I2NSF YANG Data Models

draft-ietf-i2nsf-capability-data-model-05
draft-ietf-i2nsf-consumer-facing-interface-dm-06
draft-ietf-i2nsf-nsf-facing-interface-dm-07
draft-ietf-i2nsf-registration-interface-dm-05
draft-ietf-i2nsf-nsf-monitoring-data-model-01

IETF 105, Montreal
July 25, 2019

Jaehoon Paul Jeong
pauljeong@skku.edu
Sungkyunkwan University

WG Documents of YANG Data Models

- Information Model Draft on NSF Capabilities
  - draft-ietf-i2nsf-capability-05
- Base YANG Data Model Draft
  - draft-ietf-i2nsf-capability-data-model-05
- I2NSF Interface YANG Data Model Drafts
  - draft-ietf-i2nsf-consumer-facing-interface-dm-06
  - draft-ietf-i2nsf-nsf-facing-interface-dm-07
  - draft-ietf-i2nsf-registration-interface-dm-05
  - draft-ietf-i2nsf-nsf-monitoring-data-model-01
- Verification of those YANG Data Models
  - Those will be verified through the **9 IETF Hackathons** (IETF 97 ~ IETF 105).
  - **4 Awards among 9 Hackathons**

Updates from the Previous Versions

- Consistency with **NSF Capabilities Information Model**
  - draft-ietf-i2nsf-capability-05

- We have addressed the comments from YANG doctors to the Data Model (DM) drafts and submitted the revised drafts:
  - NSF Capability DM
  - Consumer-Facing Interface DM
  - Registration Interface DM
  - NSF-Facing Interface DM
    - Two comments will be reflected in the next revision.

Updates of Capability Data Model (DM) (1/2)

- Consistency with NSF Capabilities Information Model
  - draft-ietf-i2nsf-capability-05
- Relationship with Other YANG Data Models
  - draft-ietf-i2nsf-consumer-facing-interface-dm-06
  - draft-ietf-i2nsf-nsf-facing-interface-dm-07
  - draft-ietf-i2nsf-registration-interface-dm-05
- Revision from YANG doctors' comments
  - Refer to Appendix for more detailed revision

Updates of Capability Data Model (DM) (2/2)

- Major Comment
  - The "Security Considerations" in section 8
    - not conform to the recommended template; https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines

- Changed to
  - **The attacker may provide incorrect information of the security capability of any target NSF by illegally modifying this.**
  - **The attacker may gather the security capability information of any target NSF and misuse the information for subsequent attacks.**

Updates of NSF-Facing Interface DM (1/2)

- Date and time are used, which are defined in RFC 6991 rather than new definitions are used.
- "time-intervals" are used to represent intervals rather than "time-zones".
- "acl-number" is deleted because it is not used.
- The descriptions are improved according to the reviewer's suggestions.
- The "Security Considerations" in Section 8 conform to the recommended template in https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines>.

Updates of NSF-Facing Interface DM (2/2)

- The leveraging of the definitions in RFC 8519 for packet matching.
  - Due to time limitation, this will be reflected in the next revision.

- The factoring of common types and identities into a common I2NSF types module.
  - Due to time limitation, this will be reflected in the next revision as well.

Updates of Consumer-Facing Interface DM

- Revision from a YANG doctor's comments
  - Refer to Appendix for more detailed revision

- Major Comments
  - Management access control
    - The container policy-mgnt-auth-method is now a list.

- Changed to
  - '**list policy-mgmt-auth-method-instance**' instead of 'container policy-mgnt-auth-method'

Updates of Registration Interface DM (1/2)

- Revision from a YANG doctor's comments
  - Refer to Appendix for more detailed revision
- Revision of YANG Module structure according to RFC 8407 Appendix B
- Addition of detailed description of each component of the YANG module
- Changed the prefix with "nsfreg"

Updates of Registration Interface DM (2/2)

- Modified nsf-address to deal with both IPv4 and IPv6 addresses
- Revised all examples to use IPv6 address specified in RFC 3849 in Appendix A
- “nsf-port-address” has been changed into “nsf-port”.
- Revised security considerations section, and added more explanation to Section 4

Updates of NSF Monitoring DM

- YANG Data Model (DM) corresponding to the Information Model (IM) for NSF-Facing Interface:
  - draft-ietf-i2nsf-capability-05
- This data model is derived from capability data model:
  - draft-ietf-i2nsf-capability-data-model-05
- Changed Requirement Notation, and added references RFC 8329, RFC 8342, and RFC 6020

Next Steps

- IESG Submission for NSF Capability DM Draft
  - After IETF-105 Meeting
- WG Last Call after IETF-105 Meeting
  - NSF-Facing Interface DM
  - Consumer-Facing Interface DM
  - Registration Interface DM
- NSF Monitoring Data Model Draft
  - We are planning to test it in IETF-106 Hackathon
  - WGLC in IETF-106 Meeting

Appendix

- **Capability DM**
  (Reviewers: Acee Lindem and Carl Moberg)
- **NSF-Facing Interface DM**
  (Reviewer: Acee Lindem)
- **Consumer-Facing Interface DM**
  (Reviewer: Jan Lindblad)
- **Registration Interface DM**
  (Reviewer: Reshad Rahman)

YANG DOC's Revision of Capability DM

- Comments: The "Security Considerations" in section 8 do not conform to the recommended template in 'https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines'.

- OLD: section 8 (Last version)

- NEW: added following sentences in section 8
  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:
  - ietf-i2nsf-capability: The attacker may provide incorrect information of the security capability of any target NSF by illegally modifying this.

  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:
  - ietf-i2nsf-capability: The attacker may gather the security capability information of any target NSF and misuse the information for subsequent attacks.

YANG DOC’s Revision of NSF-Facing Interface DM

- Comments: Why don't you leverage the definitions in RFC 8519 for packet matching? We don't need all this defined again?

- Answer: Due to the time limitation, this revision cannot reflect the usage of definitions In RFC 8519 for packet matching. I will reflect your comments on the next revision.

YANG DOC's Revision of NSF-Facing Interface DM

- Comments: Date and time are defined in RFC 6991. Why don't those suffice?

OLD

```
typedef start-time-type {
    type union {
      type string {
        pattern '₩d{2}:₩d{2}:₩d{2}(₩.₩d+)?'
              + '(Z|[₩+₩-]₩d{2}:₩d{2})';
      }

      type enumeration {
        enum right-away {
          description
            "Immediate rule execution
             in the system.";
        }
      }
 }
```

NEW

```
typedef start-time-type {
    type union {
      type ynag:date-and-time;

      type enumeration {
        enum right-away {
          description
            "Immediate rule execution
             in the system.";
        }
      }
    }
  }
```

YANG DOC's Revision of NSF-Facing Interface DM

- Comments: Refer to the intervals as "time-intervals" rather than "time-zones". The term "time-zone" has a completely different connotation.

OLD

```
|     |  +--rw time-zones
|     |  |  +--rw absolute-time-zone
|     |  |  |  +--rw start-time?    start-time-type
|     |  |  |  +--rw end-time?      end-time-type
|     |  |  +--rw periodic-time-zone
|     |  |     +--rw day
|     |  |     |  +--rw every-day?       boolean
|     |  |     |  +--rw specific-day*    day-type
|     |  |     +--rw month
|     |  |        +--rw every-month?       boolean
|     |  |        +--rw specific-month*    month-type
```

NEW

```
|     |  +--rw time-intervals
|     |  |  +--rw absolute-time-interval
|     |  |  |  +--rw start-time?    start-time-type
|     |  |  |  +--rw end-time?      end-time-type
|     |  |  +--rw periodic-time-interval
|     |  |     +--rw day
|     |  |     |  +--rw every-day?       boolean
|     |  |     |  +--rw specific-day*    day-type
|     |  |     +--rw month
|     |  |        +--rw every-month?       boolean
|     |  |        +--rw specific-month*    month-type
```

YANG DOC’s Revision of NSF-Facing Interface DM

- Comments: What the "acl-number"? Also, ACLs are named (RFC 8519). Also, why define all the packet matching and then reference an ACL.

- Answer: We delete acl-number because this is not used. And, in that case of packet matching, due to the time limitation, this revision cannot reflect the usage of definitions In RFC 8519 for packet matching. On the next revision, I will reflect your comment.

YANG DOC’s Revision of NSF-Facing Interface DM

- Comments: The descriptions are very awkwardly worded and in many cases simply repeat the data node or identify description without hyphens. I started trying to fix this but it was too much. I'll pass for on for some examples. There are enough co-authors and contributors that one would expect much better.

- Answer: I reflected the sentences that you revised on the revision.

YANG DOC's Revision of NSF-Facing Interface DM

- Comments: There is overlap of definitions with the I2NSF capabilities draft. The common types and identities should be factored into a common I2NSF types module.

- Answer: Due to the time limitation, this revision cannot reflect the factoring of the common types and identities. I will reflect your comments on the next revision.

YANG DOC's Revision of NSF-Facing Interface DM

- Comments: The "Security Considerations" in section 8 do not conform to the recommended template in https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines>

OLD
The YANG module specified in this document defines a data schema designed to be accessed through network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the required transport secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the required transport secure transport is TLS [RFC8446]. The NETCONF access control model [RFC8341] provides a means of restricting access to specific NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.
NEW
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: ietf-i2nsf-policy-rule-for-nsf: The attacker may provide incorrect policy information of any target NSFs by illegally modifying this. 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: ietf-i2nsf-policy-rule-for-nsf: The attacker may gather the security policy information of any target NSFs and misuse the security policy information for subsequent attacks.

YANG DOC's Revision of Consumer-Facing Interface DM

- Comments: Management access control
  - container policy-mgnt-auth-method should probably be a list.

OLD: container policy–mgnt–auth–method

```
container policy-mgnt-auth-method {
    description "This represents the list of authentication methods."; leaf auth-method {
        type string;
        description "This represents the authentication method name.";
    }
...
}
```

NEW: list policy-mgmt-auth-method-instance

```
list policy-mgmt-auth-method-instance {
    ....
    choice policy-mgmt-auth-method {
        ....
        case password-based {...}
        case token-based {...}
        case certificate-based {...}
```

YANG DOC's Revision of Consumer-Facing Interface DM

- Comments: Management access control
  - container policy-role should probably be a list, and the list access-profile removed.

OLD: poliy role

```
container policy-role {
    uses meta;
    description
    list access-profile {
      uses meta;
      key "name";

      leaf permission-type {
      type identityref {
      base permission-type;
      }
     default read-only;
     }
```

NEW: list policy-mgmt-auth-method-instance

```
The list policy role:
    list policy-role {
        key "role-type";
        leaf role-type {
           type identityref {
              base role-type;
           }
           description
          "This represents the role";
          }
```

```
The role-type identities:

 identity role-type {
      description
      "This is the base identity for the roles.";
  }
  identity user {
      base role-type;
      description
      "This represents the identity of the user
role.";
```

YANG DOC's Revision of Consumer-Facing Interface DM

- Comments: Management access control
  - the leaf owner description talks about owner of the policy, while the leaf sits on an individual rule. Either the description or the leaf placement must be wrong.

OLD: leaf owner

```
leaf owner {
    type string;
    description
    "This field defines the owner of this policy. Only the owner is authorized to modify the
contents of the policy.";
  }
```

NEW: new leaf owner

```
leaf owner {
   type identityref {
      base owner;
   }
   mandatory true;
   description
    "This field defines the owner of this
    rule. Only the owner is authorized to
    modify the contents of the rule.";
}
```

```
identity owner {
   description
   "This is the base identity for the owner";
}
identity dept-head {
   base owner;
   description
   "This represents the identity of the head of
department.";
}
```

YANG DOC's Revision of Consumer-Facing Interface DM

- Comments: container policy
  - container policy needs to change to list policy (and probably add a surrounding container policies).

OLD: container policy	NEW: list policy

container policy {
  leaf polict-name {...}
  list rule {...}
}
	list i2nsf-cfi-policy {
	  key "policy-name";
	  leaf policy-name {
	  type string;
	  mandatory true;
	  }
	  list rule {...}
	}

YANG DOC's Revision of Consumer-Facing Interface DM

- Comments: Many strings (1/2)
  - In the module, the type string is used for names, which is great, but also for a many cases where some certain format of the content is expected, but not defined. There is no reason to believe that will lead to interoperable solutions.

- OLD & NEW
  - Next slide will be continued…

YANG DOC's Revision of Consumer-Facing interface DM

- Comments: Many strings (2/2)

OLD
294: leaf-list 322: leaf-list content 388: leaf begin-time 393: leaf end-time 553: leaf primary-action 561: leaf secondary-action 581: leaf owner 697: leaf auth-method 823: leaf threat-feed-description 839: leaf-list signatures
NEW: fixed types.
294: leaf-list protocol ==> protocol type is added (FTP, SSH, HTTP, HTTPS, etc.) 322: leaf-list content ==> this is for the admin (or an entity who is creating or modifying the rule with such content) to copy & paste the payload content. It is meant to be string. 388: leaf begin-time ==> This type is replaced with yang:date-and-time 393: leaf end-time ==> This type is replaced with yang:date-and-time 553: leaf primary-action ==> It is no longer a string type; primary-action identities are added. 561: leaf secondary-action ==> It is no longer a string type; secondary-action identities are added. 581: leaf owner ==> identities for owner is now added. The type is now identitref not string. 697: leaf auth-method ==> leaf auth-method is removed. Instead, auth-intance-type is added with auth-type (authentication type) identities. Its type is identityref now. 823: leaf threat-feed-description ==> this is for the admin (or an entity who is creating or modifying the rule related to threat feeds) to describe what information is obtained from a treat feed. It is meant to be string. 839: leaf-list signatures ==> The signatures are like track of a security threats. They are usually a bunch of strings (or binary codes), and used to generate a security rule as part of its contents. Therefore, the type for the entries in the leaf-list signature should remain as strings.

YANG DOC's Revision of Consumer-Facing Interface DM

- Comments: container policy-mgnt-auth-method (1/2)
  - A container is obviously not a list, so exactly what the author has in mind is somewhat unclear. At the top of the container, there is a leaf

```
leaf auth-method {
  type string;
  description
      "This represents the authentication method name.";
```

- OLD & NEW
  - Next slide will be continued…

YANG DOC's Revision of Consumer-Facing Interface DM

- Comments: container policy-mgnt-auth-method (2/2)

OLD

```
container policy-mgnt-auth-method {
        leaf auth-method {...}
        leaf mutual-authentication {...}
        list password-based {
           key "password";
           leaf password {...}
        }
        list token-based {
           key "token";
           ...
        ...
```

NEW: The authentication methods are now choices. If this model needs to be extended for new authentication methods, simply create an identity, grouping, and case for the new method.

```
    list policy-mgmt-auth-method-instance {
       key "auth-instance-type";
       description
       "This represents the list of instances for
       policy management authentication methods.";

       leaf auth-instance-type {
          type identityref {
             base auth-type;
          }
          description
          "This identifies whether the authentication type
          is server authentication, client authentication,
          or both.";
       }
       choice policy-mgmt-auth-method {
          description
          "This represents the choices for which
          authentication method is used.";
          case password-based {
          uses password-based-method;
          }
          case token-based {
             description
             "This represents the token-based method.";
             uses token-based-method;
          }
          case certificate-based {
             description
             "This represents the certificate-based-method.";
             uses certificate-based-method;
          }
          case ipsec {
             description
             "This repreents authentication method based on IPSEC.";
             uses ipsec-method;
          }
       }
    }
```

YANG DOC's Revision of Consumer-facing interface DM

- Comments: Enumerations vs. identities
  - 203, 366: enum -> identity
  - 55, 168: identity -> enum
  - 145: misspelled of ransomware

OLD	NEW
203: certificate-type: enumeration	203: certificate-type: identity (cer, crt, key...)
366: enforce-type: enumeration	366: enforce-type: identity (admin, time, event)
55: permission-type: identity	55: permission-type: identity (read, write, execute, read-and-write, read-and-execute, write-and-execute, no-permission)
168: continent: identity	168: continent: identity
145: ransomeware	145: ransomware

YANG DOC's Revision of Registration Interface DM

- Comments: Look at appendix B of RFC8407 for an example of how a YANG module should be structured. This document does not abide to that.
  - modified the structure of the YANG module to abide to the template described in appendix B of RFC8407.

- Comments: Poor descriptions e.g."nsf-name" for leaf "nsf-name" etc
  - added a detailed description of each component of the YANG module.

```
container i2nsf-nsf-registrations {
 description
  "Information of an NSF that DMS registers to Security Controller";
 list i2nsf-nsf-capability-registration {
  key "nsf-name";
  description
   "Required information for registration";
  leaf nsf-name {
```

YANG DOC's Revision of Registration Interface DM

- Comments: prefix "iiregi" doesn't seem right. What about "nsfreg"? Probably needs coordination with the other I2NSF YANG modules to have consistency between the prefixes.
  - changed the prefix with "nsfreg."

```
module ietf-i2nsf-reg-interface {
 yang-version 1.1;

 namespace "urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface";
 prefix nsfreg;
```

YANG DOC's Revision of Registration Interface DM

- Comments: nsf-address is IPv4 specific (1/2)
  - revised nsf-address so that it can deal with both IPv4 and IPv6 as follows

- OLD & NEW
  - Next slide will be continued…

YANG DOC's Revision of Registration Interface DM

- Comments: nsf-address is IPv4 specific (2/2)

[OLD]:

```
NSF Access Information
  +--rw i2nsf-nsf-access-info
    +--rw nsf-instance-name      string
    +--rw nsf-address            inet:ipv4-address
    +--rw nsf-port-number        inet:port-number
```

Figure 10: YANG tree of I2NSF NSF Access Informantion

```
      }
      leaf nsf-address {
        type inet:ipv4-address;
        mandatory true;
        description
          "nsf-address";
      }
```

[NEW]:

```
NSF Access Information
  +--rw i2nsf-nsf-access-info
    +--rw nsf-instance-name      string
    +--rw i2nsf-nsf-address
      +-- rw nsf-ipv4-address      inet:ipv4-address
      +-- rw nsf-ipv6-address      inet:ipv6-address
    +--rw nsf-port-number        inet:port-number
```

Figure 10: YANG tree of I2NSF NSF Access Informantion

```
typedef nsf-address {
  leaf nsf-ipv4-address{
    type inet:ipv4-address
    description
    "ipv4-address"
  }
  leaf nsf-ipv6-address {
    type inet:ipv6-address
    description
    "ipv6-address"
  }
}
```

```
    container i2nsf-nsf-address {
      uses nsf-address
      description
        "ipv4 and ipv6";
    }
```

YANG DOC's Revision of Registration Interface DM

- Comments: Examples should use IPv6 as examples (use the range from RFC3849).
  - In Appendix A, we revised all the examples to use the IPv6 address specified in RFC3849.

```
<i2nsf-nsf-address>
  <nsf-ipv6-address>2001:DB8:8:4::2</nsf-ipv6-address>
</i2nsf-nsf-address>
```

- Comments: nsf-port-address should be nsf-port?
  - "nsf-port-address" has been changed into "nsf-port."

```
leaf nsf-port {
 type inet:port-number;
 description
  "Port available on this NSF";
}
```