I2NSF Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 8, 2019

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

November 4, 2018

I2NSF Consumer-Facing Interface YANG Data Model
draft-ietf-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

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 Security Policies 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.

Multi-tenancy in this document enables multiple administrative domains in order to manage application resources. An Enterprise organization may have multiple tenants or departments such as human resources (HR), finance, and legal departments. Thus, we need an object which defines a set of permissions assigned to a user in an organization that wants to manage its own Security Policies. You can think of it as a way to assign policy users to a job function or a set of permissions within the organization. The policy-role object SHALL have Name, Date and access-profile to grant or deny permissions for the perpose of security policy management.

```
module: policy-general
    +--rw policy
    |  +--rw rule* [rule-id]
    |     +--rw rule-id                      uint16
    |     +--rw name?                        string
    |     +--rw date?                        yang:date-and-time
    |     +--rw case?                        string
    |     +--rw event* [event-id]
    |     |  +--rw event-id                  string
    |     |  +--rw name?                     string
    |     |  +--rw date?                     yang:date-and-time
    |     |  +--rw event-type?               string
    |     |  +--rw time-information?         string
    |     |  +--rw event-map-group?    -> /threat-feed/event-map-group
    |     |                               /event-map-group-id
    |     |  +--rw enable?                   boolean
    |     +--rw condition* [condition-id]
    |     |  +--rw condition-id              string
    |     |  +--rw source?                   string
    |     |  +--rw destination?              string
    |     |  +--rw match?                    boolean
    |     |  +--rw match-direction?          string
    |     |  +--rw exception?                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 owner?                    string
```

```
    +--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 policy-tenant* [policy-tenant-id]
    |  |  |  +--rw policy-tenant-id          uint16
    |  |  |  +--rw name                      string
    |  |  |  +--rw date                      yang:date-and-time
    |  |  |  +--rw domain?               ->  /multi-tenancy
    |  |  |                                  /policy-domain
    |  |  |                                  /policy-domain-id
    |  |  +--rw authentication-method?   ->  /multi-tenancy
    |  |                                     /policy-mgnt-auth-method
    |  |                                     /policy-mgnt-auth-method-id
    |  +--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-mgnt-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        enumeration
    |     +--rw mutual-authentication        boolean
    |     +--rw token-server                 inet:ipv4-address
    |     +--rw certificate-server           inet:ipv4-address
    |     +--rw single-sing-on-server        inet:ipv4-address
    +--rw endpoint-group
    |  +--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?      inet:ipv4-address
    |  |  +--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?                  enumeration
    |  |  +--rw meta-data-server?            inet:ipv4-address
    |  |  +--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?                  enumeration
    |  |  +--rw meta-data-server?            inet:ipv4-address
    |  |  +--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?                  enumeration
    |  |  +--rw meta-data-server?            inet:ipv4-address
    |  |  +--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?                  enumeration
    |     +--rw meta-data-server?            inet:ipv4-address
    |     +--rw group-member?                string
    |     +--rw risk-level?                  uint16
    +--rw threat-feed
    |  +--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?                 inet:ipv4-address
    |  |  +--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?            inet:ipv4-address
    |  |  +--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?                 enumeration
       |  +--rw nsf-source?                  inet:ipv4-address
       |  +--rw nsf-credentials?             string
       |  +--rw collection-interval?         uint16
       |  +--rw collection-method?           enumeration
       |  +--rw heartbeat-interval?          uint16
       |  +--rw qos-marking?                 uint16
       +--rw telemetry-destination* [telemetry-destination-id]
          +--rw telemetry-destination-id     uint16
          +--rw name?                        string
          +--rw date?                        yang:date-and-time
          +--rw collector-source?            inet:ipv4-address
          +--rw collector-credentials?       string
          +--rw data-encoding?               string
          +--rw data-transport?              enumeration
```

Figure 1: Generic Data Model for Security Policies for cf Interface

4.1. YANG Data Model for Security Policies 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 "policy-general.yang"
module ietf-policy-general {
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-policy-general";
  prefix
    cf-interface;

  import ietf-yang-types{
    prefix yang;
  }

  import ietf-inet-types{
    prefix inet;
  }

  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 "2018-11-04"{
    description "fourth revision";
    reference
      "draft-kumar-i2nsf-client-facing-interface-im-04";
  }
```

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

    list rule {
     key "rule-id";
     leaf rule-id {
     type uint16;
     description
     "This is ID for rules.";
    }
      description
      "This is a container for rules.";
      leaf name {
       type string;
        description
         "This field idenfifies the name of this object.";
      }
      leaf date {
       type yang:date-and-time;
        description
         "Date this object was created or last
          modified";
      }
      leaf case {
       type string;
        description
         "to identify whether the rule belongs to
         web filter or enterprise mode.";
      }
      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 name {
        type string;
         description
```

```
          "This field idenfifies the name of this object.";
       }
       leaf date {
        type yang:date-and-time;
        description
         "Date this object was created or last
          modified";
       }
       leaf event-type {
        type string;
         description
          "This field identifies the event of
          policy enforcement trigger type.";
       }
       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-group {
        type leafref {
        path "/threat-feed/event-map-group/event-map-group-id";
        }
        description
        "This field contains security events or threat
        map in order to determine when a policy need
        to be activated. This is a reference to
        Evnet-Map-Group.";
       }
       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.";
      }
      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 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 match {
       type boolean;
        description
         "This field identifies the match criteria used to
         evaluate whether the specified action need to be
         taken or not.  This could be either a Policy-
         Endpoint-Group identifying a Application set or a
         set of traffic rules.";
      }
      leaf match-direction {
       type string;
        description
         "This field identifies if the match criteria is
         to evaluated for both direction of the traffic or
         only in one direction with default of allowing in
         the other direction for stateful match conditions.
         This is optional and by default rule should apply
         in both directions.";
      }
      leaf exception {
       type string;
        description
         "This field identifies the exception
         consideration when a rule is evaluated for a
         given communication.  This could be reference to
         Policy-Endpoint-Group object or set of traffic
```

```
         matching criteria.";
      }
     }
     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.";
      }
      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.";
```

```
      }
     }
    }
 }
     container multi-tenancy {
        description
          "The descriptions of multi-tenancy.";

        list policy-domain {
          key "policy-domain-id";

          leaf policy-domain-id {
            type uint16;
            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.";
          }
          list policy-tenant {
            key "policy-tenant-id";
            leaf policy-tenant-id {
              type uint16;
```

```
              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 leafref {
                path "/multi-tenancy/policy-domain/policy-domain-id";
              }
              description
              "This field identifies the domain to which this
              tenant belongs. This should be reference to a
              'Policy-Domain' object.";
            }
          }
          leaf authentication-method {
            type leafref {
              path "/multi-tenancy/policy-mgnt-auth-method/policy-mgnt-auth-
method-id";
            }
            description
            "Authentication method to be used for this domain.
            It should be a reference to a 'policy-mgmt-auth-method'
            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-mgnt-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 enumeration{
              enum password{
                description
                  "password-based authentication.";
              }
              enum token{
                description
                  "token-based authentication.";
              }
              enum certificate{
                description
                  "certificate-based authentication.";
              }
            }
            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 inet:ipv4-address;
            mandatory true;
            description
              "The token-server information if the
              authentication method is token-based";
          }
          leaf certificate-server {
            type inet:ipv4-address;
            mandatory true;
            description
```

```
              "The certificate-server information if
              the authentication method is certificate-based";
          }
          leaf single-sing-on-server {
            type inet:ipv4-address;
            mandatory true;
            description
              "The single-sign-on-server information
              if the authentication method is
              single-sign-on-based";
          }
        }
     }
container endpoint-group {
        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 inet:ipv4-address;
            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 enumeration{
              enum user-tag{
                description
```

```
                  "The user group is based on user-tag.";
              }
              enum user-name{
                description
                  "The user group is based on user-name.";
              }
              enum ip-address{
                description
                  "The user group is based on ip-address.";
              }
            }
            description
              "This describes the group type; User-tag,
              User-name or IP-address.";
          }
          leaf meta-data-server {
            type inet:ipv4-address;
            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 enumeration{
              enum device-tag{
                description
                  "The device group is based on device-tag.";
              }
              enum device-name{
                description
                  "The device group is based on device-name.";
              }
              enum ip-address{
                description
                  "The device group is based on ip-address.";
              }
            }
            description
              "This describes the group type; device-tag,
              device-name or IP-address.";
          }
          leaf meta-data-server {
            type inet:ipv4-address;
            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 enumeration{
              enum application-tag{
                description
                  "The application group is based on application-tag.";
              }
              enum device-name{
                description
                  "The application group is based on application-name.";
              }
              enum ip-address{
                description
                  "The application group is based on ip-address.";
              }
            }
            description
              "This identifies the group type;
              application-tag, application-name or
              IP-address.";
          }
          leaf meta-data-server {
            type inet:ipv4-address;
            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 enumeration{
              enum location-tag{
                description
                  "The location group is based on location-tag.";
              }
              enum location-name{
                description
                  "The location group is based on location-name.";
              }
              enum ip-address{
                description
                  "The location group is based on ip-address.";
              }
            }
            description
              "This identifies the group type;
```

```
              location-tag, location-name or
              IP-address.";
          }
          leaf meta-data-server {
            type inet:ipv4-address;
            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-feed {
      description
      "this describes the list of threat-feed.";

        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 inet:ipv4-address;
            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 mac-address {
                description
                  "list-type is MAC 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 inet:ipv4-address;
            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 enumeration {
            enum network-nsf {
              description
                "NSF telemetry source type is network-nsf.";
            }

            enum firewall-nsf {
              description
                "NSF telemetry source type is firewall-nsf.";
            }
            enum ids-nsf {
              description
                "NSF telemetry source type is ids-nsf.";
            }
            enum ips-nsf {
              description
                "NSF telemetry source type is ips-nsf.";
            }
            enum proxy-nsf {
              description
                "NSF telemetry source type is proxy-nsf.";
            }
            enum other-nsf {
              description
                "NSF telemetry source type is other-nsf.";
            }
          }
```

```
          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-source {
          type inet:ipv4-address;
           description
            "This field contains information such as
            IP address and protocol (UDP or TCP) port
            number of the NSF providing telemetry data.";
        }
        leaf nsf-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 uint16;
          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-source {
            type inet:ipv4-address;
              description
                "This field contains information such as
                IP address and protocol (UDP or TCP) port
```

```
                number for the collector's destination.";
          }
          leaf collector-credentials {
            type string;
              description
              "This field contains the 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 enumeration{
              enum grpc {
                description
                  "telemetry data protocol is grpc.";
              }
              enum buffer-over-udp{
                description
                  "telemetry data protocol is buffer over UDP.";
              }
            }
            description
            "This field contains streaming telemetry data
            protocols. This could be gRPC, protocol
            buffer over UDP, etc.";
          }
      }
    }
  }
<CODE ENDS>
```

Figure 2: YANG for policy-general

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

A common scenario for VoIP/VoLTE policy enforcement could be that a malicious call is made to a benign user of any telecommunication company. For example, imagine a case wherea company "A" employs a hacker with a malicious attempt to hack a user's phone with malware. The company "A" is located in a country, such as Africa, and uses the user's hacked phone to call the company. The hacked user is unaware of the company "A" so complains about the international call that was made to the company "B", which is the user's telecommunications company. The company "A" charges the company "B" for the

international call. The company "B" cannot charge the user for the call, and has no choice but to pay the company "A". 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: policy-voip
    +--rw policy-voip
    |  +--rw rule-voip* [rule-voip-id]
    |  |  +--rw rule-voip-id           uint16
    |  |  +--rw name?                  string
    |  |  +--rw date?                  yang:date-and-time
    |  |  +--rw event* [event-id]
    |  |  |  +--rw event-id            string
    |  |  |  +--rw name?               string
    |  |  |  +--rw date?               yang:date-and-time
    |  |  |  +--rw event-type?         string
    |  |  |  +--rw Time-Information?   string
    |  |  |  +--rw event-map-group?    -> /threat-feed/event-map-group
    |  |  |                               /event-map-group-id
    |  |  |  +--rw enable?             boolean
    |  |  +--rw condition* [condition-id]
    |  |  |  +--rw condition-id        string
    |  |  |  +--rw source-caller?      -> /threat-feed/threat-feed
    |  |  |                               /threat-feed-id
    |  |  |  +--rw destination-callee? -> /threat-feed/custom-list
    |  |  |                               /custom-list-id
    |  |  |  +--rw match?              boolean
    |  |  |  +--rw match-direction?    string
    |  |  |  +--rw exception?          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?            uint16
    |  +--rw owner* [owner-id]
    |     +--rw owner-id               string
    |     +--rw name?                  string
    |     +--rw date?                  yang:date-and-time
    +--rw threat-feed
       +--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?           inet:ipv4-address
|  +--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?      inet:ipv4-address
|  +--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
```

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

5.1. 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-cf-interface-voip.yang"

  module ietf-policy-voip {
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-policy-voip";
  prefix
    "cf-interface";

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

```
   import ietf-inet-types{
     prefix inet;
   }
   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 "2018-11-04"{
     description "sixth revision";
     reference
       "draft-kumar-i2nsf-client-facing-interface-im-07";
   }
   container policy-voip {
           description
             "This object is a policy instance to have
             complete information such as where and when
             a policy need to be applied.";
     list rule-voip {
           key "rule-voip-id";
           leaf rule-voip-id {
           type uint16;
           mandatory true;
           description
           "This is ID for rules.";
           }
           description
           "This is a container for rules.";
           leaf name {
             type string;
             description
               "This field idenfifies the name of this object.";
```

```
             }
             leaf date {
               type yang:date-and-time;
               description
                 "Date this object was created or last
                 modified";
             }
           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 name {
               type string;
               description
                 "This field idenfifies the name of this object.";
             }
             leaf date {
               type yang:date-and-time;
               description
                 "Date this object was created or last
                 modified";
             }
             leaf event-type {
               type string;
               description
                 "This field identifies the event event type
                 .";
             }
             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-group {
               type leafref{
                 path "/threat-feed/event-map-group/event-map-group-id";
             }
               description
```

```
               "This field contains security events or threat
               map in order to determine when a policy need
               to be activated. This is a reference to
               Evnet-Map-Group.";
             }
             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.";
             }
             leaf source-caller {
               type leafref {
                 path "/threat-feed/threat-feed/threat-feed-id";
               }
               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-callee {
               type leafref {
                 path "/threat-feed/custom-list/custom-list-id";
               }
               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 match {
               type boolean;
```

```
              description
                "This field identifies the match criteria used to
               evaluate whether the specified action need to be
               taken or not.  This could be either a Policy-
               Endpoint-Group identifying a Application set or a
               set of traffic rules.";
            }
            leaf match-direction {
              type string;
              description
                "This field identifies if the match criteria is
               to evaluated for both direction of the traffic or
               only in one direction with default of allowing in
               the other direction for stateful match conditions.
               This is optional and by default rule should apply
               in both directions.";
            }
            leaf exception {
              type string;
              description
                "This field identifies the exception
                consideration when a rule is evaluated for a
                given communication.  This could be reference to
                Policy-Endpoint-Group object or set of traffic
                matching criteria.";
            }
          }
          list action {
            key "action-id";
            leaf 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.";
            }
          }
          leaf precedence {
            type uint16;
              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 owner {
            key "owner-id";
            leaf owner-id {
            type string;
            mandatory true;
            description
              "this represents the owner-id.";
            }
            description
              "This field defines the owner of this policy.
              Only the owner is authorized to modify the
              contents of the policy.";
            leaf name {
              type string;
              description
                "The name of the owner.";
            }
            leaf date {
              type yang:date-and-time;
```

```
               description
                 "When the object was created or last
                 modified.";
             }
       }
     }
     container threat-feed {
         description
         "this describes the list of threat-feed.";

           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 inet:ipv4-address;
               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 mac-address {
                   description
                     "list-type is MAC 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 inet:ipv4-address;
               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.";
             }
         }
       }
   }
   <CODE ENDS>
```

Figure 4: Policy Instance YANG Example for VoIP Security Services

6. Example XML Output for Various Use Cases

In this section, we present an XML example for various use cases. Here, we show the policy examples that can be delivered through consumer-facing interface. For now, the considered use cases are: VoIP security service, DDoS-attack mitigation, time-based firewall, and web-filter.

6.1. Case 1: VoIP Security Service

The first example is a VoIP policy. Here, we are going to drop calls commin from a country with an Ip from South Africa that is classified as malicious. The below figure shows the XML document generated by using the YANG data tree as shown in the previous section.

```
<?xml version="1.1" encoding="UTF-8"?>
<rpc message-id="1" xmlns="urn:ietf:params:xml:ns:restconf:base:1.0">
```

```
 <edit-config>
  <target>
   <running/>
  </target>
   <config>
    <i2nsf-cf-interface-voip-req nc:operation="create">
        <policy-voip>
            <rule-voip>
                <rule-voip-id>01</rule-voip-id>
                <rule-voip-name>voip-policy-example</rule-voip-name>
                <rule-voip-date>2017.10.25/20:30:32</rule-voip-date>
                <event>
                    <event-id>01</event-id>
                    <event-name>voip_call</event-name>
                    <event-date>2017.10.25/20:30:32</event-date>
                    <event-type>malicious</event-type>
                    <event-map-group>19</event-map-group>
                    <enable>True</enable>
                </event>
                <condition>
                    <condition-id>01</condition-id>
                    <source-caller>105.176.0.0</source-caller>
                    <destination-callee>192.168.171.35</destination-callee>
                    <time-information>
                        <begin-time>22:00</begin-time>
                        <end-time>08:00</end-time>
                    </time-information>
                    <match-direction>default</match-direction>
                    <exeption>00</exeption>
                </condition>
                <action>
                    <action-id>01</action-id>
                    <action-name>action-voip</action-name>
                    <action-date>2017.10.25/20:30:32</action-date>
                    <primary-action>DENY</primary-action>
                    <secondary-action>LOG</secondary-action>
                </action>
                <precedence>none</precedence>
              <owner>
                <owner-id>01</owner-id>
                <name>i2nsf-admin</name>
              </owner>
            </rule-voip>
        </policy-voip>
    </i2nsf-cf-interface-voip-req>
   </config>
 </edit-config>
</rpc>
```

Figure 5: An XML Example for VoIP Security Service

6.2. Case 2: DDoS-Attack Mitigation

The second example is a DDoS-attack mitigation policy. Here, the time information is not set because the service provided by the network should be maintained at all times. If the packets sent by any sources are more than the set threshold, then the admin can set the percentage of the packets to be dropped to safely maintain the service.

```
<?xml version="1.1" encoding="UTF-8"?>
<rpc message-id="2" xmlns="urn:ietf:params:xml:ns:restconf:base:1.0">
 <edit-config>
  <target>
   <running/>
  </target>
   <config>
    <i2nsf-cf-interface-ddos-req nc:operation="create">
        <policy-ddos>
            <rule-ddos>
                <rule-ddos-id>03</rule-ddos-id>
                <rule-ddos-name>ddos-policy-example</rule-ddos-name>
                <rule-ddos-date>2018.10.25/11:25:32</rule-ddos-date>
                <event>
                    <event-id>03</event-id>
                    <event-name>ddos</event-name>
                    <event-date>2018.10.25/11:25:32</event-date>
                    <event-type>ddos</event-type>
                    <event-map-group>03</event-map-group>
                    <enable>True</enable>
                </event>
                <condition>
                    <condition-id>03</condition-id>
                    <source-ip>Any</source-ip>
                    <destination-ip>192.168.173.37</destination-ip>
                    <threshold>30</threshold>
                    <time-information>
                        <begin-time>--:--</begin-time>
                        <end-time>--:--</end-time>
                    </time-information>
                    <match-direction>default</match-direction>
                    <exeption>00</exeption>
                </condition>
                <action>
                    <action-id>03</action-id>
                    <action-name>action-ddos</action-name>
                    <action-date>2018.10.25/11:25:32</action-date>
```

```
                    <primary-action>REJECT</primary-action>
                    <secondary-action>LOG</secondary-action>
                </action>
                <precedence>none</precedence>
              <owner>
                <owner-id>03</owner-id>
                <name>i2nsf-admin</name>
              </owner>
            </rule-ddos>
        </policy-ddos>
    </i2nsf-cf-interface-ddos-req>
   </config>
 </edit-config>
</rpc>
```

Figure 6: An XML Example for DDoS-attack Mitigation

6.3. Case 3: Time-Based Firewall

The third example is a time-based firewall policy. Consider a Smart Factory which operates from 9 am to 7 pm during the working days. During these hours, only the admin responsible for operating the factory is allow to access a control system. The below figure show that any access during outside the operating hours is rejected.

```
<?xml version="1.1" encoding="UTF-8"?>
<rpc message-id="3" xmlns="urn:ietf:params:xml:ns:restconf:base:1.0">
 <edit-config>
  <target>
   <running/>
  </target>
   <config>
    <i2nsf-cf-interface-fw-req nc:operation="create">
        <policy-fw>
            <rule-fw>
                <rule-fw-id>01</rule-fw-id>
                <rule-fw-name>fw-policy-example</rule-fw-name>
                <rule-fw-date>2018.10.25/11:19:05</rule-fw-date>
                <event>
                    <event-id>01</event-id>
                    <event-name>invalid_access</event-name>
                    <event-date>2018.10.25/11:19:05</event-date>
                    <event-type>invalid</event-type>
                    <event-map-group>02</event-map-group>
                    <enable>True</enable>
                </event>
                <condition>
                    <condition-id>02</condition-id>
```

```
                       <source-ip>115.176.0.1</source-ip>
                       <destination-ip>192.168.173.41</destination-ip>
                       <time-information>
                           <begin-time>09:00</begin-time>
                           <end-time>17:00</end-time>
                       </time-information>
                       <match-direction>default</match-direction>
                       <exeption>00</exeption>
                   </condition>
                   <action>
                       <action-id>02</action-id>
                       <action-name>action-fw</action-name>
                       <action-date>2018.10.25/11:19:05</action-date>
                       <primary-action>PASS</primary-action>
                       <secondary-action>LOG</secondary-action>
                   </action>
                   <precedence>none</precedence>
                 <owner>
                   <owner-id>02</owner-id>
                   <name>i2nsf-admin</name>
                 </owner>
               </rule-fw>
           </policy-fw>
       </i2nsf-cf-interface-fw-req>
      </config>
    </edit-config>
   </rpc>
```

Figure 7: An XML Example for Time-based Firewall

6.4. Case 4: Time-Based Web-Filter

The last example is a time-based web-filter policy. Let us suppose that a owner of an enterprise wants to forbid access to a specific set of websites, such as Facebook, Youtube, Instagram, etc. The below figure shows an example policy an admin of a sector or department can deploy.

```
 <?xml version="1.1" encoding="UTF-8"?>
 <rpc message-id="4" xmlns="urn:ietf:params:xml:ns:restconf:base:1.0">
  <edit-config>
   <target>
    <running/>
   </target>
    <config>
     <i2nsf-cf-interface-wf-req nc:operation="create">
         <policy-wf>
             <rule-wf>
```

```
                 <rule-wf-id>03</rule-wf-id>
                 <rule-wf-name>wf-policy-example</rule-wf-name>
                 <rule-wf-date>2018.10.26/14:03:17</rule-wf-date>
                 <event>
                     <event-id>04</event-id>
                     <event-name>wf</event-name>
                     <event-date>2018.10.26/14:03:17</event-date>
                     <event-type>wf</event-type>
                     <event-map-group>04</event-map-group>
                     <enable>True</enable>
                 </event>
                 <condition>
                     <condition-id>04</condition-id>
                     <source-ip>192.168.1.3</source-ip>
                     <destination-url>www.facebook.com</destination-url>
                     <time-information>
                         <begin-time>09:00</begin-time>
                         <end-time>18:00</end-time>
                     </time-information>
                     <match-direction>default</match-direction>
                     <exeption>00</exeption>
                 </condition>
                 <action>
                     <action-id>04</action-id>
                     <action-name>action-wf</action-name>
                     <action-date>2018.10.26/14:03:17</action-date>
                     <primary-action>REJECT</primary-action>
                     <secondary-action>LOG</secondary-action>
                 </action>
                 <precedence>none</precedence>
               <owner>
                 <owner-id>03</owner-id>
                 <name>i2nsf-admin</name>
               </owner>
             </rule-wf>
         </policy-wf>
     </i2nsf-cf-interface-wf-req>
    </config>
  </edit-config>
 </rpc>
```

Figure 8: An XML Example for Time-based Web-filter

6.5. Case 5: Threat-Feed Configuration

The threat-feed container described above can convey various sources containing information concerning security threats. One good example can be STIX. STIX (Structured Threat Information Expression) is a

language and serialization format used to exchange cyber threat intelligence (CTI). It is a lanauge to describe threat information in a standardized format to enable exchanging and sharing them. The blow figure show the necessary configuration, which can be generated and delivered by consumer-facing interface.

```
   ...
   ...
     <configuration-tf>
         <threat-feed>
             <threat-feed-id>02</threat-feed-id>
             <threat-feed-name>stix</threat-feed-name>
             <threat-feed-date>2018.10.25/11:25:32</threat-feed-date>
             <threat-feed-type>ip-address</threat-feed-type>
             <feed-server>105.134.171.24</feed-server>
             <feed-priority>ip-address</feed-priority>
         </threat-feed>
     </configuration-tf>
 ...
 ...
```

Figure 9: An XML Example for Threat-feed Configuration

Usually, STIX can be obtained from a TAXII server which contains a collection of cyber threat information formatted in STIX. Here, the "feed-server" leaf contains the ip-address of the TAXII server, so that recent threat related information can be collected when the configuration is set.

7. 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.

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

[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-07 (work in progress), July 2018.

[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-05 (work in progress), May 2018.

[i2nsf-framework]
Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", RFC 8329, February 2018.

[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-06 (work in progress), July 2018.

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

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

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

- In Section 6, four additional XML output examples (VoIP, DDoS-attack, Time-based Firewall and Web-filter) for security policies are added. Also, an example XML output for Threat-feed configuration is added using STIX and TAXII as a threat-feed example.

- The overall organization of the YANG data model and its data types have also been reviewed and corrected, and produced the corresponding data tree as shown in the Sections 4 and 5.

- Overall editorial errors have been corrected.

Appendix B. Acknowledgments

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).

Appendix C. Contributors

This document is made by the group effort of I2NSF working group. Many people actively contributed to this document, such as Mahdi F. Dachmehchi and Daeyoung Hyun. The following are considered co-authors:

- Hyoungshick Kim (Sungkyunkwan University)

- Seungjin Lee (Sungkyunkwan University)

- Jinyong Tim Kim (Sungkyunkwan University)

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