Module Tag Operations
draft-bierman-netconf-module-tag-ops-00

Abstract

This document describes enhancements to existing NETCONF and RESTCONF (NMDA) operations for using module tags to represent YANG datastore content. This can simplify usage of these operations by a client.

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 September 11, 2019.

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 the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

There is a need for standard mechanisms to allow NETCONF [RFC6241] and RESTCONF [RFC8040] protocol operations, as well as NMDEM operations for NETCONF [RFC8526] and RESTCONF [RFC8527], to utilize the module tag mapping definitions defined in [I-D.ietf-netmod-module-tags]. Netconf Access Control rules defined in [RFC8341] can also utilize module tags to simplify access control rule configuration.

1.1. 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.
1.1.1. NMDA

The following terms are defined in the Network Management Datastore Architecture (NMDA) [RFC8342]:

- configuration
- client
- datastore
- notification
- operational state
- operational state datastore
- server

1.1.2. RESTCONF

The following terms are defined in [RFC8040]:

- data resource
- target resource

1.1.3. YANG

The following terms are defined in [RFC7950]:

- choice
- container
- data model
- data node
- grouping
- leaf
- leaf-list
- list
1.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

2. Module Tag Operations

A module tag is a string associated with a module name. Modules are associated with the same module tag for the purpose of simplifying protocol operations, and other tool-specific operations.

The definition and management of module tags is defined in the ietf-module-tags module. This document defines augmentations to NETCONF protocol operations that use module tags defined in that module to represent YANG datastore content instead of a list of module names or data nodes.

A server vendor and operator can install module to module tag mappings on a server, using the ietf-module-tags YANG module. Support for that module is required to utilize the mechanisms defined in this document.

The following tree diagram shows the 4 separate augmentations defined in this module:

```
module: ietf-module-tag-ops
  augment /ncds:get-data/ncds:input:
    +---w module-tag*   tags:tag
  augment /nacm:nacm/nacm:rule-list/nacm:rule/nacm:rule-type:
    +++-:(module-tags)
      +---rw module-tag*   tags:tag
  augment /nc:get-config/nc:input:
    +---w module-tag*   tags:tag
  augment /nc:get/nc:input:
    +---w module-tag*   tags:tag
```

2.1. Module Tag Filters

Data retrieval filters based on module tags allow an operator to easily include only data of specific interest, without having to know the exact path identifiers for these objects within the datastore. Module tags can be pre-defined in the YANG module or YANG Module Tags Registry ([I-D.ietf-netmod-module-tags], sec. 7.2). Module tag mappings can also be pre-installed by the server vendor, so no complex setup is required by an operator to use module tag filters.

The NETCONF and RESTCONF protocols do not have any way to select content by module name at all. In every case, either all content is
included or specific data node paths have to be provided by the client to include the associated data instances.

2.1.1. NETCONF "module-tag" RPC Input Parameter

The NETCONF protocol has 2 non-NMDA [RFC6241] retrieval operations (<get-config> and <get>) and 1 NMDA [RFC8526] retrieval operation (<get-data>). This document defines a "module-tag" grouping that is used an additional rpc input parameter for each operation. It can be applied to all server content that is accessible with these protocol operations.

2.1.2. RESTCONF "module-tag" Query Parameter

The RESTCONF protocol has a GET operation that allows query parameters to be provided to modify the retrieval operation. This document defines a new query parameter named "module-tag" that has the same semantics as the "module-tag" YANG data node definition. It also defines a RESTCONF Capability URN for a server to indicate that this query parameter is supported.

2.2. NACM "module-tags" Rule Type

The Network Configuration Access Control Protocol (NACM) [RFC8341] allows access control entries to apply to one module or all modules. This document defines a new "rule-type" case within a "rule" list entry, which allows the access control rule to apply to all the modules associated with one or more module tag values.

This new rule type is intended to replace the "module-name" parameter in the "rule" list entry. Module tag filters are intended to be used with the default "module-name" value "*" to represent all modules. This new "module-tags" rule imposes new restrictions on the rule selection, so it is backward compatible with the definitions in [RFC8341].

NACM rules using the "module-tag" parameter within the new "module-tags" case can be applied to configuration data, operational state, protocol operations and notification events. This rule type works exactly the same way as the "module-name" leaf within the "rule" list entry, except it applies to all associated module names, instead of one.

3. Definitions
3.1. YANG Module

This module imports definitions from [I-D.ietf-netmod-module-tags], [RFC6241], [RFC8341], and [RFC8526].

<CODE BEGINS> file "ietf-module-tag-ops@2019-03-10.yang"

module ietf-module-tag-ops {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-module-tag-ops";

  prefix mto;

  import ietf-module-tags { prefix tags; }
  import ietf-netconf { prefix nc; }
  import ietf-netconf-acm { prefix nacm; }
  import ietf-netconf-nmda { prefix ncds; }

  organization
    "IETF NETCONF (Network Configuration) Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/netconf/>
    WG List:  <mailto:netconf@ietf.org>
    Author:   Andy Bierman
    <mailto:andy@yumaworks.com>"

  description
    "This module defines enhancements to existing NETCONF
    operations for using module tags to represent
    YANG datastore content.

    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.

    Copyright (c) 2019 IETF Trust and the persons identified
    as authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4.c of the IETF Trust’s Legal Provisions
    Relating to IETF Documents
    (http://trusteee.ietf.org/license-info).";
grouping module-tag {
  description
  "Contains a reusable module-tag filter parameter";

  leaf-list module-tag {
    type tags:tag;
    description
    "Include only data nodes that match the module-tag value. A data node is matched to a module tag in the following manner:

    1) The module name associated with the data node is determined according to the protocol and message encoding.

    2) The module name is associated with the specified module-tag if a 'tag' entry exists within a /module-tags/module list entry with the same value as this entry, and a 'masked-tag' entry does not exist within the same /module-tags/module list entry.

    3) Each child data node is tested in recursive fashion. If the module name changes from the parent node, then this procedure is repeated. Once a module name does not match, then no further descendant nodes are included.

    Multiple module-tag parameters are combined as a logical OR expression. Matching any tag value will cause the data node to be included.

    It is not an error to include an unknown module-tag value. Such tag values will simply be treated as a 'false' match result, when evaluating the filter.

    If any module-tag parameters are provided at all, and there are no matches found, then no data will be
The output of all module-tag parameters are combined with other retrieval filters in a logical AND expression.

";
}
}

augment /ncds:get-data/ncds:input {
  description
  "Return data only if it matches according to the rules specified in the module-tag parameter.";
  uses module-tag {
    description
    "The module-tag values are applied starting at the top-level YANG data node within the target datastore.";
    reference
    "RFC 8526: NETCONF Extensions to Support the Network Management Datastore Architecture; Section 3.1.1";
  }
}

augment /nacm:nacm/nacm:rule-list/nacm:rule/nacm:rule-type {
  description
  "Match datastore content, protocol operations, or notification events only if the associated module name matches according to the rules specified in the module-tag parameter."

  If this rule type is used then the associated module-name parameter needs to be omitted or set to the default value. Otherwise it will interact with the module-tag parameter and the specified module-name will only apply if it is also included in the module-tag parameters provided.";

  case module-tags {
    uses module-tag {
      description
      "The module-tag values are applied to the conceptual document according to the NACM rules, starting at the top-level YANG data node. This is different in each access control enforcement procedure phase:
      
      - Incoming RPC Message Validation
        The module name of the association protocol operation is used to match a module-tag parameter.";

Bierman Expires September 11, 2019
- **Data Node Access Validation**
The module name associated with each data node within the target datastore, or within non-NMDA operational state (in an implementation-specific manner).

- **Outgoing <notification> Authorization:**
The module name of the association notification event is used to match a module-tag parameter.

```
";
reference
"RFC 8341: Network Configuration Access Control Model;
  Sections 3.2.4, 3.4.4, 3.5.5, 3.4.6";
```

```
augment /nc:get-config/nc:input {
  status deprecated;
  description
    "Return configuration data only if it matches according to the rules specified in the module-tag parameter.";
  uses module-tag {
    status deprecated;
    description
      "The module-tag values are applied starting at the top-level YANG data node within the target datastore.";
    reference
      "RFC 6241: Network Configuration Protocol; Section 7.1";
  }
}
```

```
augment /nc:get/nc:input {
  status deprecated;
  description
    "Return data only if it matches according to the rules specified in the module-tag parameter.";
  uses module-tag {
    status deprecated;
    description
      "The module-tag values are applied starting at the top-level YANG data node within the <running> datastore for configuration and the top-level YANG data nodes for all operational state data nodes.";
    reference
      "RFC 6241: Network Configuration Protocol; Section 7.7";
  }
}
```
3.2.  RESTCONF Query Parameter

The "module-tag" parameter can be used as a query parameter in a
RESTCONF protocol GET operation.  This new query parameter can be
applied to any data resource that is retrievable from the server.
This is done using the "(+restconf)/data" resource defined in section
3.3.1 of [RFC8040], or any datastore resource defined in section 3.1
of [RFC8527].

The retrieval filtering is processed exactly the same as the
"module-tag" parameter for NETCONF, defined in Section 3.1.

The module tag filtering starts with the top-level data nodes, the
same as for NETCONF.  All data nodes specified in the target resource
are subject to the same module-tag filter test as data nodes within
the target resource.

3.3.  RESTCONF Query Parameter Capability

The following RESTCONF Capability URI is defined to indicate that the
module-tag query parameter is supported by a RESTCONF server.  It
MUST be advertised as a "capability" in the /restconf-
state/capabilities/capability leaf-list.

The server MUST support the "module-tag" query parameter for GET and
HEAD methods if this capability is advertised.

4.  IANA Considerations

This document registers one URI as a namespace in the "IETF XML
Registry" [RFC3688]:

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

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

    name:         ietf-module-tag-ops
    prefix:       sx
    // RFC Ed.: replace XXXX with RFC number and remove this note
    reference:    RFC XXXX
This document registers one RESTCONF Capability URN in the registry defined in [RFC8040]:

name:        :module-tag
URN:         urn:ietf:params:restconf:capability:module-tag:1.0
// RFC Ed.: replace XXXX with RFC number and remove this note
reference:   RFC XXXX

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 [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 is one data node defined in this YANG module that is 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:

   /nacm/rule-list/rule/module-tag

This object allows an access control rule to be configured based on a module tag mapping. This object is vulnerable to modifications to the /module-tags configuration within the server. Care must be taken not to allow users to modify the /module-tags contents in a way that will expose protocol access in an unauthorized manner.

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:

   /nacm/rule-list/rule/module-tag
This object allows an access control rule to be configured based on a module tag mapping. Allowing read access to this object can expose the access control rule details.

6. References

6.1. Normative References

[I-D.ietf-netmod-module-tags]


6.2. Informative References


Appendix A. Examples

A.1. NETCONF <get-data> Example

This example uses the module tag value "ietf:hardware" which is defined in the YANG Module Tags registry. It is assumed in this case to be mapped on the server to the "ietf-hardware" module defined in [RFC8348].

Note that some lines are incorrectly wrapped in the examples below for display purposes only.
The server might send the following `<get-data>` request on the operational state datastore:

```xml
<rpc message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <datastore>ds:operational</datastore>
    <mto:module-tag
      ietf:hardware
    </mto:module-tag>
  </get-data>
</rpc>
```

The server might send the following reply:

```xml
<rpc-reply message-id="101"
            xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda">
    <hw:hardware xmlns:hw="urn:ietf:params:xml:ns:yang:ietf-hardware">
      <!-- rest of descendant nodes that match the module-tag -->
    </hw:hardware>
  </data>
</rpc-reply>
```

A.2.  RESTCONF GET Example

This example uses the same module-tag and server assumptions as Appendix A.1.

The client might send the following request:

GET /restconf/ds/ietf-datastores:operational?module-tag=ietf:hardware
HTTP/1.1
Host: example.com
Accept: application/yang-data+json

The server might respond as follows. The contents of the "hardware" container are omitted for brevity.
A.3. NACM Example

In this example, a module tag rule is created to deny guests all access to hardware information.

Note that some lines are incorrectly wrapped in the example below for display purposes only.

```xml
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <rule-list>
    <name>guest-acl</name>
    <group>guest</group>
    <rule>
      <name>deny-hw</name>
      <mto:module-tag
        ietf:hardware
      </mto:module-tag>
      <access-operations>*</access-operations>
      <action>deny</action>
      <comment>
        Do not allow guests to access any hardware information
      </comment>
    </rule>
  </rule-list>
</nacm>
```

Author’s Address

Andy Bierman
YumaWorks

Email: andy@yumaworks.com