Abstract

This document defines a YANG data module for configuring HTTPS based configured subscription, as defined in Subscribed Notifications (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 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 May 2, 2020.

Copyright Notice

Copyright (c) 2019 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 (https://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 BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents.
Subscribed Notifications [RFC8639] defines a YANG data module for configuring subscribed notifications. It even 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 a YANG [RFC7950] data module for configuring and managing HTTPS based receivers for the notifications. Such a configured receiver can be a third party collector, collecting events on behalf of receivers that want to correlate events from different publishers. 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.

This document describes how to enable the transmission of YANG modeled notifications, in the configured encoding (i.e., XML, JSON) over HTTPS. It comes 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).
1.1.  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.

2019-10-30 with the actual date of the publication of this document.

1.2.  Abbreviations

+---------+-------------------------------+
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>Hyper Text Transport Protocol</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
</tbody>
</table>
+---------+-------------------------------+

1.3.  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.

1.3.1.  Subscribed Notifications

The following terms are defined in Subscribed Notifications [RFC8639].

- Subscribed Notifications

1.4.  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]
### 1.4.1. Pipelining of messages

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

<table>
<thead>
<tr>
<th>Publisher</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establish TCP</td>
<td>-----&gt;</td>
</tr>
<tr>
<td>Establish TLS</td>
<td>-----&gt;</td>
</tr>
<tr>
<td>Send HTTPS POST message with YANG defined</td>
<td>-----&gt;</td>
</tr>
<tr>
<td>notification #1</td>
<td></td>
</tr>
<tr>
<td>Send HTTPS POST message with YANG defined</td>
<td>-----&gt;</td>
</tr>
<tr>
<td>notification #2</td>
<td></td>
</tr>
<tr>
<td>Send HTTPS POST message with YANG defined</td>
<td>-----&gt;</td>
</tr>
<tr>
<td>notification #3</td>
<td></td>
</tr>
</tbody>
</table>

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. YANG module

2.1. Overview

The 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 Subscribed Notifications [RFC8639] receiver container to create a reference to a receiver defined by the YANG module. As mentioned earlier, it uses POST method to deliver the notification. The attribute ‘path’ defines the absolute 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.

An abridged tree diagram representing the module is shown below.
module: ietf-https-notif
  +--rw receivers
      +--rw receiver* [name]
        +--rw tcp-params
          |  +--rw remote-address  inet:host
          |  +--rw remote-port?    inet:port-number
          |  +--rw local-address?   inet:ip-address
          |  +--rw local-port?      inet:port-number
          +--rw keepalives!
          ... |
        +--rw tls-params
          |  +--rw client-identity
          |  |  ... |
          |  +--rw server-authentication
          |  |  ... |
          |  +--rw hello-params {tls-client-hello-params-config}?
          |  |  ... |
          |  +--rw keepalives! {tls-client-keepalives}?
          |  |  ... |
          +--rw http-params
            |  +--rw protocol-version?  enumeration
            |  +--rw client-identity
            |  |  ... |
            |  +--rw proxy-server! {proxy-connect}?
            |  |  ... |
            |  +--rw path?              inet:uri
          +--rw receiver-identity
          +--rw cert-maps
          ... |

augment /sn:subscriptions/sn:subscription/sn:receivers/sn:receiver:
  +--rw receiver-ref?  -> /receivers/receiver/name

2.2. YANG module

The YANG module imports Common YANG Data Types [RFC6991], A YANG Data Model for SNMP Configuration [RFC7407], and Subscription to YANG Notifications [RFC8639].

<CODE BEGINS> file "ietf-https-notif@2019-10-30.yang"
module ietf-https-notif {
  yang-version 1.1;
  prefix "hsn";

  import ietf-inet-types {

Jethanandani & Watsen Expires May 2, 2020 [Page 7]
prefix inet;
reference
   "RFC 6991: Common YANG Data Types.";
}

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

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

import ietf-tcp-client {
   prefix tcpc;
}

import ietf-tls-client {
   prefix tlsc;
}

import ietf-http-client {
   prefix httpc;
}

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
This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.

revision "2019-10-30" {
  description
    "Initial Version.";
  reference
    "RFC XXXX, YANG Data Module for HTTPS Notifications.";
}

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

container receivers {
  list receiver {
    key "name";

    leaf name {
      type string;
      description
        "A name that uniquely identifies this receiver.";
    }

  }

  container tcp-params {
    uses tcpc:tcp-client-grouping;
    description
      "TCP client parameters.";
  }

  container tls-params {
    description
      "TLS client parameters.";
    uses tlsc:tls-client-grouping;
  }

  container http-params {
    description
      "HTTP client parameters.";
    uses httpc:http-client-grouping;

    leaf path {
      type string;
      description
        "Path to the resource to notify.

      
      
Jethanandani & Watsen      Expires May 2, 2020                  

}
type inet:uri;

description
"The absolute path for the resource on the remote HTTPS server. The absolute path as specified in RFC 3986 as 'path-absolute'."

reference
"RFC 3986: URI Generic Syntax."

}

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 {

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 send any notifications over it."

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

}

augment "/sn:subscriptions/sn:subscription/sn:receivers/sn:receiver" {

description
"Augment the subscriptions container to define the receiver."

}
3. 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:

4. IANA Considerations

This document registers one URI and one YANG module.

4.1. URI Registration

in the IETF XML registry [RFC3688] [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.
## 4.2. YANG Module Name Registration

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

```
name: ietf-https-notif
prefix: hn
reference: RFC XXXX
```

## 5. Examples

This section tries to show some examples in how the model can be used.

### 5.1. HTTPS Configured Subscription

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.

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <receivers
    xmlns:x509c2n="urn:ietf:params:xml:ns:yang:ietf-x509-cert-to-name">
    <receiver>
      <name>foo</name>
      <tcp-params>
        <remote-address>my-receiver.my-domain.com</remote-address>
        <remote-port>443</remote-port>
      </tcp-params>
      <tls-params>
        <server-authentication>
          <ca-certs>explicitly-trusted-server-ca-certs</ca-certs>
          <server-certs>explicitly-trusted-server-certs</server-certs>
        </server-authentication>
      </tls-params>
      <http-params>
        <client-identity>
          <basic>
            <user-id>my-name</user-id>
            <password>my-password</password>
          </basic>
        </client-identity>
      </http-params>
    </receiver>
  </receivers>
</config>
```
</basic>
</client-identity>
<path>/some/path</path>
</http-params>
<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>
</receiver>
</receivers>

<subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
<subscription>
{id>6666</id>
<stream-subtree-filter>foo</stream-subtree-filter>
<stream>some-stream</stream>
<receivers>
<receiver>
<name>my-receiver</name>
<receiver-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>
</truststore>
<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>
</truststore>
</config>

6. Contributors

7. Acknowledgements

8. Normative references

[I-D.ietf-netconf-notification-messages]


Authors’ Addresses
Mahesh Jethanandani
VMware

Email: mjethanandani@gmail.com

Kent Watsen
Watsen Networks
USA

Email: kent+ietf@watsen.net