NETCONF M. Jethanandani
Internet-Draft Kloud Services
Intended status: Standards Track K. Watsen
Expires: January 10, 2021 Watsen Networks
July 9, 2020

An HTTPS-based Transport for Configured Subscriptions
draft-ietf-netconf-https-notif-03

Abstract

This document defines a YANG data module for configuring HTTPS based configured subscription, as defined in RFC 8639. The use of HTTPS maximizes transport-level interoperability, while allowing for encoding selection from text, e.g. XML or JSON, to binary.

Status of This Memo

This Internet-Draft will expire on January 10, 2021.

Copyright Notice

Table of Contents

1. Introduction

Subscription to YANG Notifications [RFC8639] defines a YANG data module for configuring subscribed notifications. It defines a "subscriptions" container that contains a list of receivers, but it defers the configuration and management of those receivers to other documents. This document defines two YANG 1.1 [RFC7950] data modules, one for augmenting the Subscription to YANG Notifications [RFC8639] to add a transport type, and another for configuring and managing HTTPS based receivers for the notifications.

The first module allows for different transports to be configured for the same receiver instance. The second module describes how to enable the transmission of YANG modeled notifications, in the configured encoding (i.e., XML, JSON) over HTTPS. Notifications are delivered in the form of a HTTPS POST. The use of HTTPS maximizes transport-level interoperability, while the encoding selection pivots between implementation simplicity (XML, JSON) and throughput (text versus binary).

Configured subscriptions enable a 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 8.2.

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.

2020-07-10 with the actual date of the publication of this document.

1.3. Abbreviations

Acronym	Expansion
HTTP	Hyper Text Transport Protocol
HTTPS	Hyper Text Transport Protocol Secure
TCP	Transmission Control Protocol
TLS	Transport Layer Security

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. Subscribed Notifications

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

- Subscribed Notifications

1.5. Receiver and Publisher Interaction

The interaction between the receiver and the publisher can be of type "pipelining" or send multiple notifications as part of a "bundled-message", as defined in Notification Message Headers and Bundles [I-D.ietf-netconf-notification-messages]

1.5.1. Pipelining of messages

In the case of "pipelining", the flow of messages would look something like this.

```
       -------------                               --------------
       | Publisher |                               | Receiver   |
       -------------                               --------------

       Establish TCP             ------>

       Establish TLS             ------>

       Send HTTPS POST message
       with YANG defined         ------>
       notification #1

       Send HTTPS POST message
       with YANG defined         ------>
       notification #2
                                                   Send 204 (No Content)
                                 <------           for notification #1

                                                   Send 204 (No Content)
                                 <------           for notification #2

       Send HTTPS POST message
       with YANG defined         ------->
       notification #3

                                                   Send 204 (No Content)
                                 <------           for notification #3
```

The content of the exchange would look something like this.

Request:

```
    POST /some/path HTTP/1.1
    Host: my-receiver.my-domain.com
    Content-Type: application/yang-data+xml

    <notification
      xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
      <eventTime>2019-03-22T12:35:00Z</eventTime>
      <foo xmlns="https://example.com/my-foobar-module">
        ...
      </foo>
    </notification>

    <notification
      xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
      <eventTime>2019-03-22T12:35:00Z</eventTime>
      <bar xmlns="https://example.com/my-foobar-module">
        ...
      </bar>
    </notification>

    <notification
      xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
      <eventTime>2019-03-22T12:35:01Z</eventTime>
      <baz xmlns="https://example.com/my-foobar-module">
        ...
      </baz>
    </notification>
```

Response:

```
      HTTP/1.1 204 No Content
      Date: Fri, 03 Mar 2019 12:35:00 GMT
      Server: my-receiver.my-domain.com


      HTTP/1.1 204 No Content
      Date: Fri, 03 Mar 2019 12:35:00 GMT
      Server: my-receiver.my-domain.com

      HTTP/1.1 204 No Content
      Date: Fri, 03 Mar 2019 12:35:01 GMT
      Server: my-receiver.my-domain.com
```

2. Learning Receiver Capabilities

2.1. Introduction

To learn the capabilities of the receiver, the publisher can issue a HTTPS GET request with Accept-Type set to application/ietf-https-notif-cap+xml or application/ietf-https-notif-cap+json, with latter as the mandatory to implement, and the default in case the type is not specified. If the receiver supports capabilities such as binary encoding of data, it can return that as a capability in a response. Please note that, when used in conjunction with Subscription to YANG Notifications [RFC8639], dynamic discovery of the receiver's supported encoding is considered only when the "/subscriptions/subscription/encoding" leaf is not configured, per the "encoding" leaf's description statement.

2.2. Example

The publisher can send the following request to learn the receiver capabilities. The Accept-Type states its preferred order for Content-Type that it wants to receive starting with XML, and if not supported, to use JSON encoding. Currently, there is only one capability of binary encoding defined.

```
GET / HTTP/1.1
Host: example.com
Accept-Type: application/ietf-https-notif-cap+xml, application/ietf-https-
notif-cap+json
```

In case the receiver supports the first Accept-Type, its response should 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/ietf-https-notif-cap+xml
Content-Length: nnn

<receiver-capabilities>
  <receiver-capability>
    <urn:ietf:params:https-config:capability:binary-encoding:1.0>
  </receiver-capability>
</receiver-capabilities>
```

3. The "ietf-sub-notif-recv-list" Module

3.1. Data Model Overview

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

```
   module: ietf-sub-notif-recv-list
     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
```

3.2. YANG Module

```
<CODE BEGINS> file "ietf-sub-notif-recv-list@2020-07-10.yang"
module ietf-sub-notif-recv-list {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-sub-notif-recv-list";
  prefix "snrl";

  import ietf-subscribed-notifications {
    prefix sn;

    reference
      "I-D.ietf-netconf-subscribed-notifications";
  }

  organization
    "IETF NETCONF Working Group";

  contact
    "WG Web:   <http://tools.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
    "YANG module for augmenting Subscribed Notifications to add
     a transport type.
```

```
     Copyright (c) 2018 IETF Trust and the persons identified as
     the document authors.  All rights reserved.
     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD
     License set forth in Section 4.c of the IETF Trust's Legal
     Provisions Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 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 "2020-07-10" {
    description
      "Initial Version.";
    reference
      "RFC XXXX, YANG Data Module for HTTPS 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.";
        }
        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/snrl:receiver-instances/" +
             "snrl:receiver-instance/snrl:name";
      }
      description
        "Reference to a receiver instance.";
    }
    description
      "Augment the subscriptions container to define an optional
       reference to a receiver instance.";
  }

}
<CODE ENDS>
```

4. The "ietf-https-notif" Module

4.1. 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-sub-notif-recv-list module to define a transport specific receiver. As mentioned earlier, it uses POST method to deliver the notification. The attribute 'path' 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
     augment /sn:subscriptions/snrl:receiver-instances
               /snrl:receiver-instance/snrl:transport-type:
       +--:(https)
          +--rw https-receiver
             +--rw (transport)
             |  +--:(tcp) {tcp-supported,not httpc:tcp-supported}?
             |  |     ...
             |  +--:(tls) {tls-supported}?
             |        ...
             +--rw receiver-identity
                +--rw cert-maps
                      ...
```

4.2. YANG module

The YANG module imports Common YANG Data Types [RFC6991], A YANG Data Model for SNMP Configuration [RFC7407], JSON Encoding of Data Modeled with YANG [RFC7951], and Subscription to YANG Notifications [RFC8639].

The YANG module is shown below.

```
   <CODE BEGINS> file "ietf-https-notif@2020-07-10.yang"
   module ietf-https-notif {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-https-notif";
     prefix "hn";

     import ietf-subscribed-notifications {
       prefix sn;
       reference
         "I-D.ietf-netconf-subscribed-notifications";
     }

     import ietf-http-client {
       prefix httpc;

       reference
         "I-D.ietf-netconf-http-client-server";
     }

     import ietf-sub-notif-recv-list {
       prefix snrl;

       reference
         "RFC XXXX, YANG Data Module for HTTPS Notifications.";
```

```
     }

     import ietf-x509-cert-to-name {
       prefix x509c2n;

       reference
         "RFC 7407: YANG Data Model for SNMP Configuration.";
     }

     organization
       "IETF NETCONF Working Group";

     contact
       "WG Web:   <http://tools.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
       "YANG module for configuring HTTPS base configuration.

        Copyright (c) 2018 IETF Trust and the persons identified as
        the document authors.  All rights reserved.
        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD
        License set forth in Section 4.c of the IETF Trust's Legal
        Provisions Relating to IETF Documents
        (http://trustee.ietf.org/license-info).

        This version of this YANG module is part of RFC 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 "2020-07-10" {
       description
         "Initial Version.";
       reference
         "RFC XXXX, YANG Data Module for HTTPS Notifications.";
     }

     identity https {
       base sn:transport;
```

```
       description
         "HTTPS transport for notifications.";
     }

     augment "/sn:subscriptions/snrl:receiver-instances/" +
             "snrl:receiver-instance/snrl:transport-type" {
       case https {
         container https-receiver {
           description
             "HTTPS receiver for notification";

           uses httpc:http-client-stack-grouping {
             refine "transport/tcp" {
               // create the logical impossibility of enabling "tcp"
               // transport
               if-feature "not httpc:tcp-supported";
             }
             augment "transport/tls/tls/http-client-parameters" {
               leaf path {
                 type string;
                 description
                   "Relative URI to the target resource.";
               }
               description
                 "Augmentation to add a path to the target resource.";
             }
           }

           container receiver-identity {
             description
               "Specifies mechanism for identifying the receiver.
                The publisher MUST NOT include any content in a
                notification that the user is not 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. If no matching and
                  valid cert-to-name list entry is found, the publisher
                  MUST close the connection, and MUST NOT
                  not send any notifications over it.";
               reference
                 "RFC 7407: A YANG Data Model for SNMP Configuration.";
             }
           }
         }
```

```
       }
       description
         "Augment the transport-type choice to define this transport.";
     }
   }
   <CODE ENDS>
```

5. Security Considerations

The YANG module specified in this document defines 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.

There are a number of data nodes defined in this YANG module 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:

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

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

6. Receiving Event Notifications

Encoding notifications for the HTTPS notifications is the same as the encoding notifications as defined in RESTCONF [RFC8040] Section 6.4, with the following changes. Instead of saying that for JSON-encoding purposes, the module name for "notification" element will be "ietf-restconf, it will say that for JSON-encoding purposes, the module name for "notification" element will be "ietf-https-notif".

With those changes, the SSE event notification encoded JSON example that would be sent over the HTTPS notif transport would appear as follows:

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

7. IANA Considerations

This document registers two URI, two YANG module and two Media Types.

7.1. URI Registration

in the IETF XML registry [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made:

```
URI: urn:ietf:params:xml:ns:yang:ietf-http-notif
URI: urn:ietf:params:xml:ns:yang:ietf-sub-notif-recv-list
```

Registrant Contact: The IESG. XML: N/A, the requested URI is an XML namespace.

7.2. YANG Module Name Registration

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

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

name: ietf-sub-recv-list
namespace: urn:ietf:params:xml:ns:yang:ietf-sub-notif-recv-list
prefix: snrl
reference: RFC XXXX
```

7.3. Media Types

7.3.1. Media Type "application/ietf-https-notif-cap+xml

Type name: application

Subtype name: ietf-https-notif-cap+xml

Required parameters: None

Optional parameters: None

Encoding considerations:
    8-bit Each conceptual YANG data node is encoded according to the XML Encoding Rules and Canonical Format for the specific YANG data node type defined in YANG 1.1 [RFC7950].

Security considerations:
    Security considerations related to the generation and consumption of RESTCONF messages are discussed in Section NN of RFC XXXX.

    Additional security considerations are specific to the semantics of particular YANG data models. Each YANG module is expected to specify security considerations for the YANG data defined in that module.

Interoperability considerations: N/A

Published specification: RFC XXXX

Applications that use this media type:
    Instance document data parsers used within a protocol or automation tool that utilize YANG-defined data structures.

Fragment identifier considerations:
    Fragment identifiers for this type are not defined. All YANG data nodes are accessible as resources using the path in the request URI.

Additional information:

    Deprecated alias names for this type: N/A
    Magic number(s): N/A
    File extension(s): None
    Macintosh file type code(s): "TEXT"

Person & email address to contact for further information:
    See Author's Address section of RFC XXXX.

Intended usage: COMMON

Restrictions on usage: N/A

Author: See Author's Address section of RFC XXXX

Change controller:
    Internet Engineering Task Force (mailto:iesg@ietf.org)

Provisional registration? (standards tree only): no

7.3.2. Media Type "application/ietf-https-notif-cap+json

Type name: application

Subtype name: ietf-https-notif-cap+json

Required parameters: None

Optional parameters: None

Encoding considerations:
    8-bit Each conceptual YANG data node is encoded according to the XML Encoding Rules and Canonical Format for the specific YANG data node type defined in JSON Encoding of Data Modeled with YANG [RFC7951].

Security considerations:
    Security considerations related to the generation and consumption of RESTCONF messages are discussed in Section NN of RFC XXXX.

    Additional security considerations are specific to the semantics of particular YANG data models. Each YANG module is expected to specify security considerations for the YANG data defined in that module.

Interoperability considerations: N/A

Published specification: RFC XXXX

Applications that use this media type:
    Instance document data parsers used within a protocol or automation tool that utilize YANG-defined data structures.

Fragment identifier considerations:
    Fragment identifiers for this type are not defined. All YANG data nodes are accessible as resources using the path in the request URI.

Additional information:

    Deprecated alias names for this type: N/A
    Magic number(s): N/A

File extension(s): None
Macintosh file type code(s): "TEXT"

Person & email address to contact for further information:
See Author's Address section of RFC XXXX.

Intended usage: COMMON

Restrictions on usage: N/A

Author: See Author's Address section of RFC XXXX

Change controller:
Internet Engineering Task Force (mailto:iesg@ietf.org)

Provisional registration? (standards tree only): no

8. Examples

This section shows some examples in how the module can be used.

8.1. Subscribed Notification based Configuration

This example shows how a HTTPS client can be configured to send notifications to a receiver at address 192.0.2.1, port 443, a 'path', with server certificates, and the corresponding trust store that is used to authenticate a connection.

[note: '\' line wrapping for formatting only]

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <subscriptions
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notificatio\
ns">
    <receiver-instances
        xmlns="urn:ietf:params:xml:ns:yang:ietf-sub-notif-recv-list">
      <receiver-instance>
        <name>foo</name>
        <https-receiver
            xmlns="urn:ietf:params:xml:ns:yang:ietf-https-notif"
            xmlns:x509c2n="urn:ietf:params:xml:ns:yang:ietf-x509-cert-to-na\
me">
          <tls>
            <tcp-client-parameters>
              <remote-address>my-receiver.my-domain.com</remote-address>
              <remote-port>443</remote-port>
            </tcp-client-parameters>
```

```
            <tls-client-parameters>
              <server-authentication>
                <ca-certs>explicitly-trusted-server-ca-certs</ca-certs>
                <server-certs>explicitly-trusted-server-certs</server-certs>
              </server-authentication>
            </tls-client-parameters>
            <http-client-parameters>
              <client-identity>
                <basic>
                  <user-id>my-name</user-id>
                  <password>my-password</password>
                </basic>
              </client-identity>
              <path>/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>
      <stream-subtree-filter>foo</stream-subtree-filter>
      <stream>some-stream</stream>
      <receivers>
        <receiver>
          <name>my-receiver</name>
          <receiver-instance-ref
              xmlns="urn:ietf:params:xml:ns:yang:ietf-sub-notif-recv-list">\
foo</receiver-instance-ref>
        </receiver>
      </receivers>
    </subscription>
  </subscriptions>

  <truststore xmlns="urn:ietf:params:xml:ns:yang:ietf-truststore">
    <certificates>
      <name>explicitly-trusted-server-certs</name>
      <description>
        Specific server authentication certificates for explicitly
```

```
        trusted servers.  These are needed for server certificates
        that are not signed by a pinned CA.
      </description>
      <certificate>
        <name>Fred Flintstone</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
    </certificates>
    <certificates>
      <name>explicitly-trusted-server-ca-certs</name>
      <description>
        Trust anchors (i.e. CA certs) that are used to authenticate
        server connections.  Servers are authenticated if their
        certificate has a chain of trust to one of these CA
        certificates.
      </description>
      <certificate>
        <name>ca.example.com</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
    </certificates>
  </truststore>
</config>
```

8.2. Non Subscribed Notification based Configuration

In the case that it is desired to use HTTPS notif outside of Subscribed Notifications, there would have to be a module to define the configuration for where and how to send the notification, such as the following:

[note: '\' line wrapping for formatting only]

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

  import ietf-http-client {
    prefix httpc;
    reference
      "I-D.ietf-netconf-http-client-server";
  }

  organization
    "Example, Inc.";
```

```
     contact
       "Support at example.com";

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

     revision "2020-07-10" {
       description
         "Initial Version.";
       reference
         "RFC XXXX, YANG Data Module for HTTPS 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";

           leaf name {
             type string;
             description
               "A unique name for the https notif receiver.";
           }

           uses httpc:http-client-stack-grouping {
             refine "transport/tcp" {
               // create the logical impossibility of enabling "tcp"
               // transport
               if-feature "not httpc:tcp-supported";
             }
             augment "transport/tls/tls/http-client-parameters" {
               leaf path {
                 type string;
                 description
                   "Relative URI to the target resource.";
               }
               description
                 "Augmentation to add a path to the target resource.";
             }
           }
           description
```

```
             "Just include the grouping from ietf-http-client to
              realize the 'HTTPS stack'.";
         }
       }
     }
   }
```

This example shows how a HTTPS client can be configured to send notifications to a receiver at address 192.0.2.1, port 443, a 'path', with server certificates, and the corresponding trust store that is used to authenticate a connection.

[note: '\' line wrapping for formatting only]

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <example-module
      xmlns="http://example.com/example-custom-module">
    <https-receivers>
      <https-receiver>
        <name>foo</name>
        <tls>
          <tcp-client-parameters>
            <remote-address>my-receiver.my-domain.com</remote-address>
            <remote-port>443</remote-port>
          </tcp-client-parameters>
          <tls-client-parameters>
            <server-authentication>
              <ca-certs>explicitly-trusted-server-ca-certs</ca-certs>
              <server-certs>explicitly-trusted-server-certs</server-certs>
            </server-authentication>
          </tls-client-parameters>
          <http-client-parameters>
            <client-identity>
              <basic>
                <user-id>my-name</user-id>
                <password>my-password</password>
              </basic>
            </client-identity>
            <path>/some/path</path>
          </http-client-parameters>
        </tls>
      </https-receiver>
    </https-receivers>
  </example-module>

  <truststore xmlns="urn:ietf:params:xml:ns:yang:ietf-truststore">
```

```
    <certificates>
      <name>explicitly-trusted-server-certs</name>
      <description>
        Specific server authentication certificates for explicitly
        trusted servers.  These are needed for server certificates
        that are not signed by a pinned CA.
      </description>
      <certificate>
        <name>Fred Flintstone</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
    </certificates>
    <certificates>
      <name>explicitly-trusted-server-ca-certs</name>
      <description>
        Trust anchors (i.e. CA certs) that are used to authenticate
        server connections.  Servers are authenticated if their
        certificate has a chain of trust to one of these CA
        certificates.
      </description>
      <certificate>
        <name>ca.example.com</name>
        <cert>base64encodedvalue==</cert>
      </certificate>
    </certificates>
  </truststore>
</config>
```

8.3. Bundled Message

In the case of "bundled-message" as defined in Notification Message Headers and Bundles [I-D.ietf-netconf-notification-messages], something that this module supports, the flow of messages would look something like this.

```
       -------------                               --------------
       | Publisher |                               | Receiver   |
       -------------                               --------------
       Establish TCP             ------>
       Establish TLS             ------>
       Send HTTPS POST message
       with YANG defined         ------>
       notification #1
       Send HTTPS POST message
       with YANG defined         ------>
       notification #2
                                                   Send 204 (No Content)
                                 <------           for notification #1

                                                   Send 204 (No Content)
                                 <------           for notification #2
       Send HTTPS POST message
       with YANG defined         ------->
       notification #3
                                                   Send 204 (No Content)
                                 <------           for notification #3
```

The content of the exchange would look something like this.

```
   Request:
       POST /some/path HTTP/1.1
       Host: my-receiver.my-domain.com
       Content-Type: application/yang-data+xml
       <notification
         xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
         <eventTime>2019-03-22T12:35:00Z</eventTime>
         <foo xmlns="https://example.com/my-foobar-module">
           ...
         </foo>
       </notification>
       <notification
         xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
         <eventTime>2019-03-22T12:35:00Z</eventTime>
         <bar xmlns="https://example.com/my-foobar-module">
           ...
         </bar>
       </notification>
       <notification
         xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
         <eventTime>2019-03-22T12:35:01Z</eventTime>
         <baz xmlns="https://example.com/my-foobar-module">
           ...
         </baz>
       </notification>
   Response:
         HTTP/1.1 204 No Content
         Date: Fri, 03 Mar 2019 12:35:00 GMT
         Server: my-receiver.my-domain.com
         HTTP/1.1 204 No Content
         Date: Fri, 03 Mar 2019 12:35:00 GMT
         Server: my-receiver.my-domain.com
         HTTP/1.1 204 No Content
         Date: Fri, 03 Mar 2019 12:35:01 GMT
         Server: my-receiver.my-domain.com
```

9. Contributors

10. Acknowledgements

11. Normative references

[I-D.ietf-netconf-http-client-server]
Watsen, K., "YANG Groupings for HTTP Clients and HTTP Servers", draft-ietf-netconf-http-client-server-04 (work in progress), July 2020.

[I-D.ietf-netconf-notification-messages]
Voit, E., Jenkins, T., Birkholz, H., Bierman, A., and A. Clemm, "Notification Message Headers and Bundles", draft-ietf-netconf-notification-messages-08 (work in progress), November 2019.

[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>.

[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>.

[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>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[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>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>.

[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>.

[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>.

Authors' Addresses

Mahesh Jethanandani
Kloud Services

Email: mjethanandani@gmail.com

Kent Watsen
Watsen Networks
USA

Email: kent+ietf@watsen.net