Workgroup: I2NSF Working Group
Internet-Draft:
draft-ietf-i2nsf-consumer-facing-interface-dm-29
Published: 14 April 2023
Intended Status: Standards Track
Expires: 16 October 2023
Authors: J. Jeong, Ed. C. Chung
Sungkyunkwan University Sungkyunkwan University
T. Ahn R. Kumar S. Hares
Korea Telecom Juniper Networks Huawei

I2NSF Consumer-Facing Interface YANG Data Model

Abstract

This document describes a YANG data model of the Consumer-Facing Interface of the Security Controller in an Interface to Network Security Functions (I2NSF) system in a Network Functions Virtualization (NFV) environment. This document defines various types of managed objects and the relationship among them needed to build the flow policies from users' perspective. The YANG data model is based on the "Event-Condition-Action" (ECA) policy defined by a capability YANG data model for I2NSF. The YANG data model enables different users of a given I2NSF system to define, manage, and monitor flow policies within an administrative domain (e.g., user group).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In a framework of Interface to Network Security Functions (I2NSF) [RFC8329], each vendor can register their Network Security Functions (NSFs) using a Developer's Management System (DMS). Then the I2NSF User (e.g., an application for a security administrator such as a web application) can configure the NSFs by defining high-level security policies. Most vendors provide various proprietary applications or tools to define security policies for their own NSFs. The Consumer-Facing Interface is required because the applications developed by each vendor need to have a standard interface specifying the data types used when the I2NSF User and Security Controller (i.e., Network Operator Management System) communicate with each other using this interface. Therefore, this document specifies the required YANG data model such as their data types and encoding schemes so that high-level security policies (or configuration information for security policies) can be transferred to the Security Controller through the Consumer-Facing Interface. Security Controller will use the given information to translate the high-level security policies into the corresponding low-level security policies. The Security Controller delivers the translated security policies to the NSFs according to their respective security capabilities for the required security enforcement.

The Consumer-Facing Interface would be built using a set of objects, with each object capturing a unique set of information from an I2NSF User [RFC8329] needed to express a Security Policy. An object may have relationship with various other objects to express a complete set of requirements. The YANG data model in this document captures the managed objects and relationship among these objects. This model is structured in accordance with the "Event-Condition-Action" (ECA) policy.

An NSF Capability YANG data model is defined in [I-D.ietf-i2nsf-capability-data-model] as the basic model for both the NSF-Facing interface and Consumer-Facing Interface security policy model of this document.

```
                    +-----------------+
                    | Consumer-Facing |
                    |    Interface    |
                    +--------+--------+
                             ^
                             |
                             +-------------+------------+
                             |             |            |
                       +-----+----+  +-----+----+  +----+---+
                       |  Policy  |  | Endpoint |  | Threat |
                       |          |  |  groups  |  |  feed  |
                       +-----+----+  +----------+  +--------+
                             ^
                             |
                      +------+------+
                      |     Rule    |
                      +------+------+
                             ^
                             |
            +----------------+----------------+
            |                |                |
     +------+------+  +------+------+  +------+------+
     |    Event    |  |  Condition  |  |    Action   |
     +-------------+  +-------------+  +-------------+
```

Figure 1: Diagram for High-level Abstraction of Consumer-Facing Interface

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.

The efficient and flexible provisioning of network functions by a Network Functions Virtualization (NFV) system supports rapid deployment of newly developed functions. As practical applications, Network Security Functions (NSFs), such as firewall, Intrusion Detection System (IDS)/Intrusion Prevention System (IPS), and attack mitigation, can also be provided as Virtual Network Functions (VNFs) in the NFV system. By the efficient virtualization 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. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and

"OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the terminology described in [RFC8329].

This document follows the guidelines of [RFC8407], uses the common YANG types defined in [RFC6991], and adopts the Network Management Datastore Architecture (NMDA) [RFC8342]. The meaning of the symbols in tree diagrams is defined in [RFC8340].

3. YANG Tree Diagram of Policy

A Policy object is a means to express a Security Policy set by an I2NSF User with the Consumer-Facing Interface. It is sent to the Security Controller which converts it into an NSF-specific configuration via the NSF-Facing Interface for enforcement of the NSF. Figure 2 shows the YANG tree of the Policy object. The Policy object SHALL have the following information:

Name: This field identifies the name of this object.

Language: The language field indicates the language tag that is used for the natural language text that is included in all of the 'description' attributes. The language field is encoded following the rules in Section 2.1 of [RFC5646]. The default language tag is "en-US".

Resolution-strategy: This field represents how to resolve conflicts that occur between actions of the same or different policy rules that are matched and contained in this particular NSF. The resolution strategy is described in Section 3.2 of [I-D.ietf-i2nsf-capability-data-model] in detail. The default resolution strategy is "fmr" (First Matching Rule).

Rules: This field contains a list of rules. These rules are defined for implementing business requirements such as 1) supporting communication between two Endpoint Groups (see Section 4), 2) preventing communication with externally or internally identified threats, and 3) controlling access to internal or external resources for meeting regulatory compliance. An organization may restrict certain communication between a set of users and applications for example. The threats may be identified from threat feeds obtained from external sources. Note that rule conflict analysis should be performed by a monitoring service for policy rule conflicts in Security Controller to detect such rule conflicts among the policy rules installed into network security functions.

```
module: ietf-i2nsf-cons-facing-interface
  +--rw i2nsf-cfi-policy* [name]
  |  +--rw name                   string
  |  +--rw language?              string
  |  +--rw resolution-strategy?   identityref
  |  +--rw rules* [name]
  |     ...
  +--rw endpoint-groups
  |  ...
  +--rw threat-prevention
     ...
```

Figure 2: Policy YANG Data Tree

A policy contains a list of rules. In order to express a Rule, the Rule must have complete information such as where and when a policy needs to be applied. This is done by defining a set of managed objects and relationship among them. A Policy Rule defined in this module is a set of management guidelines that defines a desired behavior based on the Event-Condition-Action policy model (Section 3.1 of [I-D.ietf-i2nsf-capability-data-model]), but that is independent of a specific device and implementation. Figure 3 shows the YANG data tree of the Rule object. The rule object SHALL have the following information:

Name: This field identifies the name of this object.

Priority: This field identifies the priority of the rule.

Event: This field includes the information to determine whether the Rule Condition can be evaluated or not (see the definition of Event in Section 3.1 of [I-D.ietf-i2nsf-capability-data-model]). See details of the Event Object in Section 3.1.

Condition: This field contains a set of attributes, features, and/or values that are to be matched with the attributes of a packet or traffic flow to determine whether the Rule Action can be executed or not (see Section 3.1 of [I-D.ietf-i2nsf-capability-data-model]). See details of the Condition Object in Section 3.2.

Action: This field identifies the action taken when a rule is matched (see Section 3.1 of [I-D.ietf-i2nsf-capability-data-model]). There is always an implicit action to drop traffic if no rule is matched for a traffic type. See details of the Action Object in Section 3.3.

```
     +--rw rules* [name]
     |  +--rw name         string
     |  +--rw priority?    uint8
     |  +--rw event
     |  |  ...
     |  +--rw condition
     |  |  ...
     |  +--rw action
     |     ...
```

Figure 3: Rule YANG Data Tree

3.1. Event Sub-model

The Event Object contains information related to scheduling a Rule. The Event Object activates the evaluation of the Condition Object based on a security event (i.e., system event or system alarm). Note that an empty Event Object means that the event will always be evaluated as true and start the evaluation of the Condition Object. Figure 4 shows the YANG tree of the Event object. Event object SHALL have the following information:

System-event (also called alert): is defined as a warning about any changes of configuration, any access violation, the information of sessions and traffic flows.

System-alarm: is defined as a warning related to service degradation in system hardware.

```
  +--rw event
  |  +--rw system-event*   identityref
  |  +--rw system-alarm*   identityref
```

Figure 4: Event Sub-model YANG Data Tree

3.2. Condition Sub-model

The Condition object describes the network traffic pattern or fields that must be matched against the observed network traffic for the rule to trigger. The fields used to express the required conditions to trigger the rule are organized around the class of NSFs expected to be able to observe or compute them. Figure 5 shows the YANG tree of the Condition object. The Condition Sub-model SHALL have the following information:

firewall: This field represents the layer-2 header (e.g., MAC addresses), layer-3 header (e.g., IPv4 or IPv6 addresses, ICMPv4 or ICMPv6 parameters, and transport layer protocol) and layer-4 header (e.g., port numbers) of the network traffic. Note that the YANG module only provides high-level ICMP messages that are concretely specified by either ICMPv4 or ICMPv6 messages (e.g., Destination Unreachable: Port Unreachable which is ICMPv4's type 3 and code 3 or ICMPv6's type 1 and code 4). Also note that QUIC protocol [RFC9000] is excluded in the data model as it is not considered in the initial I2NSF documents [RFC8329]. The QUIC traffic should not be treated as UDP traffic. The data model should be extended or augmented appropriately to support the handling of QUIC traffic according to the needs of the implementer.

ddos: This field represents the threshold limit for the rate of the network traffic to mitigate a DDoS attack. The threshold configuration can be given in packet rate, byte rate, and flow rate. Definition of packet rate, byte rate, and flow rate are defined in Section 6 of [I-D.ietf-i2nsf-capability-data-model].

anti-virus: This field represents the configuration for an Antivirus service. A specific security profile can be added to Security Controller in order to update the configuration of the Antivirus service. Also, either a filename or path for such a profile can be configured for the Antivirus service.

payload: This field represents the payload information of the network traffic. The configuration is given in a high-level form that maps into the corresponding binary form registered with the Threat Prevention object (see Section 5.2).

url-category: This field represents the URL category to be filtered. The URLs can be categorized into a group with the URL-Group defined in Section 4.4, such as "sns-websites" for URLs that provide Social Networking Services (SNS). This information can be used to block or allow a certain URL or website.

voice: This field contains the call source-id, call destination-id, and user-agent. This information describes a caller identification or receiver identification in order to prevent any exploits or attacks (e.g., voice phishing) of Voice over IP (VoIP) or Voice over Cellular Network (VoCN). Note that VoCN can be either Voice over LTE (VoLTE) [TR-29.949-3GPP] or Voice over 5G (Vo5G) [TR-21.915-3GPP].

context: This field represents the extra information for the condition such as time, application, device type, user condition, and geographic location (see Section 5.1 of [I-D.ietf-i2nsf-capability-data-model]).

threat-feed:
This field contains the information obtained from threat-feeds. This field is used when security rule condition is based on the existing threat reports gathered from other sources.

Note that due to the exclusion of QUIC protocol in the I2NSF documents, HTTP/3 is also excluded in the document along with the QUIC protocol. HTTP/3 should neither be interpreted as HTTP/1.1 nor HTTP/2. The data model should be extended or augmented appropriately to support the handling of HTTP/3 traffic according to the needs of the implementer.

Note that the identities for ICMP messages provided in the YANG module are combined for ICMPv4 and ICMPv6 such as echo/echo-reply for ICMPv4 and echo-request/echo-reply for ICMPv6. For more information about the comparison between ICMPv4 and ICMPv6 messages, refer to [IANA-ICMP-Parameters] and [IANA-ICMPv6-Parameters].

```
+--rw condition
|  +--rw firewall
|  |  +--rw source*                     union
|  |  +--rw destination*                union
|  |  +--rw transport-layer-protocol?   identityref
|  |  +--rw range-port-number* [start end]
|  |  |  +--rw start    inet:port-number
|  |  |  +--rw end      inet:port-number
|  |  +--rw icmp
|  |     +--rw message*   identityref
|  +--rw ddos
|  |  +--rw rate-limit
|  |     +--rw packet-rate-threshold?   uint64
|  |     +--rw byte-rate-threshold?     uint64
|  |     +--rw flow-rate-threshold?     uint64
|  +--rw anti-virus
|  |  +--rw profile*   string
|  |  +--rw exception-files*   string
|  +--rw payload
|  |  +--rw content*   -> /threat-prevention/payload-content/name
|  +--rw url-category
|  |  +--rw url-name?   -> /endpoint-groups/url-group/name
|  +--rw voice
|  |  +--rw source-id*        -> /endpoint-groups/voice-group/name
|  |  +--rw destination-id*   -> /endpoint-groups/voice-group/name
|  |  +--rw user-agent*       string
|  +--rw context
|  |  +--rw time
|  |  |  +--rw start-date-time?   yang:date-and-time
|  |  |  +--rw end-date-time?     yang:date-and-time
|  |  |  +--rw period
|  |  |  |  +--rw start-time?   time
|  |  |  |  +--rw end-time?     time
|  |  |  |  +--rw day*          day
|  |  |  |  +--rw date*         int8
|  |  |  |  +--rw month* [start end]
|  |  |  |     +--rw start    string
|  |  |  |     +--rw end      string
|  |  |  +--rw frequency?         enumeration
|  |  +--rw application
|  |  |  +--rw protocol*   identityref
|  |  +--rw device-type
|  |  |  +--rw device*   identityref
|  |  +--rw users
|  |  |  +--rw user* [id]
|  |  |  |  +--rw id      uint32
|  |  |  |  +--rw name?   string
|  |  |  +--rw group* [id]
|  |  |     +--rw id      uint32
|  |  |     +--rw name?   string
|  |  +--rw geographic-location
|  |     +--rw source
|  |     |  +--rw country?   -> /endpoint-groups/location-group/country
|  |     |  +--rw region?    -> /endpoint-groups/location-group/region
|  |     |  +--rw city?      -> /endpoint-groups/location-group/city
|  |     +--rw destination
|  |        +--rw country?   -> /endpoint-groups/location-group/country
|  |        +--rw region?    -> /endpoint-groups/location-group/region
|  |        +--rw city?      -> /endpoint-groups/location-group/city
|  +--rw threat-feed
|     +--rw name*   -> /threat-prevention/threat-feed-list/name
```

Figure 5: Condition Sub-model YANG Data Tree

3.3. Action Sub-model

This object represents actions that Security Admin wants to perform based on certain traffic class. Figure 6 shows the YANG tree of the Action object. The Action object SHALL have the following information:

Primary-action: This field identifies the action when a rule is matched by an NSF. The action could be one of "pass", "drop", "reject", "rate-limit", "mirror", "invoke-signaling", "tunnel-encapsulation", "forward", and "transform". This action is related to the ingress-action-capability and egress-action-capability in [I-D.ietf-i2nsf-capability-data-model]. Note that if the action is "rate-limit", the limit value should be given to Security Controller in order to determine the threshold of the traffic rate.

Secondary-action: This field identifies the action when a rule is matched by an NSF. The action could be one of "rule-log" and "session-log". This action is related to the log-action in [I-D.ietf-i2nsf-capability-data-model].

```
+--rw action
   +--rw primary-action
   |  +--rw action    identityref
   |  +--rw limit?    decimal64
   +--rw secondary-action
      +--rw log-action?   identityref
```

Figure 6: Action Sub-model YANG Data Tree

4. YANG Tree Diagram of Policy Endpoint Groups

The Policy Endpoint Group is the collection of network nodes that are labeled and placed together into a group. As shown in Figure 7, endpoint groups include User-Group (Section 4.1), Device-Group (Section 4.2), Location-Group (Section 4.3), URL-Group (Section 4.4), and Voice-Group (Section 4.5). An I2NSF User can create and use these objects to represent a logical entity in their business environment, where a security policy is to be applied. Figure 8 shows the YANG tree of the Endpoint-Groups object.

The endpoint group information delivered by the I2NSF User should be stored into a secure database available to the Security Controller for the translation from a high-level security policy to the corresponding low-level security policy. The information should be synchronized with other systems in real-time for accurate translation.

```
                           +-------------------+
                           |  Endpoint Groups  |
                           +---------+---------+
                                     ^
                                     |
       +-------------+---------------+--------------+-----------+
  0..n |        0..n |          0..n |         0..n |      0..n |
 +-----+----+ +------+-----+ +-------+------+ +-----+---+ +-----+-----+
 |User-group| |Device-group| |Location-group| |URL-group| |Voice-group|
 +----------+ +------------+ +--------------+ +---------+ +-----------+
```

Figure 7: Endpoint Group Diagram

```
  +--rw endpoint-groups
  |  +--rw user-group* [name]
  |  |  ...
  |  +--rw device-group* [name]
  |  |  ...
  |  +--rw location-group* [country region city]
  |  |  ...
  |  +--rw url-group* [name]
  |  |  ...
  |  +--rw voice-group* [name]
  |     ...
```

Figure 8: Endpoint Group YANG Data Tree

4.1. User-Group

The User-Group object represents the MAC addresses and IP (IPv4 or IPv6) addresses that are labeled as a group of users (e.g., employees). Figure 9 shows the YANG tree of the User-Group object. The User-Group object SHALL have the following information:

Name: This field identifies the name of the user-group.

mac-address: This represents the MAC address(es) for the user-group.

ipv4: This represents the IPv4 addresses as an IPv4 prefix or IPv4 address range for the user-group.

ipv6: This represents the IPv6 addresses as an IPv6 prefix or IPv6 address range for the user-group.

```
+--rw user-group* [name]
|  +--rw name                              string
|  +--rw mac-address*                      yang:mac-address
|  +--rw (match-type)
|     +--:(ipv4)
|     |  +--rw (ipv4-range-or-prefix)?
|     |     +--:(prefix)
|     |     |  +--rw ipv4-prefix*          inet:ipv4-prefix
|     |     +--:(range)
|     |        +--rw range-ipv4-address* [start end]
|     |           +--rw start    inet:ipv4-address-no-zone
|     |           +--rw end      inet:ipv4-address-no-zone
|     +--:(ipv6)
|        +--rw (ipv6-range-or-prefix)?
|           +--:(prefix)
|           |  +--rw ipv6-prefix*          inet:ipv6-prefix
|           +--:(range)
|              +--rw range-ipv6-address* [start end]
|                 +--rw start    inet:ipv6-address-no-zone
|                 +--rw end      inet:ipv6-address-no-zone
```

Figure 9: User-Group YANG Data Tree

4.2. Device-Group

The Device-Group object represents the labeled network devices that provide services (e.g., servers) hosted on the IP (IPv4 or IPv6) addresses and application protocol. Figure 10 shows the YANG tree of the Device-group object. The Device-Group object SHALL have the following information:

Name: This field identifies the name of this object.

ipv4: This represents the IPv4 addresses as an IPv4 prefix or IPv4 address range for the device-group.

ipv6: This represents the IPv6 addresses as an IPv6 prefix or IPv6 address range for the device-group.

Application-protocol: This represents the application layer protocols of devices for the device-group.

```
+--rw device-group* [name]
|  +--rw name                              string
|  +--rw (match-type)
|  |  +--:(ipv4)
|  |  |  +--rw (ipv4-range-or-prefix)?
|  |  |     +--:(prefix)
|  |  |     |  +--rw ipv4-prefix*          inet:ipv4-prefix
|  |  |     +--:(range)
|  |  |        +--rw range-ipv4-address* [start end]
|  |  |           +--rw start    inet:ipv4-address-no-zone
|  |  |           +--rw end      inet:ipv4-address-no-zone
|  |  +--:(ipv6)
|  |     +--rw (ipv6-range-or-prefix)?
|  |        +--:(prefix)
|  |        |  +--rw ipv6-prefix*          inet:ipv6-prefix
|  |        +--:(range)
|  |           +--rw range-ipv6-address* [start end]
|  |              +--rw start    inet:ipv6-address-no-zone
|  |              +--rw end      inet:ipv6-address-no-zone
|  +--rw application-protocol*             identityref
```

Figure 10: Device-Group YANG Data Tree

4.3. Location-Group

The Location-Group object represents the IP (IPv4 or IPv6) addresses labeled as a geographic location (i.e., country, region, and city). Figure 11 shows the YANG tree of the Location-Group object. The Location-Group object SHALL have the following information:

Country: This field represents the 2-letter ISO country code conforming to ISO3166-1 alpha 2, e.g., 'US' for United States, 'JP' for Japan, and 'PL' for Poland.

Region: This field represents the region code conforming to ISO 3166-2. Examples include 'ID-RI' for Riau province of Indonesia and 'NG-RI' for the Rivers province in Nigeria.

City: This field represents the city of a region, e.g., 'Dublin', 'New York', and 'Sao Paulo'.

ipv4: This represents the IPv4 addresses as an IPv4 prefix or IPv4 address range for the location-group.

ipv6: This represents the IPv6 addresses as an IPv6 prefix or IPv6 address range for the location-group.

```
+--rw location-group* [country region city]
|  +--rw country                           string
|  +--rw region                            string
|  +--rw city                              string
|  +--rw (match-type)
|     +--:(ipv4)
|     |  +--rw (ipv4-range-or-prefix)?
|     |     +--:(prefix)
|     |     |  +--rw ipv4-prefix*          inet:ipv4-prefix
|     |     +--:(range)
|     |        +--rw range-ipv4-address* [start end]
|     |           +--rw start    inet:ipv4-address-no-zone
|     |           +--rw end      inet:ipv4-address-no-zone
|     +--:(ipv6)
|        +--rw (ipv6-range-or-prefix)?
|           +--:(prefix)
|           |  +--rw ipv6-prefix*          inet:ipv6-prefix
|           +--:(range)
|              +--rw range-ipv6-address* [start end]
|                 +--rw start    inet:ipv6-address-no-zone
|                 +--rw end      inet:ipv6-address-no-zone
```

Figure 11: Location-Group YANG Data Tree

4.4. URL-Group

The URL-Group object represents the collection of Uniform Resource Locators (URLs) or hostnames labeled into a group (e.g., sns-websites). Figure 12 shows the YANG tree of the URL-Group object. The URL-Group object SHALL have the following information:

Name: This field identifies the name of this object.

URL: This field represents the URL.

```
+--rw url-group* [name]
|  +--rw name    string
|  +--rw url*    inet:uri
```

Figure 12: URL-Group YANG Data Tree

4.5. Voice-Group

The Voice-Group object represents the collection of Session Initiation Protocol (SIP) identities labeled into a group. Figure 13 shows the YANG tree of the Voice-Group object. The Voice-Group object SHALL have the following information:

Name:
This field identifies the name of this object.

SIP-id: This field represents the SIP identities in SIP URI scheme (Section 19.1.1 of [RFC3261]).

```
  +--rw voice-group* [name]
     +--rw name      string
     +--rw sip-id*   inet:uri
```

Figure 13: Voice-Group YANG Data Tree

5. YANG Tree Diagram of Threat Prevention

The Threat Prevention model describes information obtained from threat feeds (i.e., sources for obtaining the threat information). The presented information contains the features or attributes that identify a well-known threat (e.g., signatures or payload) to prevent malicious activity entering the secured network. There are multiple managed objects that constitute this category as shown in Figure 14. Figure 15 shows the YANG tree of a Threat-Prevention object.

```
                   +-------------------+
                   | Threat Prevention |
                   +---------+---------+
                             ^
                             |
                   +---------+---------+
           0..n    |         0..n      |
            +------+------+   +--------+--------+
            | Threat-feed |   | Payload-content |
            +-------------+   +-----------------+
```

Figure 14: Threat Prevention Diagram

```
  +--rw threat-prevention
     +--rw threat-feed-list* [name]
     |  ...
     +--rw payload-content* [name]
        ...
```

Figure 15: Threat Prevention YANG Data Tree

5.1. Threat Feed

This object represents a threat feed which provides the signatures of malicious activities. Figure 16 shows the YANG tree of a Threat-feed-list. The Threat-Feed object SHALL have the following information:

Name: This field identifies the name of this object.

IOC: This field represents the Indicators of Compromise (IOC), i.e., the critical information of patterns or characteristics in the threat feed that identifies malicious activities. The format of the information given in this field is based on the format field (e.g., STIX, MISP, OpenIOC, and IODEF).

Format: This field represents the format or structure of the IOC field for the threat-feed such as Structured Threat Information Expression (STIX) [STIX], MISP Core [MISPCORE], OpenIOC [OPENIOC], and Incident Object Description Exchange Format (IODEF) [RFC8727]. This can be extended depending on the implementation of the existing threat-feed.

It is assumed that the I2NSF User obtains the threat signatures (i.e., threat content patterns) from a threat-feed server (i.e., feed provider), which is a server providing threat signatures. With the obtained threat signatures, the I2NSF User can deliver them to the Security Controller via the Consumer-Facing Interface. The retrieval of the threat signatures by the I2NSF User is out of the scope of this document.

Note that the information of a threat feed (i.e., a pair of IOC and Format) is used as information to alert or block traffic that matches IOCs identified in the threat feed. This information is used to update the NSFs that have various content security control capabilities (e.g., IPS, URL-Filtering, Antivirus, and VoIP/VoCN Filter) derived in [I-D.ietf-i2nsf-capability-data-model]. Those capabilities derive specific content security controls such as signature-set, exception-signature, and detect.

It is noted that DDoS Open Threat Signaling (dots) can be used to collect threat feeds in the form of signatures [RFC8811].

```
+--rw threat-feed-list* [name]
|  +--rw name       string
|  +--rw ioc*       string
|  +--rw format     identityref
```

Figure 16: Threat Feed YANG Data Tree

5.2. Payload Content

This object represents a list of raw binary patterns of a packet payload content (i.e., data after a transport layer header) to describe a threat. Figure 17 shows the YANG tree of a Payload-content list. The Payload-content object SHALL have the following information:

Name: This field identifies the name of this object. It is recommended to use short and simple words that describe the content. For example, the name "backdoor" indicates the payload content is related to a backdoor attack.

Description: This represents the description to further describe the content field in detail. This field is not mandatory, but it is recommended to use this field as it is helpful for future usage.

Content: This represents the payload content patterns (i.e., data after a transport layer header), which are involved in a security attack, in binary. If multiple instances of contents are defined, all defined contents must be matched somewhere in the session stream. The content pattern should be matched based on the order given by the user. The scope of the payload to be matched can be defined by the depth and offset/distance fields.

Depth: This field specifies how far a packet should be searched for the specified content pattern defined in the content field. If this field is undefined, then the content pattern should be searched within the whole payload.

Starting-point: This field specifies the starting point of matching the content pattern to the payload. If this field is undefined, then the content pattern should be searched from the beginning of the payload. The starting point can be defined by either the offset value or distance value. The offset keyword specifies where to start searching for the specified content pattern. The offset is calculated from the beginning of the payload. The distance keyword specifies how far a payload should be ignored before starting to search for the specified content pattern relative to the end of the previous specified content pattern match. This can be thought of as exactly the same thing as offset, except it is relative to the end of the last pattern match instead of the beginning of the packet. Note that this field cannot be used if the content is the first order of the list.

```
  +--rw payload-content* [name]
     +--rw name           string
     +--rw description?   string
     +--rw contents* [content]
        +--rw content           binary
        +--rw depth?            uint16
        +--rw (starting-point)?
           +--:(offset)
           |  +--rw offset?     int32
           +--:(distance)
              +--rw distance?   int32
```

Figure 17: Payload Content in YANG Data Tree

6. YANG Data Model of Consumer-Facing Interface

The main objective of this document is to provide the YANG data model of the I2NSF Consumer-Facing Interface. This interface can be used to deliver control and management messages between an I2NSF User and Security Controller for the I2NSF User's high-level security policies.

The semantics of the data model is aligned with the information model of the Consumer-Facing Interface. 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.

With the YANG data model of I2NSF Consumer-Facing Interface, this document provides examples for security policy rules such as time-based firewall, VoIP/VoCN security service, and DDoS-attack mitigation in Section 7.

6.1. YANG Module of Consumer-Facing Interface

This section describes a YANG module of Consumer-Facing Interface. This document provides identities in the data model to be used for configuration of an NSF. Each identity is used for a different type of configuration. The details are explained in the description of each identity. This YANG module imports from [RFC6991] and [I-D.ietf-i2nsf-nsf-monitoring-data-model]. It makes references to [RFC0768] [RFC0792] [RFC0854] [RFC0959] [RFC1939] [RFC2595] [RFC3022] [RFC3261] [RFC3986] [RFC4250] [RFC4340] [RFC4443] [RFC5321] [RFC5646] [RFC8075] [RFC8335] [RFC8727] [RFC9051] [RFC9110] [RFC9112] [RFC9113] [RFC9260] [RFC9293] [GLOB] [IANA-ICMP-Parameters] [IANA-ICMPv6-Parameters] [ISO-3166-1alpha2] [ISO-3166-2] [I-D.ietf-i2nsf-capability-data-model] [MISPCORE] [OPENIOC] [STIX]

```
<CODE BEGINS> file "ietf-i2nsf-cons-facing-interface@2023-04-14.yang"

module ietf-i2nsf-cons-facing-interface {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-cons-facing-interface";
  prefix
    i2nsfcfi;

  import ietf-inet-types {
    prefix inet;
    reference "RFC 6991";
  }

  import ietf-yang-types {
    prefix yang;
    reference "RFC 6991";
  }

  import ietf-i2nsf-monitoring-interface {
    prefix i2nsfmi;
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-20";
      // RFC Ed.: replace with an actual RFC number and remove
      // this note.
  }

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

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

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

     Editor: Patrick Lingga
     <mailto:patricklink@skku.edu>";

  description
    "This module is a YANG module for Consumer-Facing Interface.

     The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
     'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',
     'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this
     document are to be interpreted as described in BCP 14
     (RFC 2119) (RFC 8174) when, and only when, they appear
     in all capitals, as shown here.
     Copyright (c) 2023 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Revised BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX
     (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
     for full legal notices.";

  // RFC Ed.: replace XXXX with an actual RFC number and remove
  // this note.

  revision "2023-04-14" {
    description "Initial revision.";
    reference
      "RFC XXXX: I2NSF Consumer-Facing Interface YANG Data Model";

    // RFC Ed.: replace XXXX with an actual RFC number and remove
    // this note.
  }

  identity resolution-strategy {
    description
      "Base identity for resolution strategy.";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Resolution Strategy";
  }

  identity fmr {
    base resolution-strategy;
    description
      "Conflict resolution with First Matching Rule (FMR).";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Resolution Strategy";
  }

  identity lmr {
    base resolution-strategy;
    description
      "Conflict resolution with Last Matching Rule (LMR).";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Resolution Strategy";
  }

  identity pmre {
    base resolution-strategy;
    description
      "Conflict resolution with Prioritized Matching Rule with
       Errors (PMRE).";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Resolution Strategy";
  }

  identity pmrn {
    base resolution-strategy;
    description
      "Conflict resolution with Prioritized Matching Rule with
       No Errors (PMRN).";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Resolution Strategy";
  }

  identity action {
    description
      "Base identity for action.";
  }

  identity primary-action {
    base action;
    description
      "Base identity for primary action. Primary action is an action
       that handles the forwarding of the packets or flows in an
       NSF.";
  }

  identity secondary-action {
    base action;
    description
      "Base identity for secondary action. Secondary action is an
       action in the background that does not affect the network,
       such as logging.";
  }

  identity ingress-action {
    base primary-action;
    description
      "Base identity for ingress action. This action is to handle the
       network traffic that is entering the secured network.";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Ingress Action";
  }

  identity egress-action {
    base primary-action;
    description
      "Base identity for egress action. This action is to handle the
       network traffic that is exiting the secured network.";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Egress Action";
  }

  identity pass {
    base ingress-action;
    base egress-action;
    description
      "The pass action allows traffic that matches
       the rule to proceed through the NSF to reach the
       destination.";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Actions and
       Default Action";
  }

  identity drop {
    base ingress-action;
    base egress-action;
    description
      "The drop action denies the traffic that
       matches the rule. The drop action should do a silent drop,
       which does not give any response to the source.";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Actions and
       Default Action";
  }

  identity reject {
    base ingress-action;
    base egress-action;
    description
      "The reject action denies a packet to go through the NSF
       entering or exiting the internal network and sends a response
       back to the source. The response depends on the packet and
       implementation. For example, a packet may be rejected with
       an ICMPv4 Type 3 Code 13 or ICMPv6 Type 1 Code 1 reply message
       (i.e., Destination Unreachable: Communication Administratively
       Prohibited) by an administrative purpose (e.g., firewall
       filter).";
  }

  identity mirror {
    base ingress-action;
    base egress-action;
    description
      "The mirror action copies a packet and sends the packet's copy
       to the monitoring entity while still allowing the packet or
       flow to go through the NSF.";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Actions and
       Default Action";
  }

  identity rate-limit {
    base ingress-action;
    base egress-action;
    description
      "The rate limit action limits the number of packets or flows
       that can go through the NSF by dropping packets or flows
       (randomly or systematically). The drop mechanism, e.g., silent
       drop and unreachable drop (i.e., reject), is up to the
       implementation.";
    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Actions and
       Default Action";
  }

  identity invoke-signaling {
    base egress-action;
    description
      "The invoke-signaling action is used to convey information of
       the event triggering this action to a monitoring entity.";
  }

  identity tunnel-encapsulation {
    base egress-action;
    description
      "The tunnel encapsulation action is used to encapsulate the
       packet to be tunneled across the network to enable a secure
       connection.";
  }
```

```
     identity forwarding {
       base egress-action;
       description
         "The forwarding action is used to relay the packet from one
          network segment to another node in the network.";
     }

     identity transformation {
       base egress-action;
       description
         "The transformation action is used to transform a packet by
          modifying it (e.g., HTTP-to-CoAP packet translation).
          Note that a subset of transformation (e.g., HTTP-to-CoAP) is
          handled in this YANG module, rather than all the existing
          transformations.  Specific algorithmic transformations can be
          executed by a middlebox (e.g., NSF) for a given transformation
          name.";
       reference
         "RFC 8075: Guidelines for Mapping Implementations: HTTP to the
          Constrained Application Protocol (CoAP) - Translation between
          HTTP and CoAP.";
     }

     identity log-action {
       base secondary-action;
       description
         "Base identity for log action.";
     }

     identity rule-log {
       base log-action;
       description
         "Log the policy rule that has been triggered by a packet or
          flow.";
     }

     identity session-log {
       base log-action;
       description
         "A session is a connection (i.e., traffic flow) of a data plane
          that includes source and destination information of IP
          addresses and transport port numbers with the protocol used.
          Log the session that triggered a policy rule.";
     }

     identity icmp-message {
       description
         "Base identity for ICMP Message types. Note that this YANG
          module only provides ICMP messages that are shared between
          ICMPv4 and ICMPv6 (e.g., Destination Unreachable: Port
          Unreachable which is ICMPv4 type 3 code 3 or ICMPv6 type 1
          code 4).";
       reference
         "RFC  792: Internet Control Message Protocol
          RFC 8335: PROBE: A Utility for Probing Interfaces
          IANA: Internet Control Message Protocol (ICMP)
          Parameters
          IANA: Internet Control Message Protocol version 6
          (ICMPv6) Parameters";
     }

     identity echo-reply {
       base icmp-message;
       description
         "Identity for 'Echo Reply' ICMP message type 0 in ICMPv4 or
          type 129 in ICMPv6.";
     }

     identity destination-unreachable {
       base icmp-message;
       description
         "Identity for 'Destination Unreachable' ICMP message type 3 in
          ICMPv4 or type 1 in ICMPv6.";
     }

     identity redirect {
       base icmp-message;
       description
         "Identity for 'Redirect' ICMP message type 5 in ICMPv4
          or type 137 in ICMPv6.";
     }

     identity echo {
       base icmp-message;
       description
         "Identity for 'Echo' ICMP message type 8 in ICMPv4 or type 128
          in ICMPv6.";
     }

     identity router-advertisement {
       base icmp-message;
       description
         "Identity for 'Router Advertisement' ICMP message type 9 in
          ICMPv4 or type 134 in ICMPv6.";
     }

     identity router-solicitation {
       base icmp-message;
```

```
  description
    "Identity for 'Router Solicitation' ICMP message type 10 in
     ICMPv4 or type 135 in ICMPv6.";
}

identity time-exceeded {
  base icmp-message;
  description
    "Identity for 'Time exceeded' ICMP message type 11 in ICMPv4
     or type 3 in ICMPv6.";
}

identity parameter-problem {
  base icmp-message;
  description
    "Identity for 'Parameter Problem' ICMP message type 12 in
     ICMPv4 or type 4 in ICMPv6.";
}

identity experimental-mobility-protocols {
  base icmp-message;
  description
    "Identity for 'Experimental Mobility Protocols' ICMP message
     type 41 in ICMPv4 or type 150 in ICMPv6.";
}

identity extended-echo-request {
  base icmp-message;
  description
    "Identity for 'Extended Echo Request' ICMP message type 42
     in ICMPv4 or type 160 in ICMPv6.";
}

identity extended-echo-reply {
  base icmp-message;
  description
    "Identity for 'Extended Echo Reply' ICMP message type 43 in
     ICMPv4 or type 161 in ICMPv6.";
}

identity port-unreachable {
  base destination-unreachable;
  description
    "Identity for port unreachable in destination unreachable
     message (i.e., ICMPv4 type 3 code 3 or ICMPv6 type 1 code 4).";
}

identity request-no-error {
  base extended-echo-request;
  description
    "Identity for request with no error in extended echo request
     message (i.e., ICMPv4 type 42 code 0 or ICMPv6 type 160
     code 0).";
}

identity reply-no-error {
  base extended-echo-reply;
  description
    "Identity for reply with no error in extended echo reply
     message (i.e., ICMPv4 type 43 code 0 or ICMPv6 type 161
     code 0).";
}

identity malformed-query {
  base extended-echo-reply;
  description
    "Identity for malformed query in extended echo reply message
     (i.e., ICMPv4 type 43 code 1 or ICMPv6 type 161 code 1).";
}

identity no-such-interface {
  base extended-echo-reply;
  description
    "Identity for no such interface in extended echo reply message
     (i.e., ICMPv4 type 43 code 2 or ICMPv6 type 161 code 2).";
}

identity no-such-table-entry {
  base extended-echo-reply;
  description
    "Identity for no such table entry in extended echo reply
     message (i.e., ICMPv4 type 43 code 3 or ICMPv6 type 161
     code 3).";
}

identity multiple-interfaces-satisfy-query {
  base extended-echo-reply;
  description
    "Identity for multiple interfaces satisfy query in extended
     echo reply message (i.e., ICMPv4 type 43 code 4 or ICMPv6
     type 161 code 4).";
  reference
    "RFC  792: Internet Control Message Protocol
     RFC 8335: PROBE: A Utility for Probing Interfaces";
}

identity ioc-format {
  description
```

```
    "This represents the base identity for the format of the
     Indicators of Compromise (IOC).";
}

identity stix {
  base ioc-format;
  description
    "This represents the Structured Threat Information Expression
     (STIX) format in JSON.";
  reference
    "STIX: Structured Threat Information Expression version 2.1 - JSON
     format";
}

identity misp {
  base ioc-format;
  description
    "This represents the Malware Information Sharing Platform (MISP)
     Core format.";
  reference
    "MISPCORE: Malware Information Sharing Platform (MISP) Core
     Format";
}

identity openioc {
  base ioc-format;
  description
    "This represents the OpenIOC format.";
  reference
    "OPENIOC: OpenIOC 1.1 Schema document";
}

identity iodef {
  base ioc-format;
  description
    "This represents the Incident Object Description Exchange Format
     (IODEF) format.";
  reference
    "RFC 8727: JSON Binding of the Incident Object Description
     Exchange Format";
}

identity device-type {
  description
    "Base identity for types of device. This identity is used for
     type of the device for the source or destination of a packet
     or traffic flow.";
}
```

```
identity computer {
  base device-type;
  description
    "Identity for computer such as personal computer (PC)
     and server.";
}

identity mobile-phone {
  base device-type;
  description
    "Identity for mobile-phone such as smartphone and
     cellphone.";
}

identity voip-vocn-phone {
  base device-type;
  description
    "Identity for VoIP (Voice over Internet Protocol) or VoCN
     (Voice over Cellular Network, such as Voice over LTE or 5G)
     phone.";
}

identity tablet {
  base device-type;
  description
    "Identity for tablet devices.";
}

identity network-infrastructure-device {
  base device-type;
  description
    "Identity for network infrastructure devices
     such as switch, router, and access point";
}

identity iot-device {
  base device-type;
  description
    "Identity for Internet of Things (IoT) devices
     such as sensors, actuators, and low-power
     low-capacity computing devices.";
}

identity ot {
  base device-type;
  description
    "Identity for Operational Technology (OT) devices (also
     known as industrial control systems) that interact
     with the physical environment and detect or cause direct
```

```
      change through the monitoring and control of devices,
      processes, and events such as programmable logic
      controllers (PLCs), digital oscilloscopes, building
      management systems (BMS), and fire control systems.";
 }

 identity vehicle {
   base device-type;
   description
     "Identity for transportation vehicles that connect to and
      share data through the Internet over Vehicle-to-Everything
      (V2X) communications.";
 }

/*
 * Typedefs
 */

 typedef time {
   type string {
     pattern '(0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9](\.[0-9]+)?'
           + '(Z|[\+\-]((1[0-3]|0[0-9]):([0-5][0-9])|14:00))?';
   }
   description
     "The time type represents an instance of time of zero-duration
      in the specified timezone that recurs every day.";
 }

 typedef day {
   type enumeration {
     enum monday {
       description
         "This represents Monday.";
     }
     enum tuesday {
       description
         "This represents Tuesday.";
     }
     enum wednesday {
       description
         "This represents Wednesday";
     }
     enum thursday {
       description
         "This represents Thursday.";
     }
     enum friday {
       description
         "This represents Friday.";
     }
     enum saturday {
       description
         "This represents Saturday.";
     }
     enum sunday {
       description
         "This represents Sunday.";
     }
   }
   description
     "The type for representing the day of the week.";
 }

/*
 * Groupings
 */

 grouping ip-address-info {
   description
     "There are two types to configure a security policy
      for an IP address, such as IPv4 address and IPv6 address.";
   choice match-type {
     description
       "User can choose between IPv4 and IPv6.";
     case ipv4 {
       choice ipv4-range-or-prefix {
         description
           "User can choose between IPv4 address range and
            prefix type.";
         case prefix {
           leaf-list ipv4-prefix {
             type inet:ipv4-prefix;
             description
               "The IPv4 addresses in a prefix type.";
           }
         }
         case range {
           list range-ipv4-address {
             key "start end";
             leaf start {
               type inet:ipv4-address-no-zone;
               mandatory true;
               description
                 "A start IPv4 address for a range match.";
             }
             leaf end {
               type inet:ipv4-address-no-zone;
               mandatory true;
```

```
          description
            "An end IPv4 address for a range match.";
        }
        description
          "A range match for IPv4 addresses is provided.
           The ranges are inclusive, i.e., the range values
           include the value of 'start' and 'end'.
           Note that the start IPv4 address must be lower than
           the end IPv4 address.";
      }
    }
  }
}
case ipv6 {
  choice ipv6-range-or-prefix {
    description
      "User can choose between IPv6 address range and
       prefix type.";
    case prefix {
      leaf-list ipv6-prefix {
        type inet:ipv6-prefix;
        description
          "The IPv6 addresses in a prefix type.";
      }
    }
    case range {
      list range-ipv6-address {
        key "start end";
        leaf start {
          type inet:ipv6-address-no-zone;
          mandatory true;
          description
            "A start IPv6 address for a range match.";
        }
        leaf end {
          type inet:ipv6-address-no-zone;
          mandatory true;
          description
            "An end IPv6 address for a range match.";
        }
        description
          "A range match for IPv6 addresses is provided.
           The ranges are inclusive, i.e., the range values
           include the value of 'start' and 'end'.
           Note that the start IPv6 address must be lower than
           the end IPv6 address.";
      }
    }
  }
```

```
    }
  }
}

grouping user-group {
  description
    "This group represents user group information to label MAC
     addresses and IP (IPv4 or IPv6) addresses as a group of users.";
  leaf name {
    type string;
    description
      "This represents the name of a user-group.  A user-group name
       is used to map a user-group's name (e.g., employees) to IP
       address(es), MAC address(es).
       It is dependent on implementation.";
  }
  leaf-list mac-address {
    type yang:mac-address;
    description
      "Represent the MAC Address of a user-group. A user-group
       can have multiple MAC Addresses.";
  }
  uses ip-address-info {
    description
      "This represents the IP addresses of a user-group.";
    refine match-type {
      mandatory true;
    }
  }
}

grouping device-group {
  description
    "This group represents device group information to label
     IP (IPv4 or IPv6) addresses that provide services hosted
     on the application protocol.";
  leaf name {
    type string;
    description
      "This represents the name of a device-group.";
  }
  uses ip-address-info{
    description
      "This represents the IP addresses of a device-group.";
    refine match-type{
      mandatory true;
    }
  }
  leaf-list application-protocol {
    type identityref {
      base i2nsfmi:application-protocol;
    }
    description
      "This represents the application layer protocols of devices.
       If this is not set, it cannot support the appropriate
       protocol.";
  }
}

grouping location-group {
  description
    "This group represents location-group information to map
     IPv4 or IPv6 address to the geographical location.";
  leaf country {
    type string {
      length "2";
      pattern "[a-zA-Z]{2}";
    }
    description
      "This represents the 2-letter ISO country code conforming to
       ISO3166-1 alpha 2. Examples include 'US' for United States,
       'JP' for Japan, and 'PL' for Poland.";
    reference
      "ISO 3166-1: Decoding table alpha-2 country code";
  }
  leaf region {
    type string {
      length "5..6";
      pattern "[a-zA-Z]{2}-[a-zA-Z0-9]{2,3}";
    }
    description
      "This represents the ISO region code conforming to ISO 3166-2.
       Examples include 'ID-RI' for Riau province of Indonesia and
       'NG-RI' for the Rivers province in Nigeria.";
    reference
      "ISO 3166-2: 3166-2 subdivision code";
  }
  leaf city {
    type string;
    description
      "This represents the city of a region in English. Examples
       include 'Dublin', 'New York', and 'Sao Paulo'.";
  }
  uses ip-address-info{
    refine match-type{
      mandatory true;
      description
        "This represents the IP addresses of a location-group.";
    }
  }
}

grouping payload-string {
  description
    "The grouping for payload-string content.  It contains
     information such as name and string content.";
}

list i2nsf-cfi-policy {
  key "name";
  description
    "This is a security policy list.  Each policy in the list
     contains a list of security policy rules, and is a policy
     instance to have the information of where and when a policy
     needs to be applied.";
  leaf name {
    type string;
    description
      "The name which identifies the policy.";
  }
  leaf language {
    type string {
      pattern '((([A-Za-z]{2,3}(-[A-Za-z]{3}(-[A-Za-z]{3})'
            + '{0,2})?)|[A-Za-z]{4}|[A-Za-z]{5,8})(-[A-Za-z]{4})?'
            + '(-([A-Za-z]{2}|[0-9]{3}))?(-([A-Za-z0-9]{5,8}'
            + '|([0-9][A-Za-z0-9]{3})))*(-[0-9A-WYZa-wyz]'
            + '(-([A-Za-z0-9]{2,8}))+)*(-[Xx](-([A-Za-z0-9]'
            + '{1,8}))+)?|[Xx](-([A-Za-z0-9]{1,8}))+|'
            + '(([Ee][Nn]-[Gg][Bb]-[Oo][Ee][Dd]|[Ii]-'
            + '[Aa][Mm][Ii]|[Ii]-[Bb][Nn][Nn]|[Ii]-'
            + '[Dd][Ee][Ff][Aa][Uu][Ll][Tt]|[Ii]-'
            + '[Ee][Nn][Oo][Cc][Hh][Ii][Aa][Nn]'
            + '|[Ii]-[Hh][Aa][Kk]|'
            + '[Ii]-[Kk][Ll][Ii][Nn][Gg][Oo][Nn]|'
            + '[Ii]-[Ll][Uu][Xx]|[Ii]-[Mm][Ii][Nn][Gg][Oo]|'
            + '[Ii]-[Nn][Aa][Vv][Aa][Jj][Oo]|[Ii]-[Pp][Ww][Nn]|'
            + '[Ii]-[Tt][Aa][Oo]|[Ii]-[Tt][Aa][Yy]|'
            + '[Ii]-[Tt][Ss][Uu]|[Ss][Gg][Nn]-[Bb][Ee]-[Ff][Rr]|'
            + '[Ss][Gg][Nn]-[Bb][Ee]-[Nn][Ll]|[Ss][Gg][Nn]-'
            + '[Cc][Hh]-[Dd][Ee])|([Aa][Rr][Tt]-'
            + '[Ll][Oo][Jj][Bb][Aa][Nn]|[Cc][Ee][Ll]-'
            + '[Gg][Aa][Uu][Ll][Ii][Ss][Hh]|'
            + '[Nn][Oo]-[Bb][Oo][Kk]|[Nn][Oo]-'
            + '[Nn][Yy][Nn]|[Zz][Hh]-[Gg][Uu][Oo][Yy][Uu]|'
            + '[Zz][Hh]-[Hh][Aa][Kk][Kk][Aa]|[Zz][Hh]-'
            + '[Mm][Ii][Nn]|[Zz][Hh]-[Mm][Ii][Nn]-'
            + '[Nn][Aa][Nn]|[Zz][Hh]-[Xx][Ii][Aa][Nn][Gg])))';
    }
    default "en-US";
    description
      "The value in this field indicates the language tag
       used for all of the 'leaf description' described in the
       'i2nsf-cfi-policy'.

       The attribute is encoded following the rules in Section 2.1
       in RFC 5646. The default language tag is 'en-US'.";
    reference
      "RFC 5646: Tags for Identifying Languages";
  }
  leaf resolution-strategy {
    type identityref {
      base resolution-strategy;
    }
    default fmr;
    description
      "The resolution strategies that can be used to
       specify how to resolve conflicts that occur between
       actions of the same or different policy rules that
       are matched and contained in this particular NSF.";

    reference
      "draft-ietf-i2nsf-capability-data-model-32:
       I2NSF Capability YANG Data Model - Resolution strategy";
  }
  list rules {
    key "name";

    description
      "There can be a single or multiple number of rules.";
    leaf name {
      type string;
      description
        "This represents the name for a rule. Each rule name must
         be unique. Note that since this name is a key in the
         list of rules, its uniqueness is verified.";
    }

    leaf priority {
      type uint8;
      description
        "The priority of the rule to indicate the order of the
         rules to be matched. A higher value means a higher priority.
         The packet or flow will be matched with the rule with
         the highest priority value first and continues to a lower
         priority value. Once a rule matches the packet or flow,
         the NSF should execute the rule and terminate the matching
         process. If multiple rules have an equal priority, the
         actual order is undefined. The handling of the selection
         of those rules depends on the implementer, e.g.,
         order-by-user (i.e., process the rules in the order that
         they are specified in the list), alphabetical order of the
         rule's name, or random rule selection.";
    }

    container event {
      description
        "This represents an event (i.e., a security event), for
         which a security rule is made.";
      leaf-list system-event {
        type identityref {
          base i2nsfmi:system-event;
        }
        description
          "The security policy rule according to
           system events.";
      }

      leaf-list system-alarm {
        type identityref {
          base i2nsfmi:system-alarm;
        }
        description
          "The security policy rule according to
           system alarms.";
      }
    }

    container condition {
      description
      "Conditions for general security policies. All configured
       conditions must match for a rule to trigger.";
      container firewall {
        description
          "A general firewall condition based on the packet
           header.";
        leaf-list source {
          type union {
            type leafref {
              path "/endpoint-groups/user-group/name";
            }
            type leafref {
              path "/endpoint-groups/device-group/name";
            }
          }
          description
          "This describes the path of the source.";
        }

        leaf-list destination {
          type union {
            type leafref {
              path "/endpoint-groups/user-group/name";
            }
            type leafref {
              path "/endpoint-groups/device-group/name";
            }
          }
          description
            "This describes the path to the destinations.";
        }

        leaf transport-layer-protocol {
          type identityref {
            base i2nsfmi:transport-protocol;
          }
          description
            "The transport-layer protocol to be matched.";
        }

        list range-port-number {
          key "start end";
          leaf start {
            type inet:port-number;
            description
              "A start port number for a range match.";
          }
          leaf end {
            type inet:port-number;
            must '. >= ../start' {
              error-message
                "An end port number MUST be equal to or greater than
                 a start port number.";
            }
            description
              "An end port number for a range match.";
          }
          description
            "A range match for transport-layer port number.
             The ranges are inclusive, i.e., the range values include
             the value of 'start' and 'end'. Note that the start port
             number value must be lower than the end port number
             value.";
        }
```

```
  container icmp {
    description
       "Represents the ICMPv4 and ICMPv6 packet header
        information to determine if the set of policy
        actions in this ECA policy rule should be executed
        or not.";
     reference
       "RFC  792: Internet Control Message Protocol
        RFC 8335: PROBE: A Utility for Probing Interfaces";

    leaf-list message {
      type identityref {
        base icmp-message;
      }
      description
        "The security policy rule according to
         ICMP message. The type is representing the
         ICMP message corresponds to the ICMP type and
         code.";
      reference
        "RFC  792: Internet Control Message Protocol
         RFC 8335: PROBE: A Utility for Probing Interfaces
         IANA: Internet Control Message Protocol (ICMP)
         Parameters
         IANA: Internet Control Message Protocol version 6
         (ICMPv6) Parameters";
    }
  }
}

container ddos {
  description
    "A condition for a DDoS attack.";
  container rate-limit {
    description
      "This describes the rate-limit.";
    leaf packet-rate-threshold {
      type uint64;
      units "pps";
      description
        "This is a trigger value for a rate limit of packet
         rate in packets per second (pps) for a
         DDoS-attack mitigation.";
    }
    leaf byte-rate-threshold {
      type uint64;
      units "Bps";
      description
        "This is a trigger value for a rate limit of byte
```

```
         rate in bytes per second (Bps) for a DDoS-attack
         mitigation.";
    }
    leaf flow-rate-threshold {
      type uint64;
      description
        "This is a trigger value for a rate limit of flow
         creating request rate (e.g., new TCP connection
         establishment) in flows per second for a DDoS-attack
         mitigation.";
    }
  }
}

container anti-virus {
  description
   "A condition for Antivirus service";

  leaf-list profile {
    type string;
    description
      "The path or name of the file that contains a security
       profile for the Antivirus service configuration. The
       security profile is used to scan the viruses. The
       absolute path and relative ones are to be interpreted as
       globs.";
    reference
      "GLOB: The Open Group Base Specifications Issue 7 - glob";
  }

  leaf-list exception-files {
    type string;
    description
      "The type or name of the files to be excluded by the
       Antivirus service.  This can be used to keep the known
       harmless files. Absolute paths are filenames/paths
       to be excluded, and relative ones are interpreted as
       globs.";
    reference
      "GLOB: The Open Group Base Specifications Issue 7 - glob";
  }
}

container payload {
  description
    "A condition based on a packet's content.";
  leaf-list content {
    type leafref {
      path "/threat-prevention/payload-content/name";
    }
    description
      "This describes the paths to a packet content's.";
  }
}

container url-category {
  description
    "Condition for url category.";
  leaf url-name {
    type leafref {
      path "/endpoint-groups/url-group/name";
    }
    description
      "This is description for the condition of a URL's
       category such as SNS sites, game sites, e-commerce
       sites, company sites, and university sites.";
  }
}

container voice {
  description
    "For the VoIP/VoCN security system, a VoIP/
     VoCN security system can monitor each
     VoIP/VoCN flow and manage VoIP/VoCN
     security rules controlled by a centralized
     server for VoIP/VoCN security service
     (called VoIP IPS).  The VoIP/VoCN security
     system controls each switch for the
     VoIP/VoCN call flow management by
     manipulating the rules that can be added,
     deleted, or modified dynamically.
     Note that VoIP is Voice over Internet Protocol
     and VoCN is Voice over Cellular Network such as
     Voice over LTE or 5G.";
  reference
    "RFC 3261: SIP: Session Initiation Protocol";

  leaf-list source-id {
    type leafref {
      path "/endpoint-groups/voice-group/name";
    }
    description
      "The security policy rule according to
       the 'From' header field of the SIP.";
    reference
      "RFC 3261: SIP: Session Initiation Protocol
                 - Section 8.1.1.3 (From)";
  }
  leaf-list destination-id {
    type leafref {
      path "/endpoint-groups/voice-group/name";
    }
    description
      "The security policy rule according to
       the 'To' header field of the SIP.";
    reference
      "RFC 3261: SIP: Session Initiation Protocol
                 - Section 8.1.1.2 (To)";
  }

  leaf-list user-agent {
    type string;
    description
      "The security policy rule according to
       the 'user-agent' field of the SIP.";
    reference
      "RFC 3261: SIP: Session Initiation Protocol
                 - Section 20.41 (User-Agent)";
  }
}

container context {
  description
    "Condition for matching the context of the packet, such
     as geographic location, time, packet direction.";
  container time {
    description
      "The time when a security policy rule should be
       applied.";
    leaf start-date-time {
      type yang:date-and-time;
      description
        "This is the start date and time for a security
         policy rule.";
    }
    leaf end-date-time {
      type yang:date-and-time;
      description
        "This is the end date and time for a security policy
         rule.  The policy rule will stop working after the
         specified end date and time.";
    }
    container period {
      when
        "../frequency!='only-once'";
      description
        "This represents the repetition time.";
      leaf start-time {
        type time;
        description
          "This is a period's start time for an event.";
      }
      leaf end-time {
        type time;
        description
          "This is a period's end time for an event.";
      }
      leaf-list day {
        when
          "../../frequency='weekly'";
        type day;
        min-elements 1;
        description
          "This represents the repeated day of every week
           (e.g., Monday and Tuesday).  More than one day can
           be specified.";
      }
      leaf-list date {
        when
          "../../frequency='monthly'";
        type int8 {
          range "1..31";
        }
        min-elements 1;
        description
          "This represents the repeated day of every month.
           More than one date can be specified.";
      }
      list month {
        when
          "../../frequency='yearly'";

        key "start end";

        leaf start {
          type string {
            pattern '\d{2}-\d{2}';
          }
          description
            "The starting range of the month and date of every
             year.  A pattern used here is Month and Date
             (MM-DD).";
        }
        leaf end {
          type string {
```

```
              pattern '\d{2}-\d{2}';
            }
            description
              "The ending range of the month and date of every
               year.  A pattern used here is Month and Date
               (MM-DD).  The 'end' value must be greater than or
               equal to the 'start' value.";
          }
          min-elements 1;
          description
            "This represents the repeated month and date of
             every year.  More than one range can be specified.
             If one specific month and date is needed, then set
             both start and end to the same value.

             Note that the ranges are inclusive, i.e., the range
             values include the values of start and end.";
        }
      }

      leaf frequency {
        type enumeration {
          enum only-once {
            description
              "This represents that the rule is immediately
               enforced only once and not repeated. The policy
               will continuously be active from the
               start-date-time to the end-date-time.";
          }
          enum daily {
            description
              "This represents that the rule is enforced on a
               daily basis. The policy will be repeated daily
               until the end-date-time.";
          }
          enum weekly {
            description
              "This represents that the rule is enforced on a
               weekly basis. The policy will be repeated weekly
               until the end-date-time. The repeated days can
               be specified.";
          }
          enum monthly {
            description
              "This represents that the rule is enforced on a
               monthly basis. The policy will be repeated
               monthly until the end-date-time.";
          }
          enum yearly {
```

```
        description
          "This represents that the rule is enforced on a
           yearly basis.  The policy will be repeated
           yearly until the end-date-time.";
      }
    }
    default only-once;
    description
      "This represents how frequently the rule should be
       enforced.";
  }
}

container application {
  description
    "Condition for application.";
  leaf-list protocol {
    type identityref {
      base i2nsfmi:application-protocol;
    }
    description
      "The condition based on the application layer
       protocol";
  }
}

container device-type {
  description
    "Condition for type of the destination device.";
  leaf-list device {
    type identityref {
      base device-type;
    }
    description
      "The device attribute that can identify a device (i.e.,
       computer, mobile phone, smartphone, VoIP/VoCN phone,
       tablet, network infrastructure device, IoT device,
       OT device, and vehicle).";
  }
}

container users {
  description
    "Condition for users.";
  list user {
    key "id";
    description
      "The user with which the traffic flow is associated
       can be identified by either a user ID or username.
       The user-to-IP address mapping is assumed to be
       provided by the unified user management system via
       network.";
    leaf id {
      type uint32;
      description
        "The ID of the user.";
    }
    leaf name {
      type string;
      description
        "The name of the user.";
    }
  }
  list group {
    key "id";
    description
      "The user group with which the traffic flow is
       associated can be identified by either a group ID
       or group name. The group-to-IP address and
       user-to-group mappings are assumed to be provided by
       the unified user management system via network.";
    leaf id {
      type uint32;
      description
        "The ID of the group.";
    }
    leaf name {
      type string;
      description
        "The name of the group.";
    }
  }
}

container geographic-location {
  description
    "A condition for a location-based connection.";
  container source {
    leaf country {
      type leafref {
        path "/endpoint-groups/location-group/country";
      }
      description
        "The name of the country in the 2-letter ISO country
         code conforming to ISO3166-1 alpha-2.";
      reference
        "ISO 3166-1: Decoding table alpha-2 country code";
    }
    leaf region {
      type leafref {
        path "/endpoint-groups/location-group/region";
      }
      description
        "The region code conforming to ISO 3166-2.";
      reference
        "ISO 3166-2: 3166-2 subdivision code.";
    }
    leaf city {
      type leafref {
        path "/endpoint-groups/location-group/city";
      }
      description
        "The name of the city of the location.";
    }
    description
      "This describes the paths to a location's source.
       The values in this field will be mapped into
       either IPv4 or IPv6 addresses defined in
       /endpoint-groups/location-group.";
  }
  container destination {
    leaf country {
      type leafref {
        path "/endpoint-groups/location-group/country";
      }
      description
        "The name of the country in the 2-letter ISO country
         code conforming to ISO3166-1 alpha-2.";
      reference
        "ISO 3166-1: Decoding table alpha-2 country code";
    }
    leaf region {
      type leafref {
        path "/endpoint-groups/location-group/region";
      }
      description
        "The region code conforming to ISO 3166-2.";
      reference
        "ISO 3166-2: 3166-2 subdivision code.";
    }
    leaf city {
      type leafref {
        path "/endpoint-groups/location-group/city";
      }
      description
        "The name of the city of the location.";
    }
```

```
        description
          "This describes the paths to a location's
           destination. The values in this field will be
           mapped into either IPv4 or IPv6 addresses defined in
           /endpoint-groups/location-group.";
      }
    }
  }

  container threat-feed {
    description
      "A condition based on the threat-feed information.";
    leaf-list name {
      type leafref {
        path "/threat-prevention/threat-feed-list/name";
      }
      description
        "This describes the paths to a threat-feed's sources.";
    }
  }
}

container action {
  description
    "This is the action container.";
  container primary-action {
    description
      "This represents primary actions (e.g., ingress and
       egress actions) to be applied to a condition.
       If this is not set, it cannot support the primary
       actions.";
    leaf action {
      type identityref {
        base primary-action;
      }
      mandatory true;
      description
        "Ingress actions: pass, drop, reject, rate-limit,
         and mirror.
         Egress actions: pass, drop, reject, rate-limit,
         mirror, invoke-signaling, tunnel-encapsulation,
         forward, and transform.";
    }
    leaf limit {
      when "../action = 'i2nsfcfi:rate-limit'" {
        description
          "Rate-limit is valid only when rate-limit action is
           used.";
      }
```

```
          type decimal64 {
            fraction-digits 2;
          }
          units "bytes per second";
          description
            "Specifies how to rate-limit the traffic.";
        }
      }
      container secondary-action {
        description
          "This represents secondary actions (e.g., log and syslog)
           to be applied if they are needed.  If this is not set,
           it cannot support the secondary actions.";
        leaf log-action {
          type identityref {
            base secondary-action;
          }
          description
            "Log action: rule log and session log.";
        }
      }
    }
  }
}

container endpoint-groups {
  description
    "A logical entity in a business environment, where a security
     policy is to be applied.";
  list user-group{
    uses user-group;
    key "name";
    description
      "This represents a user group.";
  }
  list device-group {
    key "name";
    uses device-group;
    description
      "This represents a device group.";
  }
  list location-group{
    key "country region city";
    uses location-group;
    description
      "This represents a location group.";
  }
  list url-group {
    key "name";
    description
      "This describes the list of URL.";
    leaf name {
      type string;
      description
        "This is the name of URL group, e.g., SNS sites,
         gaming sites, e-commerce sites.";
    }
    leaf-list url {
      type inet:uri;
      description
        "Specifies the URL to be added into the group.";
      reference
        "RFC 3986: Uniform Resource Identifier (URI): Generic
         Syntax";
    }
  }
  list voice-group {
    key "name";
    description
      "This describes the list of Voice ID.";
    leaf name {
      type string;
      description
        "This is the name of the voice group.";
    }
    leaf-list sip-id {
      type inet:uri;
      description
        "Specifies the logical identity of the SIP user written in
         SIP URI scheme.";
      reference
        "RFC3261: SIP: Session Initiation Protocol
                  - Section 19.1.1 (SIP and SIPS URI Components)";
    }
  }
}

container threat-prevention {
  description
    "The container for threat-prevention.";
  list threat-feed-list {
    key "name";
    description
      "There can be a single or multiple number of
       threat-feeds.";
    leaf name {
      type string;
      description
        "This represents the name of the threat-feed.";
    }
    leaf-list ioc {
      type string;
      description
        "This field represents the Indicators of Compromise (IOC),
         i.e., the critical information of patterns or characteristics
         (signatures) in the threat feed that identifies malicious
         activities. The format of the information given in this field
         should be parsed based on the format field (e.g., STIX, MISP,
         OpenIOC, and IODEF).";
    }
    leaf format {
      type identityref {
        base ioc-format;
      }
      mandatory true;
      description
        "This represents the format of the IOC information. This
         field is mandatory to parse the IOC. The examples of the
         format are STIX, MISP, OpenIOC, and IODEF.";
      reference
        "STIX: Structured Threat Information Expression version 2.1
         MISPCORE: Malware Information Sharing Platform (MISP) Core
         Format
         OPENIOC: OpenIOC 1.1 Schema document
         RFC 8727: JSON Binding of the Incident Object Description
         Exchange Format";
    }
  }

  list payload-content {
    key "name";
    leaf name {
      type string;
      description
        "This represents the name of a packet's payload-content.
         It should give an idea of why a specific payload content
         is marked as a threat.  For example, the name 'backdoor'
         indicates the payload content is related to a backdoor
         attack.";
    }
    leaf description {
      type string;
      description
        "This represents the description of a payload. Describe
         how the payload contents are related to a security
         attack.";
    }
```

```
list contents {
  key "content";
  ordered-by user;
  leaf content {
    type binary;
    description
      "This represents the pattern of the payload contents (i.e.,
       the data after a transport layer header) to be matched.
       Due to the types of threats, the type of the content is
       defined as a binary to accommodate any kind of payload
       type such as HTTP, HTTPS, and SIP.

       If multiple instances of contents are defined, all
       defined contents must be matched somewhere in the session
       stream.  The content pattern should be matched based on
       the order given by the user. The scope of the payload to be
       matched can be defined by the depth and offset/distance
       fields.";
  }
  leaf depth {
    type uint16 {
      range "1..max";
    }
    units "bytes";
    description
      "The field specifies how far a packet should be searched
       for the specified content pattern defined in the content
       field. For example, a depth of 5 means to only look for
       the specified content pattern within the first 5 bytes
       of the payload. This field accepts values greater than or
       equal to the content length being searched. If this
       field is undefined, then the content pattern should be
       searched within the whole payload.";
  }
  choice starting-point {
    description
      "Choice of how to specify the starting point of matching
       the pattern to the payload. If this field is undefined,
       then the content pattern should be searched from the
       beginning of the payload.";
    case offset {
      leaf offset {
        type int32;
        units "bytes";
        description
          "The field specifies where to start searching for the
           specified content pattern within the payload.
           For example, an offset of 5 means to start looking for
           the specified content pattern after the first 5 bytes
```

```
                 of the payload. A negative value means to start from
                 the last bytes of the payload. For example, an offset
                 of -5 means to start looking for the specified content
                 pattern from the last 5 bytes of the payload.";
            }
          }
          case distance {
            leaf distance {
              type int32;
              units "bytes";
              description
                "The field specifies how far a payload should be
                 ignored before starting to search for the specified
                 content pattern relative to the end of the previous
                 specified content pattern match. This can be thought
                 of as exactly the same thing as offset, except it is
                 relative to the end of the last pattern match instead
                 of the beginning of the packet. For example, a distance
                 of 5 means to start looking for the specified content
                 pattern 5 bytes after the last byte of the matched
                 pattern. A negative value means to start looking before
                 the last byte of the previous matched pattern. For
                 example, a distance of -5 means to start looking for
                 the specified content pattern 5 bytes before the last
                 byte of the previous matched pattern.

                 Note that this field cannot be used if the content is
                 the first order of the list.";
            }
          }
        }
        description
          "List of contents and their scopes for matching content
           pattern with the payload.";
      }
      description
        "This represents a payload-string group.";
    }
  }
}
```

<CODE ENDS>

Figure 18: YANG for Consumer-Facing Interface

7. XML Configuration Examples of High-Level Security Policy Rules

This section shows XML configuration examples of high-level security policy rules that are delivered from the I2NSF User to the Security Controller over the Consumer-Facing Interface. The considered examples are: Database registration, time-based firewall for web filtering, VoIP/VoCN security service, and DDoS-attack mitigation.

7.1. Database Registration: Information of Positions and Devices (Endpoint Group)

The endpoint-group is used to register known network nodes and label them into a higher-level name (i.e., human recognizable language). If new endpoints are introduced to the network, it is necessary to first register their data to the database. For example, if new members are newly introduced in different groups (i.e., user-group, device-group, url-group, and voice-group), each of them should be registered as separate entities with their corresponding information.

```
<?xml version="1.0" encoding="UTF-8" ?>
<endpoint-groups
 xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-cons-facing-interface"
 xmlns:i2nsfmi="urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitoring-interf
  <user-group>
    <name>employees</name>
    <range-ipv4-address>
      <start>192.0.2.11</start>
      <end>192.0.2.90</end>
    </range-ipv4-address>
  </user-group>
  <device-group>
    <name>webservers</name>
    <range-ipv4-address>
      <start>198.51.100.11</start>
      <end>198.51.100.20</end>
    </range-ipv4-address>
    <application-protocol>i2nsfmi:http</application-protocol>
    <application-protocol>i2nsfmi:https</application-protocol>
  </device-group>
  <url-group>
    <name>sns-websites</name>
    <url>https://www.example.com/</url>
    <url>https://www.example.net/</url>
  </url-group>
  <voice-group>
    <name>malicious-id</name>
    <sip-id>sip:alice@example.org</sip-id>
    <sip-id>sip:bob@203.0.113.15</sip-id>
  </voice-group>
</endpoint-groups>
```

Figure 19: Registering User-group, Device-group, Voice-group in IPv4 Addresses, and URL-group Information

Figure 19 shows an example XML representation of the registered information for the user-group, device-group, voice-group in IPv4 address [RFC5737], and url-group.

1. The IPv4 addresses from 192.0.2.11 to 192.0.2.90 are labeled as a group of users called "employees".

2. The IPv4 addresses from 198.51.100.11 to 198.51.100.20 provide services with HTTP and HTTPS application protocol labeled as "webservers".

3. The "https://www.example.com/" and "https://www.example.net/" URLs are labeled as "sns-websites".

4. The "sip:alice@example.org" and "sip:bob@203.0.113.15" SIP identities are labeled as "malicious-id".

```
<?xml version="1.0" encoding="UTF-8" ?>
<endpoint-groups
 xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-cons-facing-interface"
 xmlns:i2nsfmi="urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitoring-interf
  <user-group>
    <name>employees-v6</name>
    <range-ipv6-address>
      <start>2001:db8:0:1::11</start>
      <end>2001:db8:0:1::90</end>
    </range-ipv6-address>
  </user-group>
  <device-group>
    <name>webservers-v6</name>
    <range-ipv6-address>
      <start>2001:db8:0:2::11</start>
      <end>2001:db8:0:2::20</end>
    </range-ipv6-address>
    <application-protocol>i2nsfmi:http</application-protocol>
    <application-protocol>i2nsfmi:https</application-protocol>
  </device-group>
  <voice-group>
    <name>malicious-id-v6</name>
    <sip-id>sip:david@2001:db8:2ef0::32b7</sip-id>
  </voice-group>
</endpoint-groups>
```

Figure 20: Registering User-group, Device-group, Voice-group Information in IPv6 Addresses

Also, Figure 20 shows an example XML representation of the registered information for the user-group, device-group, and voice-group in IPv6 addresses [RFC3849].

1. The IPv6 addresses from 2001:db8:0:1::11 to 2001:db8:0:1::90 are labeled as a group of users called "employees-v6".

2. The IPv6 addresses from 2001:db8:0:2::11 to 2001:db8:0:2::20 provide services with HTTP and HTTPS application protocol labeled as "webservers-v6".

3. The "sip:david@[2001:db8:2ef0::32b7]" SIP identity is labeled as "malicious-id-v6".

7.2. Scenario 1: Block SNS Access during Business Hours

The first example scenario is to "block SNS access during office hours" using a time-based firewall policy. In this scenario, all users registered as "employees" in the user-group list are unable to access SNS during the office hours (weekdays). The XML instance is described below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<i2nsf-cfi-policy
 xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-cons-facing-interface">
  <name>security_policy_for_blocking_sns</name>
  <rules>
    <name>block_access_to_sns_during_office_hours</name>
    <condition>
      <firewall>
        <source>employees</source>
      </firewall>
      <url-category>
        <url-name>sns-websites</url-name>
      </url-category>
      <context>
        <time>
          <start-date-time>2021-03-11T09:00:00.00Z</start-date-time>
          <end-date-time>2021-12-31T18:00:00.00Z</end-date-time>
          <period>
            <start-time>09:00:00Z</start-time>
            <end-time>18:00:00Z</end-time>
            <day>monday</day>
            <day>tuesday</day>
            <day>wednesday</day>
            <day>thursday</day>
            <day>friday</day>
          </period>
          <frequency>weekly</frequency>
        </time>
      </context>
    </condition>
    <action>
      <primary-action>
        <action>drop</action>
      </primary-action>
    </action>
  </rules>
</i2nsf-cfi-policy>
```

Figure 21: An XML Example for Time-based Firewall

Time-based-condition Firewall

1. The policy name is "security_policy_for_blocking_sns".

2. The rule name is "block_access_to_sns_during_office_hours".

3. The Source is "employees".

4. The destination target is "sns-websites". "sns-websites" is the key which represents the list containing the information, such as URL, about sns-websites.

5. The action required is to "drop" any attempt to connect to websites related to Social networking.

7.3. Scenario 2: Block Malicious VoIP/VoCN Packets Coming to a Company

The second example scenario is to "block malicious VoIP/VoCN packets coming to a company" using a VoIP policy. In this scenario, the calls coming from VOIP and/or VoCN sources with VoCN IDs that are classified as malicious are dropped. The IP addresses of the employees and malicious VOIP IDs which should be blocked are stored in the database or datastore of the enterprise. Here and for the rest of the cases, it is assumed that the security administrators or someone responsible for the existing and newly generated policies, are not aware of which and/or how many NSFs are needed to meet the security requirements. Figure 22 represents the XML document generated from YANG discussed in previous sections. Once a high-level security policy is created by a security admin, it is delivered by the Consumer-Facing Interface, through RESTCONF server, to the security controller. The XML instance is described below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<i2nsf-cfi-policy
 xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-cons-facing-interface">
  <name>
    security_policy_for_blocking_malicious_voip_packets
  </name>
  <rules>
    <name>Block_malicious_voip_and_vocn_packets</name>
    <condition>
      <voice>
        <source-id>malicious-id</source-id>
      </voice>
      <firewall>
        <destination>employees</destination>
      </firewall>
    </condition>
    <action>
      <primary-action>
        <action>drop</action>
      </primary-action>
    </action>
  </rules>
</i2nsf-cfi-policy>
```

Figure 22: An XML Example for VoIP Security Service

Custom-condition Firewall

1. The policy name is "security_policy_for_blocking_malicious_voip_packets".

2. The rule name is "Block_malicious_voip_and_vocn_packets".

3. The source is "malicious-id". The "malicious-id" is the key, so that it maps to the SIP identities that are named as "malicious-id". This can be a single SIP identity or a list of SIP identities.

4. The destination target is "employees". "employees" is the key which represents the list containing information about employees, such as IP addresses.

5. The action required is "drop" when any incoming SIP packets are coming from "malicious-id" and targeting "employees".

7.4. Scenario 3: Mitigate Flood Attacks on a Company Web Server

The third example scenario is to "Mitigate flood attacks on a company web server" using 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 that target "webservers" are more than the set threshold, then the admin can set the percentage of the packets to be dropped to safely maintain the service. Once the rule is set and delivered and enforced to the NSFs by the security controller, the NSFs will monitor the incoming packet amounts to act according to the rule set. The XML instance is described below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<i2nsf-cfi-policy
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-cons-facing-interface">
  <name>security_policy_for_ddos_attacks</name>
  <rules>
    <name>1000_packets_per_second</name>
    <condition>
      <firewall>
        <destination>webservers</destination>
      </firewall>
      <ddos>
        <rate-limit>
          <packet-rate-threshold>1000</packet-rate-threshold>
        </rate-limit>
      </ddos>
    </condition>
    <action>
      <primary-action>
        <action>drop</action>
      </primary-action>
    </action>
  </rules>
</i2nsf-cfi-policy>
```

Figure 23: An XML Example for DDoS-attack Mitigation

DDoS-condition Firewall

1. The policy name is "security_policy_for_ddos_attacks".

2. The rule name is "1000_packets_per_second".

3. The destination is webservers.

4. The rate limit exists to limit the incoming amount of packets per second. In this case the rate limit is "1000" packets per second. This amount depends on the packet receiving capacity of the server devices.

5. The Source is all sources which send abnormal amount of packets. It is assumed that there is a counter per source IP address in this DDoS-condition Firewall. The rate of "1000" packets per second is set for each source to send packets toward the destinations as webservers.

6. The action required is to "drop" when the packet reception is more than "1000" packets per second for each source that sends packets to the destinations.

8. IANA Considerations

This document requests IANA to register the following URI in the "IETF XML Registry" [RFC3688]:

```
URI: urn:ietf:params:xml:ns:yang:ietf-i2nsf-cons-facing-interface
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

This document requests IANA to register the following YANG module in the "YANG Module Names" registry [RFC7950][RFC8525]:

```
name: ietf-i2nsf-cons-facing-interface
namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-cons-facing-interface
prefix: i2nsfcfi
reference: RFC XXXX

// RFC Ed.: replace XXXX with an actual RFC number and remove
// this note.
```

9. Security Considerations

The YANG module specified in this document defines a data schema designed to be accessed through network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the required secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the required secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides a means of restricting access to specific NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and contents. Thus, NACM SHOULD be used to restrict the NSF registration from unauthorized users.

There are a number of data nodes defined in this YANG module that are writable, creatable, and deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations to these data nodes could have a negative effect on network and security operations. These data nodes have the following sensitivity/vulnerability:

- list i2nsf-cfi-policy: Writing to almost any element of this YANG module would directly impact the configuration of NSFs implementing the security policy, e.g., completely turning off security monitoring and mitigation capabilities; altering the scope of this monitoring and mitigation; creating an overwhelming logging volume to overwhelm downstream analytics or storage capacity; creating logging patterns which are confusing; or reducing the efficacy of statistics or artificial models built on historical data.

- container endpoint-groups: Writing to any element in this container can alter the configuration of the security services and may cause vulnerabilities in the network, e.g., changing registered malicious endpoints can remove the defense against known hostile clients. The information given may also be considered private, hence it is strongly encouraged to inform affected users/customers of this fact and of the potential privacy-related consequences and trade-offs.

- container threat-prevention: Writing to any element in this container can alter the configuration of the security services and may cause vulnerabilities in the network, e.g., changing registered signature can let malicious content to get across the secured network without detection.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes with their sensitivity/vulnerability:

- list i2nsf-cfi-policy: The leak of this node to an attacker could reveal the specific configuration of security controls to an attacker. An attacker can craft an attack path that avoids observation or mitigations; one may reveal topology information to inform additional targets or enable lateral movement; one enables the construction of an attack path that avoids observation or mitigations; one provides an indication that the operator has discovered the attack.

- container endpoint-groups: This node holds a list of endpoint data that may be considered private to the users. Disclosure of this information may expose sensitive details which can be used to define the identity and geographical location of a user.

Malicious actors can leverage this information to threaten the user with cyber threat, e.g., voice phishing, or physical threat.

*container threat-prevention: The leak of this node to an attacker could reveal the specific detection system to an attacker. An attacker can use this information to design new unknown attack strategies to circumvent the existing detection or prevention system.

10. References

10.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[RFC0854] Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, DOI 10.17487/RFC0854, May 1983, <https://www.rfc-editor.org/info/rfc854>.

[RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, DOI 10.17487/RFC0959, October 1985, <https://www.rfc-editor.org/info/rfc959>.

[RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, DOI 10.17487/RFC1939, May 1996, <https://www.rfc-editor.org/info/rfc1939>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, DOI 10.17487/RFC2595, June 1999, <https://www.rfc-editor.org/info/rfc2595>.

[RFC3261]

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC4250] Lehtinen, S. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Assigned Numbers", RFC 4250, DOI 10.17487/RFC4250, January 2006, <https://www.rfc-editor.org/info/rfc4250>.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <https://www.rfc-editor.org/info/rfc4340>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/info/rfc5321>.

[RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <https://www.rfc-editor.org/info/rfc5646>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950]
Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8075] Castellani, A., Loreto, S., Rahman, A., Fossati, T., and E. Dijk, "Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP)", RFC 8075, DOI 10.17487/RFC8075, February 2017, <https://www.rfc-editor.org/info/rfc8075>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8335] Bonica, R., Thomas, R., Linkova, J., Lenart, C., and M. Boucadair, "PROBE: A Utility for Probing Interfaces", RFC 8335, DOI 10.17487/RFC8335, February 2018, <https://www.rfc-editor.org/info/rfc8335>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <https://www.rfc-editor.org/info/rfc8407>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI

10.17487/RFC8525, March 2019, <https://www.rfc-editor.org/info/rfc8525>.

[RFC8727] Takahashi, T., Danyliw, R., and M. Suzuki, "JSON Binding of the Incident Object Description Exchange Format", RFC 8727, DOI 10.17487/RFC8727, August 2020, <https://www.rfc-editor.org/info/rfc8727>.

[RFC9051] Melnikov, A., Ed. and B. Leiba, Ed., "Internet Message Access Protocol (IMAP) - Version 4rev2", RFC 9051, DOI 10.17487/RFC9051, August 2021, <https://www.rfc-editor.org/info/rfc9051>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[RFC9112] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <https://www.rfc-editor.org/info/rfc9112>.

[RFC9113] Thomson, M., Ed. and C. Benfield, Ed., "HTTP/2", RFC 9113, DOI 10.17487/RFC9113, June 2022, <https://www.rfc-editor.org/info/rfc9113>.

[RFC9260] Stewart, R., Tüxen, M., and K. Nielsen, "Stream Control Transmission Protocol", RFC 9260, DOI 10.17487/RFC9260, June 2022, <https://www.rfc-editor.org/info/rfc9260>.

[RFC9293] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <https://www.rfc-editor.org/info/rfc9293>.

[I-D.ietf-i2nsf-capability-data-model]
Hares, S., Jeong, J. P., Kim, J. T., Moskowitz, R., and Q. Lin, "I2NSF Capability YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-capability-data-model-32, 23 May 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-capability-data-model-32>.

[I-D.ietf-i2nsf-nsf-monitoring-data-model]
Jeong, J. P., Lingga, P., Hares, S., Xia, L., and H. Birkholz, "I2NSF NSF Monitoring Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-nsf-monitoring-data-model-20, 1 June 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-nsf-monitoring-data-model-20>.

[GLOB] IEEE, "The Open Group Base Specifications Issue 7, 2018 Edition", IEEE Std 1003.1-2017, <https://pubs.opengroup.org/onlinepubs/9699919799/functions/glob.html>.

[ISO-3166-1alpha2] ISO, "ISO 3166-1 decoding table", <https://www.iso.org/iso/home/standards/country_codes/iso-3166-1_decoding_table.htm>.

[ISO-3166-2] ISO, "ISO 3166-2:2007", <https://www.iso.org/iso/home/standards/country_codes.htm#2012_iso3166-2>.

[STIX] Jordan, B., Piazza, R., and T. Darley, "Structured Threat Information Expression (STIX)", STIX Version 2.1 https://docs.oasis-open.org/cti/stix/v2.1/os/stix-v2.1-os.html, June 2021.

10.2. Informative References

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <https://www.rfc-editor.org/info/rfc3022>.

[RFC3849] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, DOI 10.17487/RFC3849, July 2004, <https://www.rfc-editor.org/info/rfc3849>.

[RFC5737] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation", RFC 5737, DOI 10.17487/RFC5737, January 2010, <https://www.rfc-editor.org/info/rfc5737>.

[RFC8329] Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", RFC 8329, DOI 10.17487/RFC8329, February 2018, <https://www.rfc-editor.org/info/rfc8329>.

[RFC8811] Mortensen, A., Ed., Reddy.K, T., Ed., Andreasen, F., Teague, N., and R. Compton, "DDoS Open Threat Signaling (DOTS) Architecture", RFC 8811, DOI 10.17487/RFC8811, August 2020, <https://www.rfc-editor.org/info/rfc8811>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[IANA-ICMP-Parameters] Internet Assigned Numbers Authority (IANA), "Assigned Internet Protocol Numbers", February 2021, <https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml>.

[IANA-ICMPv6-Parameters] Internet Assigned Numbers Authority (IANA), "Internet Control Message Procotol version 6 (ICMPv6) Parameters", February 2021, <https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml>.

[MISPCORE] Dulaunoy, A. and A. Iklody, "MISP Core", commit 051e33b6711a660faf81733d825f1015aa0d301b, February 2022, <https://github.com/MISP/misp-rfc/blob/051e33b6711a660faf81733d825f1015aa0d301b/misp-core-format/raw.md.html>.

[OPENIOC] Gibb, W., "OpenIOC 1.1 DRAFT", commit d42a8777708e171f8bdd3c2c9f8590c83488285d, August 2013, <https://github.com/fireeye/OpenIOC_1.1/blob/d42a8777708e171f8bdd3c2c9f8590c83488285d/schemas/ioc.xsd>.

[TR-29.949-3GPP] 3GPP, "Study on technical aspects on roaming end-to-end scenarios with Voice over LTE (VoLTE) IP Multimedia Subsystem (IMS) and other networks", 3GPP TR 29.949/Version 16.0.0, July 2020.

[TR-21.915-3GPP] 3GPP, "Summary of Rel-15 Work Items", 3GPP TR 21.915/Version 15.0.0, September 2019.

Appendix A. Acknowledgments

This document is a product by the I2NSF Working Group (WG) including WG Chairs (i.e., Linda Dunbar and Yoav Nir) and Diego Lopez. This document took advantage of the review and comments from the following people: Roman Danyliw, Mahdi F. Dachmehchi, Daeyoung Hyun, Jan Lindblad (YANG doctor), Tom Petch, Charlie Kaufman, Penglin Yang, and Jung-Soo Park. The authors sincerely appreciate their sincere efforts and kind help.

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea MSIT (Ministry of Science and ICT) (R-20160222-002755, Cloud based Security Intelligence Technology Development for the Customized Security Service Provisioning). This work was supported in part by the IITP (2020-0-00395-003, Standard Development of Blockchain based Network Management Automation Technology).

Appendix B. Contributors

The following are co-authors of this document:

Patrick Lingga - Department of Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seo-ro Jangan-gu, Suwon, Gyeonggi-do 16419, Republic of Korea. EMail: patricklink@skku.edu

Jinyong Tim Kim - Department of Electronic, Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seo-ro Jangan-gu, Suwon, Gyeonggi-do 16419, Republic of Korea. EMail: timkim@skku.edu

Hyoungshick Kim - Department of Computer Science and Engineering, Sungkyunkwan University, 2066 Seo-ro Jangan-gu, Suwon, Gyeonggi-do 16419, Republic of Korea. EMail: hyoung@skku.edu

Eunsoo Kim - Department of Electronic, Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seo-ro Jangan-gu, Suwon, Gyeonggi-do 16419, Republic of Korea. EMail: eskim86@skku.edu

Seungjin Lee - Department of Electronic, Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seo-ro Jangan-gu, Suwon, Gyeonggi-do 16419, Republic of Korea. EMail: jine33@skku.edu

Anil Lohiya - Juniper Networks, 1133 Innovation Way, Sunnyvale, CA 94089, US. EMail: alohiya@juniper.net

Dave Qi - Bloomberg, 731 Lexington Avenue, New York, NY 10022, US. EMail: DQI@bloomberg.net

Nabil Bitar - Nokia, 755 Ravendale Drive, Mountain View, CA 94043, US. EMail: nabil.bitar@nokia.com

Senad Palislamovic - Nokia, 755 Ravendale Drive, Mountain View, CA 94043, US. EMail: senad.palislamovic@nokia.com

Liang Xia - Huawei, 101 Software Avenue, Nanjing, Jiangsu 210012, China. EMail: Frank.Xialiang@huawei.com

Appendix C. Changes from draft-ietf-i2nsf-consumer-facing-interface-dm-28

The following changes are made from draft-ietf-i2nsf-consumer-facing-interface-dm-28:

- This version has reflected the comments of Erik Kline, Robert Wilton, Andrew Alston, and Paul Wouters during the IESG Evaluation.

*This version has also reflected the comments of Dirk Von Hugo as the Internet Area Directorate reviewer.

Authors' Addresses

Jaehoon Paul Jeong (editor)
Department of Computer Science and Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea

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

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

Phone: +82 31 299 4957
Email: darkhong@skku.edu

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
United States of America

Email: rkkumar@juniper.net

Susan Hares
Huawei

7453 Hickory Hill
Saline, MI 48176
United States of America

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