I2NSF YANG Data Models

draft-ietf-i2nsf-capability-data-model-02
draft-ietf-i2nsf-consumer-facing-interface-dm-02
draft-ietf-i2nsf-nsf-facing-interface-dm-02
draft-ietf-i2nsf-registration-interface-dm-01

IETF 103, Bangkok
November 7, 2018

Jaehoon Paul Jeong
pauljeong@skku.edu
Sungkyunkwan University
WG Documents of YANG Data Models

• Information Model Draft on NSF Capabilities
  • draft-ietf-i2nsf-capabilities-04

• Base YANG Data Model Draft
  • draft-ietf-i2nsf-capability-data-model-02

• I2NSF Interface YANG Data Model Drafts
  • draft-ietf-i2nsf-consumer-facing-interface-dm-02
  • draft-ietf-i2nsf-nsf-facing-interface-dm-02
  • draft-ietf-i2nsf-registration-interface-dm-01

• Verification of those YANG Data Models
  • Those were verified through the 7 IETF Hackathons (IETF 97 ~ IETF 103).
I2NSF Capability YANG Data Model
(draft-ietf-i2nsf-capability-data-model-02)

IETF 103, Bangkok
November 7, 2018

Susan Hares, Jaehoon Paul Jeong, Jinyong (Tim) Kim,
Robert Moskowitz, and Qiushi Lin
Updates from the Previous Version

• Consistency with **NSF Capabilities Information Model**
  • draft-ietf-i2nsf-capabilities-04.

• Capabilities of **Advanced Network Security Functions**
  • Anti-Virus
  • Anti-DDoS
  • IPS

• **Accommodation for Advanced NSFs Capabilities**
  • draft-dong-i2nsf-asf-config-01

• **Relationship with Other YANG Data Models**
  • The further YANG data models can be used as **YANG sub-modules** for this Base YANG data model.
Capabilities of Advanced NSFs (1/2)

```yaml
module: ietf-i2nsf-capability
  +--rw nsf* [nsf-name]
    +--rw nsf-name          string
    +--rw nsf-type?         nsf-type
    +--rw nsf-address
        | +--rw (nsf-address-type)?
        |   +--rw ipv4-address
        |     | +--rw ipv4-address     inet:ipv4-address
        |     +--rw ipv6-address
        |       +--rw ipv6-address     inet:ipv6-address
    +--rw target-device
        |    +--rw pc?           boolean
        |   +--rw mobile-phone?  boolean
        |   +--rw voip-volte-phone?  boolean
        |   +--rw tablet?        boolean
        |   +--rw iot?           boolean
        |   +--rw vehicle?       boolean
    +--rw generic-nsf-capabilities
        | +--rw nct-sec-capabilities
        |   uses nct-sec-caps
        | +--rw advanced-nsf-capabilities
        |   +--rw advanced-sec-capabilities
        |     uses advanced-sec-caps
    +--rw complete-nsf-capabilities
        | +--rw con-sec-control-capabilities
        |   uses i2nsf-con-sec-control-caps
        | +--rw attack-mitigation-capabilities
        |   uses i2nsf-attack-mitigation-control-caps
```
Capabilities of Advanced NSFs (2/2)

---rw advanced-nsf-capabilities
  |   ---rw advanced-sec-capabilities
  |     |   ---rw antivirus
  |     |     |   ---rw detect? boolean
  |     |     |   ---rw exception-application? boolean
  |     |     |   ---rw exception-signature? boolean
  |     |     |   ---rw whitelists? boolean
  |     |   ---rw antiddos
  |     |     |   ---rw syn-flood-action? boolean
  |     |     |   ---rw udp-flood-action? boolean
  |     |     |   ---rw http-flood-action? boolean
  |     |     |   ---rw https-flood-action? boolean
  |     |     |   ---rw dns-request-flood-action? boolean
  |     |     |   ---rw dns-reply-flood-action? boolean
  |     |     |   ---rw icmp-flood-action? boolean
  |     |     |   ---rw sip-flood-action? boolean
  |     |     |   ---rw detect-mode? boolean
  |     |     |   ---rw baseline-learn? boolean
  |     |   ---rw ips
  |     |     |   ---rw signature-set? boolean
  |     |     |   ---rw exception-signature? boolean
Next Steps

• We will change the current YANG data model to the YANG data model of Object-Oriented Style, such as Decorator patterns.
  • Huawei (e.g., Liang (Frank) Xia) will provide us with a sample YANG data model using Decorator patterns.

• After the proofreading by the authors of the NSF Capabilities Information Model document, we will correct the data model and finalize it.

• WG Last Call with this December.
Network Security Functions Facing Interface YANG Data Model
(draft-ietf-i2nsf-nsf-facing-interface-dm-02)

IETF 103, Bangkok
November 7, 2018

Jinyong (Tim) Kim, Jaehoon Paul Jeong, Jung-Soo Park, Susan Hares,
and Qiushi Lin
Updates from the Previous Version

• Consistency with NSF Capabilities Information Model
  • draft-ietf-i2nsf-capabilities-04.

• Liang (Frank) Xia’s Comments
  • Add System Policy for multiple system policies in one NSF (Resolved)
  • Delete agg.ptr attributes due to uncleanness (Resolved)
    • policy-event-clause-agg.ptr*
    • policy-condition-clause-agg.ptr*
    • policy-action-clause-agg.ptr*
  • Add policy-usage-type for rule order with priority criteria (Resolved)
    • priority-by-order
    • priority-by-number
System Policy for multiple system policies in one NSF

OLD:

module: ietf-i2nsf-policy-rule-for-nsf
+++rw i2nsf-security-policy
| +++rw policy-name? string
| +++rw rules* [rule-name]
| | +++rw rule-name string
| | +++rw rule-description? string
| | +++rw rule-priority? uint8
| | +++rw enable? boolean
| | +++rw session-aging-time? uint16
| | +++rw long-connection
| | | +++rw enable? boolean
| | | +++rw during? uint16
| | +++rw policy-event-clause-agg-ptr* instance-identifier
| | +++rw policy-condition-clause-agg-pcr* instance-identifier
| | +++rw policy-action-clause-agg-ptr* instance-identifier

NEW:

module: ietf-i2nsf-policy-rule-for-nsf
+++rw i2nsf-security-policy
| +++rw system-policy* [system-policy-name]
| | +++rw system-policy-name string
| | +++rw priority-usage priority-usage-type
| | +++rw rules* [rule-name]
| | | +++rw rule-name string
| | | +++rw rule-description? string
| | | +++rw rule-priority? uint8
| | | +++rw enable? boolean
| | | +++rw session-aging-time? uint16
| | | +++rw long-connection
| | | | +++rw enable? boolean
| | | | +++rw during? uint16
| | | +++rw policy-event-clause-agg-ptr* instance-identifier
| | | +++rw policy-condition-clause-agg-pcr* instance-identifier
| | | +++rw policy-action-clause-agg-ptr* instance-identifier
Delete agg-ptr attributes

OLD:

module: ietf-12nsf-policy-rule-for-nsf
  +--rw i2nsf-security-policy
      |  | +--rw policy-name?  string
      |  | | +--rw rules* [rule-name]
      |  | | | +--rw rule-name  string
      |  | | | | +--rw rule-description?  string
      |  | | | | | +--rw rule-priority?  uint8
      |  | | | | | | +--rw session-aging-time?  uint16
      |  | | | | | | | +--rw enable?  boolean
      |  | | | | | | | | +--rw during?  uint16
      |  | | | | | | | | | +--rw policy-event-clause-agg-ptr*  instance-identifier
      |  | | | | | | | | | | +--rw policy-condition-clause-agg-ptr*  instance-identifier
      |  | | | | | | | | | | | +--rw policy-action-clause-agg-ptr*  instance-identifier

NEW:

module: ietf-12nsf-policy-rule-for-nsf
  +--rw i2nsf-security-policy
      |  | +--rw system-policy* [system-policy-name]
      |  | | +--rw system-policy-name  string
      |  | | | +--rw priority-usage [priority-usage-type]
      |  | | | | +--rw rule-name  string
      |  | | | | | +--rw rule-description?  string
      |  | | | | | | +--rw rule-priority?  uint8
      |  | | | | | | | +--rw enable?  boolean
      |  | | | | | | | | +--rw session-aging-time?  uint16
      |  | | | | | | | | | +--rw long-connection  boolean
      |  | | | | | | | | | | +--rw enable?  boolean
      |  | | | | | | | | | | | +--rw during?  uint16
      |  | | | | | | | | | | | | +--rw policy-event-clause-agg-ptr*  instance-identifier
      |  | | | | | | | | | | | | +--rw policy-condition-clause-agg-ptr*  instance-identifier
      |  | | | | | | | | | | | | | +--rw policy-action-clause-agg-ptr*  instance-identifier
      |  | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw event-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw condition-clause-container  boolean
      |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | +--rw action-clause-container  boolean
Add policy-usage-type

NEW:

module: ietf-i2nsf-policy-rule-for-nsf
+++rw i2nsf-security-policy
+++rw system-policy* [system-policy-name]
+++rw system-policy-name string
+++rw priority-usage priority-usage-type
+++rw rules* [rule-name]
  | +++rw rule-name string
  | +++rw rule-description? string
  | +++rw rule-priority? uint8
  | +++rw enable? boolean
  | +++rw session-aging-time? uint16
  | +++rw long-connection
    | | +++rw enable? boolean
    | | +++rw during? uint16

typedef priority-usage-type {
    type enumeration {
        enum priority-by-order {
            description
                "If priority type is order";
        }
        enum priority-by-number {
            description
                "If priority type is number";
        }
    }
    description
        "This is used for priority type.";
}
Next Steps

• We will add condition clause operator types such as exact-match, range-based, regex-based, and custom-match.

• We will change the data structure to accommodate other YANG data models, such as advanced NSFs (draft-dong-i2nsf-asf-config-01).

• We will change the current YANG data model to the YANG data model of Object-Oriented Style, such as Decorator patterns.

• After the proofreading by the authors of the NSF Capabilities Information Model document, we will correct the data model and finalize it.

• WG Last Call with this December.
Consumer-Facing Interface Data Model
(draft-ietf-i2nsf-consumer-facing-interface-dm-02)

IETF 103, Bangkok
November 7, 2018

Jaehoon (Paul) Jeong, Eunsoo Kim, Tae-Jin Ahn,
Rakesh Kumar, and Susan Hares
Updates from the Previous Version

• The following changes are made from:
  • draft-ietf-i2nsf-consumer-facing-interface-dm-01

• Major Changes:
  • Included the sources for the threat-feed, such as STIX & TAXII for Structured Threat Information Expression and Relay.
  • Included descriptions on how those sources can be used.

• Minor Changes:
  • Correction of editorial mistakes (spelling, grammatical errors, etc.)
Major Changes

- Modified threat-feed container so that it can support various threat related sources (e.g., STIX and IOC)

STIX: Structured Threat Information Expression
IOC: Indicator of Compromise

Threat-feed type:
Identify the type of the threat-feed

Threat-feed source address

The priority of the threat-feed: 0 ~ 5
Next Steps

• We will improve the current YANG data model to the YANG data model of Object-Oriented Style, such as Decorator patterns.
  • Huawei (e.g., Liang (Frank) Xia) will provide us with a sample YANG data model using Decorator patterns.

• After the proofreading by the authors of the NSF Capabilities Information Models document, we will correct the data model and finalize it.

• WG Last Call with this December.
I2NSF Registration Interface Data Model
(draft-ietf-i2nsf-registration-interface-dm-01)

IETF 103, Bangkok
November 7, 2018

Sangwon Hyun, Jaehoon (Paul) Jeong,
Taekyun Roh, Sarang Wi and Jungsoo Park
Updates from the Previous Version

• The Previous Draft:
  - draft-ietf-i2nsf-registration-interface-dm-00

• Changes from the Previous Version
  - The description of the operations performed over the Registration Interface has been revised with
    • register-select-instantiate operation sequence.
  - We revised Section 4 of the objectives of the registration interface to match the register-select-instantiate operation sequence.
  - The appendix has been added to clarify the Lifecycle Management of NSFs in I2NSF framework based on NFV.
NFV Reference Architecture for I2NSF

Architecture
- OS: Ubuntu 16.04
- NFV Infra: OpenStack Rocky release
- VNFM: OpenStack Tacker project
- Network: OpenStack Networking SFC

Interface
- Consumer-Facing Interface
- Registration Interface
- NSF-Facing Interface
- Ve-VNFm interface (RESTAPI)

Data Model
- VNFD: TOSCA Template
- VNFFG: TOSCA Template
- Data Modeling: Netconf YANG
I2NSF Hackathon Project Implementation: Registration Interface in I2NSF with NFV

Scenario 1: NSF Available

• Case 1: NSF Activated
  • High-level policy request
    - Translation: Data Conversion
    - Case 1: NSFs Available (Activated)
    - Translation: Policy Generation
  • Low-level policy request
  • Low-level policy response
  • High-level policy response

• Case 2: NSF De-activated
  • High-level policy request
    - Translation: Data Conversion
    - Case 2: NSFs Available (De-activated)
    - NSF initiation Request (Registration Interface)
    - NSF initiation Response (Ve-Vnfm)
    - NSF initiation

  • Low-level policy request
  • Low-level policy response

High-level policy request
Low-level policy request
High-level policy response
Low-level policy response
Scenario 2: NSF Unavailable

- **Case 1: Capability not searched**
  - High-level policy request
  - Translation: Data Conversion
  - Profile entry not matched
  - Capability Query
  - Search
  - Case 1: Capability not Searched
  - No-NSF-found Reply
  - High-level policy response (Failure)

- **Case 2: Capability searched**
  - High-level policy request
  - Translation: Data Conversion
  - Profile entry not matched
  - Capability Query
  - Search
  - Case 2: Capability Searched
  - NSF Creation Request
  - NSF Capability Registration
  - NSF Creation
  - NSF Creation Response (with NSF info)
  - Translation: Policy Generation
  - NSF Capability Registration
  - Low-level policy request
  - Low-level policy response
  - High-level policy response (Success)
Diego Lopez’s Comments

• No direct, interactive communication between Security Controller and Developer’s Management System (DMS) in NFV
  • [Answer] This I2NSF Hackathon Project has a direct, interactive communication between Security Controller and DMS via Registration Interface.

• Both Security Controller and DMS use Registration Interface to interact with NFV MANO Stack.
  • [Answer] This I2NSF Hackathon uses Ve-Vnfm Interface between DMS and VNFM in NFV MANO.

• Dynamic instantiation/de-instantiation of NSFs is out of the scope of this draft.
  • [Answer] We propose another draft about Lifecycle Management of NSFs:
Usages of Registration Interface based on Diego’s Comments.

- Developer’s Management System’s Use of Registration Interface
  - Registering NSFs and their capabilities into NFV MANO
  - Updating the capabilities of the registered NSFs

- Security Controller’s Use of Registration Interface
  - Retrieving the catalog of NSFs from NFV MANO
  - Requesting NFV MANO to instantiate NSFs
Registration Interface in I2NSF with NFV (1/4)

I2NSF User

Security Controller

NSF

NFV MANO Stack

run by network service provider

Registration interface

Developer’s Mgmt System

run by NSF vendor

Registration interface
Registration Interface in I2NSF with NFV (2/4)

- DMS uses the registration interface to provide NSFs and update the capabilities of the NSFs provided to SC.
  - MANO then creates a catalog of available NSFs and their capabilities that have been registered by DMSs.
• NFV MANO provides SC with the catalog of NSFs and their capabilities through the registration interface.
• SC searches the catalog for NSFs required to serve the request received from the I2NSF user.
• SC makes a selection of NSFs on the catalog.
• SC requests MANO to instantiate the select NSFs via the registration interface.
Next Steps

• We will change the current YANG data model to the YANG data model of Object-Oriented Style, such as Decorator patterns
  • Huawei will provide us with a sample YANG data model using Decorator patterns.

• After the proofreading by the authors of the NSF Capabilities Information Models document, we will correct the data model and finalize it.

• WG Adoption for our Draft about NSF Lifecycle Management:
  • draft-yang-i2nsf-nfv-architecture-04