Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2018

J. Jeong
E. Kim
Sungkyunkwan University
T. Ahn
Korea Telecom
R. Kumar
Juniper Networks
S. Hares
Huawei

July 3, 2017

I2NSF Consumer-Facing Interface YANG Data Model
draft-jeong-i2nsf-consumer-facing-interface-dm-02

Abstract

This document describes a YANG data model for the Consumer-Facing Interface between an Interface to Network Security Functions (I2NSF) User and Security Controller in an I2NSF system in a Network Functions Virtualization (NFV) environment. The data model is required for enabling different users of a given I2NSF system to define, manage, and monitor security policies for specific flows within an administrative domain.

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 http://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 January 4, 2018.

Copyright Notice

Table of Contents

1. Introduction

This document provides a YANG [RFC6020] data model that defines the required data for the Consumer-Facing Interface between an Interface to Network Security Functions (I2NSF) User and Security Controller in an I2NSF system [i2nsf-framework] in a Network Functions Virtualization (NFV) environment. The data model is required for enabling different users of a given I2NSF system to define, manage and monitor security policies for specific flows within an administrative domain. This document defines a YANG data model based on the information model of I2NSF Consumer-Facing Interface [client-facing-inf-im].

Data models are defined at a lower level of abstraction and provide many details. They provide details about the implementation of a protocol's specification, e.g., rules that explain how to map managed objects onto lower-level protocol constructs. Since conceptual

models can be implemented in different ways, multiple data models can be derived by a single information model.

The efficient and flexible provisioning of network functions by NFV leads to a rapid advance in the network industry. As practical applications, network security functions (NSFs), such as firewall, intrusion detection system (IDS)/intrusion protection system (IPS), and attack mitigation, can also be provided as virtual network functions (VNF) in the NFV system. By the efficient virtual technology, these VNFs might be automatically provisioned and dynamically migrated based on real-time security requirements. This document presents a YANG data model to implement security functions based on NFV.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC3444].

3. Terminology

This document uses the terminology described in [i2nsf-terminology][client-facing-inf-im][client-facing-inf-req].

4. Data Modeling for Consumer-Facing Interface

The main objective of this data model is to fully transform the information model [client-facing-inf-im] into a YANG data model that can be used for delivering control and management messages via the Consumer-Facing Interface between an I2NSF User and Security Controller for the I2NSF User's high-level security policies.

The semantics of the data model must be aligned with the information model of the Consumer-Facing Interface. The transformation of the information model was performed so that this YANG data model can facilitate the efficient delivery of the control or management messages.

This data model is designed to support the I2NSF framework that can be extended according to the security needs. In other words, the model design is independent of the content and meaning of specific policies as well as the implementation approach. This document suggests a VoIP/VoLTE security service as a use case for policy rule generation.

```
 module: ieft-i2nsf-consumer-facing-interface
   +--rw ietf-i2nsf-consumer-facing-interface
```

```
          +--rw multi-tenancy
          |  +--rw policy-domain* [policy-domain-id]
          |  |  +--rw policy-domain-id                 uint16
          |  |  +--rw name                             string
          |  |  +--rw address                          string
          |  |  +--rw contact                          string
          |  |  +--rw date                             yang:date-and-time
          |  |  +--rw authentication-method            string
          |  +--rw policy-tenant* [policy-tenant-id]
          |  |  +--rw policy-tenant-id                 uint16
          |  |  +--rw name                             string
          |  |  +--rw date                             yang:date-and-time
          |  |  +--rw domain                           string
          |  +--rw policy-role* [policy-role-id]
          |  |  +--rw policy-role-id                   uint16
          |  |  +--rw name                             string
          |  |  +--rw date                             yang:date-and-time
          |  |  +--rw access-profile                   string
          |  +--rw policy-user* [policy-user-id]
          |  |  +--rw policy-user-id                   uint16
          |  |  +--rw name                             string
          |  |  +--rw date                             yang:date-and-time
          |  |  +--rw password                         string
          |  |  +--rw email                            string
          |  |  +--rw scope-type?                      string
          |  |  +--rw scope-reference?                 string
          |  |  +--rw role                             string
          |  +--rw policy-mgmt-auth-method* [policy-mgnt-auth-method-id]
          |     +--rw policy-mgnt-auth-method-id       uint16
          |     +--rw name                             string
          |     +--rw date                             yang:date-and-time
          |     +--rw authentication-method            string
          |     +--rw mutual-authentication            boolean
          |     +--rw token-server                     string
          |     +--rw certificate-server               string
          |     +--rw single-sing-on-server            string
          +--rw policy-endpoint-groups
          |  +--rw meta-data-source* [meta-data-source-id]
          |  |  +--rw meta-data-source-id              uint16
          |  |  +--rw name                             string
          |  |  +--rw date                             yang:date-and-time
          |  |  +--rw tag-type?                        boolean
          |  |  +--rw tag-server-information?          string
          |  |  +--rw tag-application-protocol?        string
          |  |  +--rw tag-server-credential?           string
          |  +--rw user-group* [user-group-id]
          |  |  +--rw user-group-id                    uint16
          |  |  +--rw name?                            string
```

```
          |  |  +--rw date?                              yang:date-and-time
          |  |  +--rw group-type?                        string
          |  |  +--rw meta-data-server?                  string
          |  |  +--rw group-member?                      string
          |  |  +--rw risk-level?                        uint16
          |  +--rw device-group* [device-group-id]
          |  |  +--rw device-group-id                    uint16
          |  |  +--rw name?                              string
          |  |  +--rw date?                              yang:date-and-time
          |  |  +--rw group-type?                        string
          |  |  +--rw meta-data-server?                  string
          |  |  +--rw group-member?                      string
          |  |  +--rw risk-level?                        uint16
          |  +--rw application-group* [application-group-id]
          |  |  +--rw application-group-id               uint16
          |  |  +--rw name?                              string
          |  |  +--rw date?                              yang:date-and-time
          |  |  +--rw group-type?                        string
          |  |  +--rw meta-data-server?                  string
          |  |  +--rw group-member?                      string
          |  |  +--rw risk-level?                        uint16
          |  +--rw location-group* [location-group-id]
          |     +--rw location-group-id                  uint16
          |     +--rw name?                              string
          |     +--rw date?                              yang:date-and-time
          |     +--rw group-type?                        string
          |     +--rw meta-data-server?                  string
          |     +--rw group-member?                      string
          |     +--rw risk-level?                        uint16
          +--rw threat-prevention
          |  +--rw threat-feed* [threat-feed-id]
          |  |  +--rw threat-feed-id                     uint16
          |  |  +--rw name?                              string
          |  |  +--rw date?                              yang:date-and-time
          |  |  +--rw feed-type?                         enumeration
          |  |  +--rw feed-server?                       string
          |  |  +--rw feed-priority?                     uint16
          |  +--rw custom-list* [custom-list-id]
          |  |  +--rw custom-list-id                     uint16
          |  |  +--rw name?                              string
          |  |  +--rw date?                              yang:date-and-time
          |  |  +--rw list-type?                         enumeration
          |  |  +--rw list-property?                     enumeration
          |  |  +--rw list-content?                      string
          |  +--rw malware-scan-group* [malware-scan-group-id]
          |  |  +--rw malware-scan-group-id              uint16
          |  |  +--rw name?                              string
          |  |  +--rw date?                              yang:date-and-time
```

```
          |  |  +--rw signature-server?               string
          |  |  +--rw file-types?                     string
          |  |  +--rw malware-signatures?             string
          |  +--rw event-map-group* [event-map-group-id]
          |     +--rw event-map-group-id              uint16
          |     +--rw name?                           string
          |     +--rw date?                           yang:date-and-time
          |     +--rw security-events?                 string
          |     +--rw threat-map?                     string
          +--rw telemetry-data
          |  +--rw telemetry-data* [telemetry-data-id]
          |  |  +--rw telemetry-data-id               uint16
          |  |  +--rw name?                           string
          |  |  +--rw date?                           yang:date-and-time
          |  |  +--rw logs?                           boolean
          |  |  +--rw syslogs?                        boolean
          |  |  +--rw snmp?                           boolean
          |  |  +--rw sflow?                          boolean
          |  |  +--rw netflow?                        boolean
          |  |  +--rw interface-stats?                boolean
          |  +--rw telemetry-source* [telemetry-source-id]
          |  |  +--rw telemetry-source-id             uint16
          |  |  +--rw name?                           string
          |  |  +--rw date?                           yang:date-and-time
          |  |  +--rw source-type?                    string
          |  |  +--rw nsf-access-parameters?          string
          |  |  +--rw nsf-access-credentials?         string
          |  |  +--rw collection-interval?            uint16
          |  |  +--rw collection-method?              enumeration
          |  |  +--rw heartbeat-interval?             uint16
          |  |  +--rw qos-marking?                    uint8
          |  +--rw telemetry-destination* [telemetry-destination-id]
          |     +--rw telemetry-destination-id        uint16
          |     +--rw name?                           string
          |     +--rw date?                           yang:date-and-time
          |     +--rw collector-state?                string
          |     +--rw collector-access-parameters?    string
          |     +--rw collector-access-credentials?   string
          |     +--rw data-encoding?                  string
          |     +--rw data-transport?                 string
          +--rw policy-instance
             +--rw policy-calendar* [policy-calendar-id]
             |  +--rw policy-calendar-id              uint16
             |  +--rw name?                           string
             |  +--rw date?                           yang:date-and-time
             |  +--rw enforcement-type?               enumeration
             |  +--rw time-information?               string
             |  +--rw event-map?                      string
```

```
            +--rw policy-action* [policy-action-id]
            |  +--rw policy-action-id                string
            |  +--rw name?                           string
            |  +--rw date?                           yang:date-and-time
            |  +--rw primary-action?                 string
            |  +--rw secondary-action?               string
            +--rw policy-rule* [policy-rule-id]
            |  +--rw policy-rule-id                  string
            |  +--rw name?                           string
            |  +--rw date?                           yang:date-and-time
            |  +--rw source?                         string
            |  +--rw destination?                    string
            |  +--rw exception?                      string
            |  +--rw action?                         string
            |  +--rw precedence?                     uint8
            +--rw policy-instance* [policy-instance-id]
               +--rw policy-instance-id              string
               +--rw name?                           string
               +--rw date?                           yang:date-and-time
               +--rw rules?                          string
               +--rw scheduling-type?                enumeration
               +--rw scheduling-information?         string
               +--rw owner?                          string
```

Figure 1: Generic Data Model for Consumer-Facing Interface

5. YANG Data Model for Consumer-Facing Interface

This section describes a YANG data model for Consumer-Facing Interface, based on the information model of Consumer-Facing Interface to security controller [client-facing-inf-im].

```
<CODE BEGINS> file "ietf-i2nsf-consumer-facing-interface.yang"
module ietf-i2nsf-consumer-facing-interface {
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-consumer-facing-interface";
  prefix
    capability-interface;

  import ietf-yang-types {

    prefix inet;
  }

  import ietf-yang-types {
    prefix yang;
  }
```

```
   organization
     "IETF I2NSF (Interface to Network Security Functions)
      Working Group";

   contact
     "WG Web: <http://tools.ietf.org/wg/i2nsf>
      WG List: <mailto:i2nsf@ietf.org>

      WG Chair: Adrian Farrel
      <mailto:Adrain@olddog.co.uk>

      WG Chair: Linda Dunbar
      <mailto:Linda.duhbar@huawei.com>

      Editor: Jaehoon Paul Jeong
      <mailto:pauljeong@skku.edu>";

   description
     "This module defines a YANG data module for consumer-facing
      interface to security controller.";

   revision "2017-07-03"{
     description "Initial revision";
     reference
       "draft-kumar-i2nsf-client-facing-interface-im-02";
   }

   //Groupings
   container ietf-i2nsf-consumer-facing-interface {
   description
   " ";
     container multi-tenancy {
         description
           "The descriptions of multi-tenancy.";

         list policy-domain {
           key "policy-domain-id";
           leaf policy-domain-id {
             type uint16;
             mandatory true;
             description
               "This represents the list of domains.";
           }
           description
           "this represent the list of policy domains";
           leaf name {
             type string;
             mandatory true;
```

```
            description
              "Name of the organization or customer representing
              this domain.";
            }

            leaf address {
              type string;
              description
                "address of an organization or customer.";
            }

            leaf contact {
              type string;
              mandatory true;
              description
                "contact information of the organization
                 or customer.";
            }

            leaf date {
              type yang:date-and-time;
              mandatory true;
              description
                "The date when this account was created
                or last modified.";
            }

            leaf authentication-method {
              type string;
              mandatory true;
              description
                "The description of authentication method;
                token-based, password, certificate,
                single-sign-on";
            }
          }

          list policy-tenant {
            key "policy-tenant-id";
            leaf policy-tenant-id {
              type uint16;
              mandatory true;
              description
                "The policy tenant id.";
            }
            description
            "This represents the list of tenants";
            leaf name {
```

```
            type string;
            mandatory true;
            description
              "Name of the Department or Division within
               an organization.";
          }

          leaf date {
            type yang:date-and-time;
            mandatory true;
            description
              "Date this account was created or last modified.";
          }

          leaf domain {
            type string;
            mandatory true;
            description
            "This field identifies the domain to which this
            tenant belongs. This should be reference to a
            'Policy-Domain' object.";
          }
        }

        list policy-role {
          key "policy-role-id";
          leaf policy-role-id {
          type uint16;
          mandatory true;
          description
            "This defines a set of permissions assigned
            to a user in an organization that want to manage
            its own Security Policies.";
          }
          description
          "This represents the list of policy roles.";
          leaf name {
            type string;
            mandatory true;
            description
              "This field identifies name of the role.";
          }

          leaf date {
            type yang:date-and-time;
            mandatory true;
            description
              "Date this role was created or last modified.";
```

```
          }

          leaf access-profile {
            type string;
            mandatory true;
            description
              "This field identifies the access profile for the
              role. The profile grants or denies access to policy
              objects.  Multiple access profiles can be
              concatenated together.";
          }
        }

        list policy-user {
          key "policy-user-id";
          leaf policy-user-id {
          type uint16;
          description
            "This represents the policy-user-id.";
          }
          description
          "This represents the list of policy users.";
          leaf name {
            type string;
            mandatory true;
            description
              "The name of a user.";
          }

          leaf date {
            type yang:date-and-time;
            mandatory true;
            description
              "Date this user was created or last modified";
          }

          leaf password {
            type string;
            mandatory true;
            description
              "User password for basic authentication";
          }

          leaf email {
            type string;
            mandatory true;
            description
              "The email account of a user";
```

```
          }

          leaf scope-type {
            type string;
            description
              "identifies whether a user has domain-wide
              or tenant-wide privileges";
          }

          leaf scope-reference {
            type string;
            description
              "This references policy-domain or policy-tenant
              to identify the scope.";
          }

          leaf role {
            type string;
            mandatory true;
            description
              "This references policy-role to define specific
              permissions";
          }
        }

        list policy-mgmt-auth-method {
          key "policy-mgnt-auth-method-id";
          leaf policy-mgnt-auth-method-id {
          type uint16;
          description
            "This represents the authentication method id.";
          }
          description
          "The descriptions of policy management
            authentication methods.";
          leaf name {
            type string;
            mandatory true;
            description
              "name of the authentication method";
          }

          leaf date {
            type yang:date-and-time;
            mandatory true;
            description
              "date when the authentication method
              was created";
```

```
          }

          leaf authentication-method {
            type string;
            mandatory true;
            description
              "The description of authentication method;
              token-based, password, certificate,
              single-sign-on";
          }

          leaf mutual-authentication {
            type boolean;
            mandatory true;
            description
              "To identify whether the authentication
               is mutual";
          }

         leaf token-server {
            type string;
            mandatory true;
            description
              "The token-server information if the
              authentication method is token-based";
          }

          leaf certificate-server {
            type string;
            mandatory true;
            description
              "The certificate-server information if
              the authentication method is certificate-based";
          }

          leaf single-sing-on-server {
            type string;
            mandatory true;
            description
              "The single-sign-on-server information
              if the authentication method is
              single-sign-on-based";
          }
        }
      }

    container policy-endpoint-groups {
        description
```

```
            "A logical entity in their business
            environment, where a security policy
            is to be applied.";

          list meta-data-source {
            key "meta-data-source-id";
            leaf meta-data-source-id {
            type uint16;
            mandatory true;
            description
              "This represents the meta-data source id.";
            }
            description
            "This represents the meta-data source.";
            leaf name {
              type string;
              mandatory true;
              description
                "This identifies the name of the
                meta-datas-ource.";
            }
            leaf date {
              type yang:date-and-time;
              mandatory true;
              description
                "This identifies the date this object was
                created or last modified.";
            }

            leaf tag-type {
              type boolean;
              description
                "This identifies the group type; user group,
                app group or device group.";
            }

            leaf tag-server-information {
              type string;
              description
                "The description of suthentication method;
                token-based, password, certificate,
                single-sign-on";
            }
            leaf tag-application-protocol {
              type string;
              description
                "This filed identifies the protocol e.g. LDAP,
                Active Directory, or CMDB";
```

```
          }
          leaf tag-server-credential {
            type string;
            description
              "This field identifies the credential
              information needed to access the tag server";
          }
        }

        list user-group{
          key "user-group-id";
          leaf user-group-id {
          type uint16;
          mandatory true;
          description
            "This represents the the user group id.";
          }
          description
          "This represents the user group.";
          leaf name {
            type string;
            description
              "This field identifies the name of user-group.";
          }

          leaf date {
            type yang:date-and-time;
            description
              "when this user-group was created or last modified.";
          }
          leaf group-type {
            type string;
            description
              "This describes the group type; User-tag,
              User-name or IP-address.";
          }

          leaf meta-data-server {
            type string;
            description
              "This references metadata source";
          }

          leaf group-member {
            type string;
            description
              "This describes the user-tag information";
          }
```

```
            leaf risk-level {
              type uint16;
              description
                "This represents the threat level; valid range
                 may be 0 to 9.";
              }
            }

            list device-group{
              key "device-group-id";
              leaf device-group-id {
              type uint16;
               description
                "This represents a device group id.";
              }
              description
                "This represents a device group.";
              leaf name {
                type string;
              description
                  "This field identifies the name of
                  a device-group.";
              }
              leaf date {
                type yang:date-and-time;
                description
                "The date when this group was create or
                last modified.";
              }

              leaf group-type {
                type string;
                description
                  "This describes the group type; device-tag,
                  device-name or IP-address.";
              }

              leaf meta-data-server {
                type string;
                description
                  "This references meta-data-source
                  object.";
              }

              leaf group-member {
                type string;
                description
                  "This describes the device-tag, device-name or
```

```
            IP-address information";
          }

          leaf risk-level {
            type uint16;
            description
              "This represents the threat level; valid range
              may be 0 to 9.";
          }
        }

        list application-group{
          key "application-group-id";
          leaf application-group-id {
          type uint16;
          description
          "This represents an application group id.";
          }
          description
          "This represents an application group.";
          leaf name {
            type string;
            description
            "This field identifies the name of
            an application group";
          }

          leaf date {
            type yang:date-and-time;
            description
            "The date when this group was created or
            last modified.";
          }

          leaf group-type {
            type string;
            description
              "This identifies the group type;
              application-tag, application-name or
              IP-address.";
          }

          leaf meta-data-server {
            type string;
            description
              "This references meta-data-source
              object.";
          }
```

```
            leaf group-member {
              type string;
              description
                "This describes the application-tag,
                application-name or IP-address information";
            }

            leaf risk-level {
              type uint16;
              description
                "This represents the threat level; valid range
                 may be 0 to 9.";
            }
          }

          list location-group{
            key "location-group-id";
            leaf location-group-id {
            type uint16;
            description
            "This represents a location group id.";
            }
            description
            "This represents a location group.";
            leaf name {
              type string;
              description
              "This field identifies the name of
              a location group";

            }

            leaf date {
              type yang:date-and-time;
              description
              "The date when this group was created or
              last modified.";
            }

            leaf group-type {
              type string;
              description
                "This identifies the group type;
                location-tag, location-name or
                IP-address.";
            }

            leaf meta-data-server {
```

```
            type string;
            description
              "This references meta-data-source
              object.";
          }

          leaf group-member {
            type string;
            description
              "This describes the location-tag,
              location-name or IP-address information";
          }

          leaf risk-level {
            type uint16;
            description
              "This represents the threat level; valid range
               may be 0 to 9.";
          }
       }
    }

    container threat-prevention {
      description
      "this describes the list of threat-preventions.";

        list threat-feed {
          key "threat-feed-id";
          leaf threat-feed-id {
          type uint16;
          mandatory true;
          description
            "This represents the threat-feed-id.";
          }
          description
            "This represents the threat feed within the
            threat-prevention-list.";
          leaf name {
            type string;
            description
              "Name of the theat feed.";
          }

          leaf date {
            type yang:date-and-time;
            description
              "when the threat-feed was created.";
          }
```

```
            leaf feed-type {
              type enumeration {
                enum unknown {
                  description
                    "feed-type is unknown.";
                }
                enum ip-address {
                  description
                    "feed-type is IP address.";
                }
                enum url {
                  description
                    "feed-type is URL.";
                }
              }
              mandatory true;
              description
                "This determined whether the feed-type is IP address
                based or URL based.";
            }

            leaf feed-server {
              type string;
              description
                "this contains threat feed server information.";
            }

            leaf feed-priority {
              type uint16;
              description
                "this describes the priority of the threat from
                0 to 5, where 0 means the threat is minimum and
                5 meaning the maximum.";
            }
          }

          list custom-list {
            key "custom-list-id";
            leaf custom-list-id {
            type uint16;
            description
            "this describes the custom-list-id.";
            }
            description
            "this describes the threat-prevention custom list.";
            leaf name {
              type string;
              description
```

```
            "Name of the custom-list.";
          }

          leaf date {
            type yang:date-and-time;
            description
              "when the custom list was created.";
          }

          leaf list-type {
            type enumeration {
              enum unknown {
                description
                  "list-type is unknown.";
              }
              enum ip-address {
                description
                  "list-type is IP address.";
              }
              enum url {
                description
                  "list-type is URL.";
              }
            }
            mandatory true;
            description
              "This determined whether the feed-type is IP address
              based or URL based.";
          }

         leaf list-property {
            type enumeration {
              enum unknown {
                description
                  "list-property is unknown.";
              }
              enum blacklist {
                description
                  "list-property is blacklist.";
              }
              enum whitelist {
                description
                  "list-property is whitelist.";
              }
            }
            mandatory true;
            description
              "This determined whether the list-type is blacklist
```

```
                or whitelist.";
            }

            leaf list-content {
              type string;
              description
                "This describes the contents of the custom-list.";
            }
        }
        list malware-scan-group {
            key "malware-scan-group-id";
            leaf malware-scan-group-id {
            type uint16;
            mandatory true;
            description
            "This is the malware-scan-group-id.";
            }
            description
            "This represents the malware-scan-group.";
            leaf name {
              type string;
              description
                "Name of the malware-scan-group.";
            }

            leaf date {
              type yang:date-and-time;
              description
                "when the malware-scan-group was created.";
            }

            leaf signature-server {
              type string;
              description
                "This describes the signature server of the
                malware-scan-group.";
            }

            leaf file-types {
              type string;
              description
                "This contains a list of file types needed to
                be scanned for the virus.";
            }

            leaf malware-signatures {
              type string;
              description
```

```
                "This contains a list of malware signatures or hash.";
            }
        }

        list event-map-group {
            key "event-map-group-id";
            leaf event-map-group-id {
            type uint16;
            mandatory true;
            description
            "This is the event-map-group-id.";
            }
            description
            "This represents the event map group.";

            leaf name {
              type string;
              description
                "Name of the event-map.";
            }

            leaf date {
              type yang:date-and-time;
              description
                "when the event-map was created.";
            }

            leaf security-events {
              type string;
              description
                "This contains a list of security events.";
            }

            leaf threat-map {
              type string;
              description
                "This contains a list of threat levels.";
            }
          }
        }

        container telemetry-data {
            description
            "Telemetry provides visibility into the network
            activities which can be tapped for further
            security analytics, e.g., detecting potential
            vulnerabilities, malicious activities, etc.";
```

```
          list telemetry-data {
            key "telemetry-data-id";
            leaf telemetry-data-id {
            type uint16;
            mandatory true;
            description
            "This is ID for telemetry-data-id.";
            }
            description
            "This is ID for telemetry-data.";
            leaf name {
              type string;
              description
                "Name of the telemetry-data object.";
            }

            leaf date {
              type yang:date-and-time;
              description
                "This field states when the telemery-data
                object was created.";
            }

            leaf logs {
              type boolean;
              description
                "This field identifies whether logs
                need to be collected.";
            }

            leaf syslogs {
              type boolean;
              description
                "This field identifies whether System logs
                need to be collected.";
            }

            leaf snmp {
              type boolean;
              description
                "This field identifies whether 'SNMP traps' and
                'SNMP alarms' need to be collected.";
            }

            leaf sflow {
              type boolean;
              description
                "This field identifies whether 'sFlow' data
```

```
              need to be collected.";
            }

            leaf netflow {
              type boolean;
              description
                "This field identifies whether 'NetFlow' data
                need to be collected.";
            }

            leaf interface-stats {
              type boolean;
              description
                "This field identifies whether 'Interface' data
                such as packet bytes and counts need to be
                collected.";
            }
          }

          list telemetry-source {
            key "telemetry-source-id";
            leaf telemetry-source-id {
            type uint16;
            mandatory true;
            description
            "This is ID for telemetry-source-id.";
            }
            description
            "This is ID for telemetry-source.";
            leaf name {
              type string;
              description
                "This identifies the name of this object.";
            }

            leaf date {
              type yang:date-and-time;
              description
                "Date this object was created or last modified";
            }

            leaf source-type {
              type string;
              description
                "This should have one of the following type of
                the NSF telemetry source: NETWORK-NSF,
                FIREWALL-NSF, IDS-NSF, IPS-NSF,
                PROXY-NSF, VPN-NSF, DNS, ACTIVE-DIRECTORY,
```

```
            IP Reputation Authority, Web Reputation
            Authority, Anti-Malware Sandbox, Honey Pot,
            DHCP, Other Third Party, ENDPOINT";
          }

          leaf nsf-access-parameters {
            type string;
            description
              "This field contains information such as
              IP address and protocol (UDP or TCP) port
              number of the NSF providing telemetry data.";
          }

          leaf nsf-access-credentials {
            type string;
            description
              "This field contains username and password
              to authenticate with the NSF.";
          }

          leaf collection-interval {
            type uint16;
            units seconds;
            default 5000;
            description
            "This field contains time in milliseconds
             between each data collection. For example,
             a value of 5000 means data is streamed to
             collector every 5 seconds. Value of 0 means
             data streaming is event-based";
          }

          leaf collection-method {
            type enumeration {
              enum unknown {
                description
                  "collection-method is unknown.";
              }
              enum push-based {
                description
                  "collection-method is PUSH-based.";
              }
              enum pull-based {
                description
                  "collection-method is PULL-based.";
              }
            }
            description
```

```
            "This field contains a method of collection,
            i.e., whether it is PUSH-based or PULL-based.";
          }

          leaf heartbeat-interval {
            type uint16;
            units seconds;
            description
            "time in seconds the source sends telemetry
            heartbeat.";
          }

          leaf qos-marking {
            type uint8;
            description
            "DSCP value must be contained in this field.";
          }
        }
        list telemetry-destination {
            key "telemetry-destination-id";
            leaf telemetry-destination-id {
            type uint16;
            description
            "this represents the telemetry-destination-id";
            }
            description
            "This object contains information related to
            telemetry destination. The destination is
            usually a collector which is either a part of
            Security Controller or external system
            such as Security Information and Event
            Management (SIEM).";

            leaf name {
              type string;
              description
                "This identifies the name of this object.";
            }

            leaf date {
              type yang:date-and-time;
              description
                "Date this object was created or last
                modified";
            }

            leaf collector-state {
              type string;
```

```
            description
              "This describes collector state information.";
          }
          leaf collector-access-parameters {
            type string;
            description
              "ip address and port number of the nsf
              providing telemetry data.";
          }

          leaf collector-access-parameters {
            type string;
            description
              "This field contains information such as
              IP address and protocol (UDP or TCP) port
              number for the collector's destination.";
          }

          leaf collector-access-credentials {
            type string;
            description
              "This field contains username and password
              for the collector.";
          }

          leaf data-encoding {
            type string;
            description
            "This field contains the telemetry data encoding
            in the form of schema.";
          }

          leaf data-transport {
            type string;
            description
            "This field contains streaming telemetry data
            protocols. This could be gRPC, protocol
            buffer over UDP, etc.";
          }
        }
      }

      container policy-instance {
          description
            "This object is a policy instance to have
            complete information such as where and when
            a policy need to be applied.";
```

```
          list policy-calendar {
            key "policy-calendar-id";
            leaf policy-calendar-id {
            type uint16;
            description
            "this represents the policy-calendar-id.";
            }
            description
              "This object contains information related to
              scheduling a policy. The policy could be
              activated based on a time calendar or security
              event including threat level changes.";

            leaf name {
              type string;
              description
                "Name of the policy-calendar object.";
            }

            leaf date {
              type yang:date-and-time;
              description
                "The date when this object was created or
                last modified.";
            }

            leaf enforcement-type {
              type enumeration {
                enum unknown {
                  description
                    "enforcement-type is unknown.";
                }
                enum admin-enforced {
                  description
                    "enforcement-type is ADMIN-ENFORCED.";
                }
                enum time-enforced {
                  description
                    "enforcement-type is TIME-ENFORCED.";
                }
                enum event-enforced {
                  description
                    "enforcement-type is EVENT-ENFORCED.";
                }
              }
              description
                "This field identifies whether the policy
                enforcement is 'ADMIN-ENFORCED' or
```

```
            'TIME-ENFORCED', or 'EVENT-ENFORCED'.";
        }

        leaf time-information {
          type string;
          description
            "This field contains time calendar such as
            'BEGIN-TIME' and 'END-TIME' for one time
            enforcement or recurring time calendar for
            periodic enforcement.";
        }

        leaf event-map {
          type string;
          description
            "This field contains security events and
            threat map in order to determine when a
            policy need to be activated.";
        }
      }

      list policy-action {
        key "policy-action-id";
        leaf policy-action-id {
        type string;
        mandatory true;
        description
          "this represents the policy-action-id.";
        }
        description
          "This object represents actions that a
          Security Admin wants to perform based on
          a certain traffic class.";
        leaf name {
          type string;
          description
            "The name of the policy-action object.";
        }

        leaf date {
          type yang:date-and-time;
          description
            "When the object was created or last
            modified.";
        }

        leaf primary-action {
          type string;
```

```
            description
              "This field identifies the action when a rule
              is matched by NSF. The action could be one of
              'PERMIT', 'DENY', 'RATE-LIMIT', 'TRAFFIC-CLASS',
              'AUTHENTICATE-SESSION', 'IPS, 'APP-FIREWALL', etc.";
            }

            leaf secondary-action {
              type string;
              description
                "This field identifies additional actions if
                a rule is matched. This could be one of 'LOG',
                'SYSLOG', 'SESSION-LOG', etc.";
            }
          }

          list policy-rule {
            key "policy-rule-id";
            leaf policy-rule-id {
            type string;
            mandatory true;
            description
            "this represents the policy-rule-id";
            }
            description
              "This object represents rules that a
              Security Admin want to define in order
              to express its business objectives in
              a Security Policy.";
            leaf name {
              type string;
              description
                "This field identifies the name of
                this object.";
            }

            leaf date {
              type yang:date-and-time;
              description
                "When the object was created or last
                modified.";
            }

            leaf source {
              type string;
              description
                "This field identifies the source of
                the traffic. This could be reference to
```

```
              either 'Policy Endpoint Group' or
              'Threat-Feed' or 'Custom-List' if Security
              Admin wants to specify the source; otherwise,
              the default is to match all traffic.";
          }

          leaf destination {
            type string;
            description
              "This field identifies the destination of
              the traffic. This could be reference to
              either 'Policy Endpoint Group' or
              'Threat-Feed' or 'Custom-List' if Security
              Admin wants to specify the destination;
              otherwise, the default is to match all
              traffic.";
          }

          leaf exception {
            type string;
            description
              "This field identifies the exception
              consideration when 'Source' and
              'Destination' are matched for a given
              communication.  This should be reference
              to 'Policy Endpoint Group' object.";
          }

          leaf action {
            type string;
            description
              "This field identifies the action taken
              when 'Source' and 'Destination' are matched
              for a given communication.";
          }

          leaf precedence {
            type uint8;
            description
              "This field identifies the precedence
              assigned to this rule by Security Admin.
              This is helpful in conflict resolution
              when two or more rules match a given
              traffic class.";
          }
        }

        list policy-instance {
```

```
            key "policy-instance-id";
            leaf policy-instance-id {
            type string;
            mandatory true;
            description
            "this represents the policy-instance-id";
            }
            description
              "This object represents a mechanism to
              express a Security Policy by Security Admin
              to Security Controller via Consumer-Facing
              Interface. The policy would be enforced by
              an NSF.";
            leaf name {
              type string;
              description
                "This field identifies the name of this
                object.";
            }

            leaf date {
              type yang:date-and-time;
              description
                "Date this object was created or last
                modified.";
            }

            leaf rules {
              type string;
              description
                "This field contains a list of rules.
                If the rule does not have a user-defined
                precedence, then any conflict must be
                manually resolved.";
            }

            leaf scheduling-type {
              type enumeration {
                enum unknown {
                  description
                    "scheduling-type is unknown.";
                }
                enum time-calendar {
                  description
                    "scheduling-type is time-calendar.";
                }
                enum event-map {
                  description
```

```
                    "scheduling-type is event-map.";
                }
              }
              description
                "This field specifies when this policy
                should be scheduled. The policy could be
                scheduled based on time calendar or
                event-map.";
            }

            leaf scheduling-information {
              type string;
              description
                "This field contains either the 'Calendar'
                or 'Event-map' based on 'Schedule type'.";
            }

            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.";
            }
          }
        }
      }
    }
  }
<CODE ENDS>
```

Figure 2: YANG Data Model for Consumer-Facing_interface

6. Security Considerations

The data model for the I2NSF Consumer-Facing Interface is derived from the I2NSF Consumer-Facing Interface Information Model [client-facing-inf-im], so the same security considerations with the information model should be included in this document. The data model needs to support a mechanism to protect Consumer-Facing Interface to Security Controller.

7. Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.R-20160222-002755, Cloud based Security Intelligence Technology Development for the Customized Security Service Provisioning).

This document has greatly benefited from inputs by Hyoungshick Kim, Hoon Ko, Mahdi F. Dachmehchi, Seungjin Lee, Jinyong Tim Kim, and Daeyoung Hyun.

8. References

8.1. Normative References

[RFC3444] Pras, A., "On the Difference between Information Models and Data Models", RFC 3444, January 2003.

8.2. Informative References

[i2nsf-framework]
Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", draft-ietf-i2nsf-framework-05 (work in progress), May 2017.

[client-facing-inf-req]
Kumar, R., Lohiya, A., Qi, D., Bitar, N., Palislamovic, S., and L. Xia, "Requirements for Client-Facing Interface to Security Controller", draft-ietf-i2nsf-client-facing-interface-req-01 (work in progress), April 2017.

[client-facing-inf-im]
Kumar, R., Lohiya, A., Qi, D., Bitar, N., Palislamovic, S., and L. Xia, "Information model for Client-Facing Interface to Security Controller", draft-kumar-i2nsf-client-facing-interface-im-02 (work in progress), April 2017.

[i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., Birkholz, H., and L. Xia, "Information model for Client-Facing Interface to Security Controller", draft-ietf-i2nsf-terminology-03 (work in progress), March 2017.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

Appendix A. Changes from draft-jeong-i2nsf-consumer-facing-interface-dm-01

The following changes have been made from draft-jeong-i2nsf-consumer-facing-interface-dm-01:

- The block diagram representing the overall architecture of security management system has been remove in this draft (Section 5 in draft-jeong-i2nsf-consumer-facing-interface-dm-01) as it is more suitable to be included in the information model draft than the data model.

- Sections 4 and 5 have been revised to produce a data tree model and YANG data model according to the information model suggested in the draft about the I2NSF Consumer-Facing Interface Information Model [client-facing-inf-im].

- Overall editorial errors are corrected.

Appendix B. Use Case: Policy Instance Example for VoIP/VoLTE Security Services

The following shows the example data tree model for the VoIP/VoLTE services. Multi-tenancy, endpoint groups, threat prevention, and telemetry data components are general part of the tree model, so we can just modify the policy instance in order to generate and enforce high-level policies.

The policy-calendar can act as a scheduler to set the star and end time to block calls which uses suspicious ids, or calls from other countries.

```
module ietf-i2nsf-consumer-facing-interface-policy-instance
 +--rw policy-instance
    +--rw policy-rule* [policy-rule-id]
    |  +--rw policy-rule-id        uint16
    |  +--rw name?                 string
    |  +--rw date?                 yang:date-and-time
    |  +--rw source?               string
    |  +--rw destination?          string
    |  +--rw exception?            boolean
    |  +--rw exception-detail?     string
    +--rw action* [action-id]
    |  +--rw action-id             string
    |  +--rw name?                 string
    |  +--rw date?                 yang:date-and-time
    |  +--rw primary-action?       string
    |  +--rw secondary-action?     string
    +--rw precedence* [precedence-id]
    |  +--rw precedence-id         string
    |  +--rw rule-exist?           boolean
    +--rw event* [event-id]
    |  +--rw event-id              string
    |  +--rw security-event?       string
    |  +--rw threat-map?           string
    |  +--rw enable?               boolean
    +--rw condition* [condition-id]
    |  +--rw condition-id          string
    |  +--rw caller* [caller-id]
    |  |  +--rw caller-id          uint16
    |  |  +--rw caller-id-id?      string
    |  |  +--rw caller-country?    string
    |  |  +--rw caller-city?       string
    |  +--rw callee* [callee-id]
    |     +--rw callee-id          uint16
    |     +--rw callee-id-id?      string
    |     +--rw callee-country?    string
    |     +--rw callee-city?       string
    +--rw policy-calendar* [policy-calendar-id]
       +--rw policy-calendar-id    uint16
       +--rw name?                 string
       +--rw date?                 yang:date-and-time
       +--rw enforcement-type?     string
       +--rw begin-time?           yang:date-and-time
       +--rw end-time?             yang:date-and-time
```

Figure 3: Policy Instance Example for VoIP/VoLTE Security Services

Appendix C. Policy Instance YANG Example for VoIP/VoLTE Security Services

The following YANG data model is a policy instance for VoIP/VoLTE security services. The policy-calendar can act as a scheduler to set the start time and end time to block malicious calls which use suspicious IDs, or calls from other countries.

<CODE BEGINS> file "ietf-i2nsf-consumer-facing-inf-voip"

```
module ietf-i2nsf-consumer-facing-interface {
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-consumer-facing-interface";
  prefix
    capability-interface;

  import ietf-yang-types {

    prefix inet;
  }

  import ietf-yang-types {
    prefix yang;
  }

  organization
    "IETF I2NSF (Interface to Network Security Functions)
     Working Group";

  contact
    "WG Web: <http://tools.ietf.org/wg/i2nsf>
     WG List: <mailto:i2nsf@ietf.org>

     WG Chair: Adrian Farrel
     <mailto:Adrain@olddog.co.uk>

     WG Chair: Linda Dunbar
     <mailto:Linda.duhbar@huawei.com>

     Editor: Jaehoon Paul Jeong
     <mailto:pauljeong@skku.edu>";

  description
    "This module defines a YANG data module for consumer-facing
     interface to security controller.";

  revision "2017-07-03"{
    description "Initial revision";
```

```
     reference
       "draft-kumar-i2nsf-client-facing-interface-im-02";
   }

   //Groupings
   container policy-instance {
      description
      "this describes the policy instances.";

         list policy-rule {
           key "policy-rule-id";
           description
           "This represents the policy-rule of a
           policy instance.";

           leaf policy-rule-id {
             type uint16;
             description
             "policy rule id.";
           }

           leaf name {
             type string;
             description
               "Name of the policy-rule.";
           }

           leaf date {
             type yang:date-and-time;
             description
               "The date when the rule was created.";
           }

           leaf source {
             type string;
             description
               "This references either end-point-group,
               threat-feed, or custom-list.";
           }

           leaf destination {
             type string;
             description
               "This references either end-point-group,
               threat-feed, or custom-list.";
           }

           leaf exception {
```

```
            type boolean;
            description
              "This describes whether an exception has
              occurred or not.";
          }

          leaf exception-detail{
            type string;
            description
              "This includes detailed information about
              source and destination of
              an exception.";
          }
        }
        list action {
          key "action-id";
          description
              "This represents the action of a policy-rule.";
          leaf action-id {
            type string;
            mandatory true;
            description
              "This represents the action-id of a policy-rule.";
          }
          leaf name {
            type string;
            description
              "The action name.";
          }
          leaf date {
            type yang:date-and-time;
            description
              "When the action was taken.";
          }

          leaf primary-action {
            type string;
            description
              "This includes actions such as permit,
              mirroring, rate-limit, ips, app-firewall,
              auth-session, and etc";
          }

          leaf secondary-action {
            type string;
            description
              "This includes optional actions such as
              logging, system logging and session logging.";
```

```
          }
        }
        list precedence {
          key "precedence-id";
          description
          "This describes whether there is a preceeding
            rule and causes problems.";
          leaf precedence-id{
            type string;
            mandatory true;
            description
            "This represent the precedence-id of
            a policy-rule.";
          }
          leaf rule-exist {
            type boolean;
            description
            "This determines whether there is a preceeding.";
          }
        }
        list event {
          key "event-id";
          description
          "This represents the security event of a
              policy-rule.";
          leaf event-id {
            type string;
            mandatory true;
            description
              "This represents the event-id.";
          }
          leaf security-event {
            type string;
            description
              "This references the security event in the
              threat-prevention .";
          }
          leaf threat-map {
            type string;
            description
              "This references the threat-map in the
              threat-prevention.";
          }
          leaf enable {
            type boolean;
            description
              "This determines whether the condition
              matches the security event or not.";
```

```
          }
        }
        list condition {
          key "condition-id";
          description
          "This represents the condition of a
              policy-rule.";
          leaf condition-id {
            type string;
            description
              "This represents the condition-id.";
          }
          list caller {
            key "caller-id";
            description
            "this represents the list of callers.";
            leaf caller-id {
              type uint16;
              description
              "the id of the caller.";
            }
            leaf caller-id-id{
              type string;
              description
                "The caller's number.";
            }
            leaf caller-country {
              type string;
              description
                "This determines the country of the caller.";
            }
            leaf caller-city {
              type string;
              description
                "This determines the city of the caller.";
            }
          }

          list callee {
            key "callee-id";
            description
            "this represents the list of callees";
            leaf callee-id {
              type uint16;
              description
              "The id of the callee.";
            }
            leaf callee-id-id {
```

```
            type string;
            description
              "The callee's number.";
          }
          leaf callee-country {
            type string;
            description
              "This determines the country of the callee.";
          }
          leaf callee-city {
            type string;
            description
              "This determines the city of the callee.";
          }
        }
      }
      list policy-calendar {
          key "policy-calendar-id";
          description
          "this represents the policy calendar list.";
          leaf policy-calendar-id {
          type uint16;
          description
          "The id of the policy calendar.";
          }
          leaf name {
            type string;
            description
              "The name of the policy-calendar.";
          }
          leaf date {
            type yang:date-and-time;
            description
              "The date when this calender was
              created or last modified.";
          }
          leaf enforcement-type {
            type string;
            description
              "Whether the policy enforcement is
              admin-enforced, time-enforced, or
              event-enforced.";
          }
          leaf begin-time {
            type yang:date-and-time;
            description
              "The starting time for blocking
              suspicious calls.";
```

```
            }
            leaf end-time {
              type yang:date-and-time;
              description
                "The time when blocking ends.";
            }
          }
        }
    }
 <CODE ENDS>
```

Figure 4: Policy Instance YANG Example for VoIP/VoLTE Security Services

Authors' Addresses

Jaehoon Paul Jeong
Department of Software
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon, Gyeonggi-Do 16419
Republic of Korea

Phone: +82 31 299 4957
Fax: +82 31 290 7996
EMail: pauljeong@skku.edu
URI: http://iotlab.skku.edu/people-jaehoon-jeong.php

Eunsoo Kim
Department of Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon, Gyeonggi-Do 16419
Republic of Korea

Phone: +82 31 299 4104
EMail: eskim86@skku.edu
URI: http://seclab.skku.edu/people/eunsoo-kim/

Tae-Jin Ahn
Korea Telecom
70 Yuseong-Ro, Yuseong-Gu
Daejeon 305-811
Republic of Korea

Phone: +82 42 870 8409
EMail: taejin.ahn@kt.com

Rakesh Kumar
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA

EMail: rkkumar@juniper.net

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Phone: +1-734-604-0332
EMail: shares@ndzh.com