YANG Extension and Metadata Annotation for Immutable Flag

draft-ma-netmod-immutable-flag-05

Qiufang Ma (Huawei) Presenter

Qin Wu (Huawei)

Balazs Lengyel (Ericsson)

Hongwei Li (HPE)

Motivation

- Ideally any configuration data node is allowed to be created, updated and deleted

- In the real world, there exists data that cannot be modified by the client but still needs to be declared as “config true” in order to:
  - allow configuration of descendant data nodes of immutable list/containers
  - place “when”, “must” and “leafref” constraints between real config and immutable nodes
  - ensure existence of specific system-created entries, while additional entries may be created, updated and deleted

- If the server knows that it will always reject the configuration it should inform the client early (e.g., possibly in implementation time) thus avoiding failed operations

Immutability - Existing allowed behavior

- The server is allowed to reject configuration for reasons not documented in the YANG model
  - e.g., because it internally considers a data node immutable
- The behavior is already allowed today, not new
- A number of SDOs and vendors already implement this immutability concept (see next slide)
- The goal is to define one single IETF solution instead of the multiple existing vendor and organization specific solutions.
  - interoperability

Users of Immutability and YANG

- 3GPP uses both above concepts and has already defined a YANG extension for isInvariant.
- ORAN re-uses the 3GPP concepts
- ITU-T uses the invariant concept
- Ericsson has a similar extensions defined
- YumaPro has a similar extension defined: yuma-ncx:user-write.
- Nokia has a similar extension defined sros-ext: immutable
- Huawei also has the very similar extension defined: ext:operation-exclude
- Cisco has isInvariant=true data nodes at least in some of its YANG models

Use Cases

- HW-based configuration, e.g., system defined interface name and type
- Server capabilities that need to be leafref-ed by other configuration
- System-defined access control groups and rule-list entries
- Declaring (some) system defined configuration unchangeable
- Modeling existing data handling behavior in other standard organizations
  - Already use concept similar to immutability
  - Some are introducing YANG modeling
  - Options: document immutability in description VS. new formal statement
  - See recent 3GPP liaison statement

Document Updates Since IETF 115

- Reword the abstract and introduction sections
  - emphasize that the proposal tries to formally document existing allowed behavior
  - aim to create one single standard solution
  - Immutable behavior is allowed but discouraged
- Editorial and refactoring updates regarding solution part, but no substantial changes
- Add a temporary section in Appendix B about the usage of immutable concept among multiple vendors and organizations
- Rewrite the use cases in Appendix A
  - Try to cover the most common and reasonable cases

Why current mechanisms are insufficient?

- NACM
  - Can be disabled by setting the “enable-nacm” leaf to “false”
  - Can be bypassed by emergency recovery session
  - Can be bypassed by adding a new allow all rule
  - NACM rules are also immutable and no way to protect them from being modified
- YANG Deviation
  - Deviation cannot help modify immutability of a particular node
  - It cannot allow constraints between config false and true nodes
  - It cannot allow configuration of descendant data nodes to immutable lists or containers
  - It cannot protect some list entries while allowing modification of others

An alternative that models similar concepts in 3GPP

- isInvariant
  - An attribute is assigned a value when the object is created, but the value cannot be modified later; a only way to modify it is to delete it and re-create with the same attribute with a different value
  - Could be represented in YANG using “immutable ‘create delete’”
- systemCreated
  - Instances cannot be created or deleted by the client
  - Could be represented in YANG using “immutable ‘update’”

Next Step

- The authors would like to ask for adoption by the working group
- Comments, questions, concerns?

Solution Overview (no update)

- Editorial and refactor updates since last meeting, but no substantial changes
- Still open and could have further discussion

immutable YANG extension

```
extension immutable {
   argument exceptions;
   …
}
```

+

immutable Metadata Annotation

```
md:annotation immutable {
    type boolean;
    …
}
```

- “exceptions” is also defined to indicate specific operations (create, update, delete) are permitted
- e.g., if a configuration data can only be created and deleted, while modification is not allowed:

```
im:immutable “create delete”;
```

- Indicates that once a particular instantiated data node is created, the client cannot update/delete it
- Be applied to list/leaf-list entries or instances inside particular list entries