Notifications.";

    container https-receivers {
      description
        "A container of all HTTPS notif receivers.";
      list https-receiver {
       key "name";
        description
          "A list of HTTPS notif receivers.";
       leaf name {
         type string;
         description
           "A unique name for the https notif receiver.";
```

```
          }
           uses hnt:https-receiver-grouping;
         }
       }
     }
   }
```

Following is what the corresponding configuration looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<example-module xmlns="http://example.com/example-custom-module">
  <https-receivers>
    <https-receiver>
      <name>foo</name>
      <tls>
        <tcp-client-parameters>
          <remote-address>receiver.example.com</remote-address>
          <remote-port>443</remote-port>
        </tcp-client-parameters>
        <tls-client-parameters>
          <server-authentication>
            <ca-certs>
              <local-definition>
                <certificate>
                  <name>Server Cert Issuer #1</name>
                  <cert-data>base64encodedvalue==</cert-data>
                </certificate>
              </local-definition>
            </ca-certs>
          </server-authentication>
        </tls-client-parameters>
        <http-client-parameters>
          <client-identity>
            <basic>
              <user-id>my-name</user-id>
              <cleartext-password>my-password</cleartext-password>
            </basic>
          </client-identity>
          <path>/some/path</path>
        </http-client-parameters>
      </tls>
    </https-receiver>
  </https-receivers>
</example-module>
```

Acknowledgements

The authors would like to thank for following for lively discussions on list and in the halls (ordered by first name): Eric Voit, Henning Rogge, Martin Bjorklund, Reshad Rahman, and Rob Wilton.

In addition, the authors would also like to thank Quifang Ma for providing thoughtful comments as part of shepherd writeup.

Authors' Addresses

Mahesh Jethanandani
Kloud Services
Email: mjethanandani@gmail.com

Kent Watsen
Watsen Networks
Email: kent+ietf@watsen.net