Workgroup: I2NSF Working Group
Internet-Draft:
draft-ietf-i2nsf-nsf-facing-interface-dm-29
Published: 1 June 2022
Intended Status: Standards Track
Expires: 3 December 2022
Authors: J. Kim, Ed.
Sungkyunkwan University
J. Jeong, Ed.
Sungkyunkwan University
J. Park
ETRI
S. Hares
Huawei
Q. Lin
Huawei

I2NSF Network Security Function-Facing Interface YANG Data Model

Abstract

This document defines a YANG data model for configuring security policy rules on Network Security Functions (NSF) in the Interface to Network Security Functions (I2NSF) framework. The YANG data model in this document is for the NSF-Facing Interface between a Security Controller and NSFs in the I2NSF framework. It is built on the basis of the YANG data model in the I2NSF Capability YANG Data Model document for the I2NSF framework.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 December 2022.

respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

This document defines a YANG [RFC6020][RFC7950] data model for security policy rule configuration of Network Security Functions (NSF). The YANG data model in this document is based on the data model described in [I-D.ietf-i2nsf-capability-data-model] for the NSF-Facing Interface in the Interface to Network Security Functions (I2NSF) architecture [RFC8329]. The YANG data model in this document focuses on security policy configuration for the NSFs discussed in [I-D.ietf-i2nsf-capability-data-model], i.e., generic NSF (operate on packet header for layer 2, layer3, and layer 4) and advanced NSF (Intrusion Prevention System, URL-Filtering, anti-DDoS, Antivirus, and VoIP/VoCN Filter). Note: VoIP is an abbreviation for Voice over Internet Protocol and VoCN is an abbreviation for Voice over Cellular Network, such as Voice over LTE or 5G.

This YANG data model uses an "Event-Condition-Action" (ECA) policy model that is used as the basis for the design of I2NSF Policy described in [RFC8329] and [I-D.ietf-i2nsf-capability-data-model].

The "ietf-i2nsf-nsf-facing-interface" YANG module defined in this document provides the configuration of the following features.

*A security policy rule of a network security function.

*An event clause of a generic network security function.

*A condition clause of a generic network security function.

*An action clause of a generic network security function.

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

This section shows a YANG tree diagram of policy for network security functions.

3.1. General I2NSF Security Policy Rule

This section shows a YANG tree diagram for a general I2NSF security policy rule for generic network security functions.

```
module: ietf-i2nsf-nsf-facing-interface
  +--rw i2nsf-security-policy* [name]
     +--rw name                    string
     +--rw language?               string
     +--rw priority-usage?         identityref
     +--rw resolution-strategy?    identityref
     +--rw default-action?         identityref
     +--rw rules* [name]
     |  +--rw name                 string
     |  +--rw description?         string
     |  +--rw priority?            uint8
     |  +--rw enable?              boolean
     |  +--rw long-connection
     |  |  +--rw enable?     boolean
     |  |  +--rw duration?   uint32
     |  +--rw event
     |  |  ...
     |  +--rw condition
     |  |  ...
     |  +--rw action
     |     ...
     +--rw rule-group
        +--rw groups* [group-name]
           +--rw group-name     string
           +--rw rule-name*     -> ../../../rules/name
           +--rw enable?        boolean
           +--rw description?   string
```

Figure 1: YANG Tree Diagram for Network Security Policy

A security policy is used by one virtual instance of an NSF/device as a set of security rules to protect assets from major risk factors that threaten the system. There can be multiple security policies in a single NSF to provide the necessary protection. The security policy includes its name, language tag, priority usage, resolution strategy, default action, and rules.

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".

A resolution strategy is used to decide how to resolve conflicts that occur between the actions of the same or different policy rules that are matched and contained in a particular NSF. The resolution strategy is defined as First Matching Rule (FMR), Last Matching Rule (LMR), Prioritized Matching Rule (PMR) with Errors (PMRE), and Prioritized Matching Rule with No Errors (PMRN). The resolution strategy can be extended according to specific vendor action

features. The resolution strategy is described in detail in [I-D.ietf-i2nsf-capability-data-model].

A default action is used to execute I2NSF policy rule when no rule matches a packet. The default action can be pass, drop, reject, rate-limit, or mirror actions. The default action can be extended according to specific vendor action features. The default action is described in detail in [I-D.ietf-i2nsf-capability-data-model].

The rules include rule name, rule description, rule priority, rule enable, event, condition, and action.

3.2. Event Clause

This section shows a YANG tree diagram for an event clause for a general I2NSF security policy rule for generic network security functions.

```
module: ietf-i2nsf-nsf-facing-interface
  +--rw i2nsf-security-policy* [name]
     ...
     +--rw rules* [name]
     |  ...
     |  +--rw event
     |  |  +--rw description?    string
     |  |  +--rw system-event*   identityref
     |  |  +--rw system-alarm*   identityref
     |  +--rw condition
     |  |  ...
     |  +--rw action
     |     ...
     +--rw rule-group
        ...
```

Figure 2: YANG Tree Diagram for an Event Clause

An event clause is any important occurrence at a specific time of a change in the system being managed, and/or in the environment of the system being managed. An event clause is used to trigger the evaluation of the condition clause of the I2NSF Policy Rule. The event clause is defined as a system event, system alarm [I-D.ietf-i2nsf-nsf-monitoring-data-model], and time. The event clause can be extended according to specific vendor event features. The event clause is described in detail in [I-D.ietf-i2nsf-capability-data-model].

3.3. Condition Clause

This section shows a YANG tree diagram for a condition clause for a general I2NSF security policy rule for generic network security functions.

```
module: ietf-i2nsf-nsf-facing-interface
  +--rw i2nsf-security-policy* [name]
     ...
     +--rw rules* [name]
     |  ...
     |  +--rw event
     |  |  ...
     |  +--rw condition
     |  |  +--rw description?    string
     |  |  +--rw layer-2* [destination-mac-address source-mac-address
                           ethertype]
     |  |  |  +--rw description?                    string
     |  |  |  +--rw destination-mac-address         yang:mac-address
     |  |  |  +--rw destination-mac-address-mask?   yang:mac-address
     |  |  |  +--rw source-mac-address              yang:mac-address
     |  |  |  +--rw source-mac-address-mask?        yang:mac-address
     |  |  |  +--rw ethertype                       eth:ethertype
     |  |  +--rw (layer-3)?
     |  |  |  +--:(ipv4)
     |  |  |  |  +--rw ipv4
     |  |  |  |     +--rw description?                     string
     |  |  |  |     +--rw dscp?                            inet:dscp
     |  |  |  |     +--rw ecn?                             uint8
     |  |  |  |     +--rw length?                          uint16
     |  |  |  |     +--rw ttl?                             uint8
     |  |  |  |     +--rw protocol?                        uint8
     |  |  |  |     +--rw ihl?                             uint8
     |  |  |  |     +--rw flags?                           bits
     |  |  |  |     +--rw offset?                          uint16
     |  |  |  |     +--rw identification?                  uint16
     |  |  |  |     +--rw (destination-network)?
     |  |  |  |     |  +--:(destination-ipv4-network)
     |  |  |  |     |  |  +--rw destination-ipv4-network?
                                                 inet:ipv4-prefix
     |  |  |  |     |  +--:(destination-ipv4-range)
     |  |  |  |     |     +--rw destination-ipv4-range* [start end]
     |  |  |  |     |        +--rw start    inet:ipv4-address-no-zone
     |  |  |  |     |        +--rw end      inet:ipv4-address-no-zone
     |  |  |  |     +--rw (source-network)?
     |  |  |  |        +--:(source-ipv4-network)
     |  |  |  |        |  +--rw source-ipv4-network? inet:ipv4-prefix
     |  |  |  |        +--:(source-ipv4-range)
     |  |  |  |           +--rw source-ipv4-range* [start end]
     |  |  |  |              +--rw start    inet:ipv4-address-no-zone
     |  |  |  |              +--rw end      inet:ipv4-address-no-zone
     |  |  |  +--:(ipv6)
     |  |  |     +--rw ipv6
     |  |  |        +--rw description?                  string
     |  |  |        +--rw dscp?                         inet:dscp
```

```
|  |  |        +--rw ecn?                                uint8
|  |  |        +--rw length?                             uint16
|  |  |        +--rw ttl?                                uint8
|  |  |        +--rw protocol?                           uint8
|  |  |        +--rw (destination-network)?
|  |  |        |  +--:(destination-ipv6-network)
|  |  |        |  |  +--rw destination-ipv6-network?
                                              inet:ipv6-prefix
|  |  |        |  +--:(destination-ipv6-range)
|  |  |        |     +--rw destination-ipv6-range* [start end]
|  |  |        |        +--rw start    inet:ipv6-address-no-zone
|  |  |        |        +--rw end      inet:ipv6-address-no-zone
|  |  |        +--rw (source-network)?
|  |  |        |  +--:(source-ipv6-network)
|  |  |        |  |  +--rw source-ipv6-network? inet:ipv6-prefix
|  |  |        |  +--:(source-ipv6-range)
|  |  |        |     +--rw source-ipv6-range* [start end]
|  |  |        |        +--rw start    inet:ipv6-address-no-zone
|  |  |        |        +--rw end      inet:ipv6-address-no-zone
|  |  |        +--rw flow-label?      inet:ipv6-flow-label
|  |  +--rw (layer-4)?
|  |  |  +--:(tcp)
|  |  |  |  +--rw tcp
|  |  |  |     +--rw description?              string
|  |  |  |     +--rw source-port-number
|  |  |  |     |  +--rw (source-port)?
|  |  |  |     |     +--:(range-or-operator)
|  |  |  |     |     |  +--rw (port-range-or-operator)?
|  |  |  |     |     |     +--:(range)
|  |  |  |     |     |     |  +--rw lower-port  inet:port-number
|  |  |  |     |     |     |  +--rw upper-port  inet:port-number
|  |  |  |     |     |     +--:(operator)
|  |  |  |     |     |        +--rw operator?     operator
|  |  |  |     |     |        +--rw port        inet:port-number
|  |  |  |     |     +--:(port-list)
|  |  |  |     |        +--rw port-numbers* [start end]
|  |  |  |     |           +--rw start    inet:port-number
|  |  |  |     |           +--rw end      inet:port-number
|  |  |  |     +--rw destination-port-number
|  |  |  |     |  +--rw (destination-port)?
|  |  |  |     |     +--:(range-or-operator)
|  |  |  |     |     |  +--rw (port-range-or-operator)?
|  |  |  |     |     |     +--:(range)
|  |  |  |     |     |     |  +--rw lower-port  inet:port-number
|  |  |  |     |     |     |  +--rw upper-port  inet:port-number
|  |  |  |     |     |     +--:(operator)
|  |  |  |     |     |        +--rw operator?     operator
|  |  |  |     |     |        +--rw port        inet:port-number
|  |  |  |     |     +--:(port-list)
```

```
|  |  |  |     |        +--rw port-numbers* [start end]
|  |  |  |     |           +--rw start    inet:port-number
|  |  |  |     |           +--rw end      inet:port-number
|  |  |  |     +--rw sequence-number?           uint32
|  |  |  |     +--rw acknowledgement-number?    uint32
|  |  |  |     +--rw data-offset?               uint8
|  |  |  |     +--rw reserved?                  uint8
|  |  |  |     +--rw flags?                     bits
|  |  |  |     +--rw window-size?               uint16
|  |  |  |     +--rw urgent-pointer?            uint16
|  |  |  |     +--rw options?                   binary
|  |  |  |  +--:(udp)
|  |  |  |  |  +--rw udp
|  |  |  |     +--rw description?               string
|  |  |  |     +--rw source-port-number
|  |  |  |     |  +--rw (source-port)?
|  |  |  |     |     +--:(range-or-operator)
|  |  |  |     |     |  +--rw (port-range-or-operator)?
|  |  |  |     |     |     +--:(range)
|  |  |  |     |     |     |  +--rw lower-port  inet:port-number
|  |  |  |     |     |     |  +--rw upper-port  inet:port-number
|  |  |  |     |     |     +--:(operator)
|  |  |  |     |     |        +--rw operator?     operator
|  |  |  |     |     |        +--rw port          inet:port-number
|  |  |  |     |     +--:(port-list)
|  |  |  |     |        +--rw port-numbers* [start end]
|  |  |  |     |           +--rw start    inet:port-number
|  |  |  |     |           +--rw end      inet:port-number
|  |  |  |     +--rw destination-port-number
|  |  |  |     |  +--rw (destination-port)?
|  |  |  |     |     +--:(range-or-operator)
|  |  |  |     |     |  +--rw (port-range-or-operator)?
|  |  |  |     |     |     +--:(range)
|  |  |  |     |     |     |  +--rw lower-port  inet:port-number
|  |  |  |     |     |     |  +--rw upper-port  inet:port-number
|  |  |  |     |     |     +--:(operator)
|  |  |  |     |     |        +--rw operator?     operator
|  |  |  |     |     |        +--rw port          inet:port-number
|  |  |  |     |     +--:(port-list)
|  |  |  |     |        +--rw port-numbers* [start end]
|  |  |  |     |           +--rw start    inet:port-number
|  |  |  |     |           +--rw end      inet:port-number
|  |  |  |     +--rw length?                    uint16
|  |  |  |  +--:(sctp)
|  |  |  |  |  +--rw sctp
|  |  |  |     +--rw description?               string
|  |  |  |     +--rw source-port-number
|  |  |  |     |  +--rw (source-port)?
|  |  |  |     |     +--:(range-or-operator)
```

```
|  |  |  |     |     |  +--rw (port-range-or-operator)?
|  |  |  |     |     |     +--:(range)
|  |  |  |     |     |     |  +--rw lower-port  inet:port-number
|  |  |  |     |     |     |  +--rw upper-port  inet:port-number
|  |  |  |     |     |     +--:(operator)
|  |  |  |     |     |        +--rw operator?     operator
|  |  |  |     |     |        +--rw port        inet:port-number
|  |  |  |     |     +--:(port-list)
|  |  |  |     |        +--rw port-numbers* [start end]
|  |  |  |     |           +--rw start    inet:port-number
|  |  |  |     |           +--rw end      inet:port-number
|  |  |  |     +--rw destination-port-number
|  |  |  |     |  +--rw (destination-port)?
|  |  |  |     |     +--:(range-or-operator)
|  |  |  |     |     |  +--rw (port-range-or-operator)?
|  |  |  |     |     |     +--:(range)
|  |  |  |     |     |     |  +--rw lower-port  inet:port-number
|  |  |  |     |     |     |  +--rw upper-port  inet:port-number
|  |  |  |     |     |     +--:(operator)
|  |  |  |     |     |        +--rw operator?     operator
|  |  |  |     |     |        +--rw port        inet:port-number
|  |  |  |     |     +--:(port-list)
|  |  |  |     |        +--rw port-numbers* [start end]
|  |  |  |     |           +--rw start    inet:port-number
|  |  |  |     |           +--rw end      inet:port-number
|  |  |  |     +--rw chunk-type*               uint8
|  |  |  |     +--rw chunk-length?             uint16
|  |  |  +--:(dccp)
|  |  |  |  +--rw dccp
|  |  |  |     +--rw description?              string
|  |  |  |     +--rw source-port-number
|  |  |  |     |  +--rw (source-port)?
|  |  |  |     |     +--:(range-or-operator)
|  |  |  |     |     |  +--rw (port-range-or-operator)?
|  |  |  |     |     |     +--:(range)
|  |  |  |     |     |     |  +--rw lower-port  inet:port-number
|  |  |  |     |     |     |  +--rw upper-port  inet:port-number
|  |  |  |     |     |     +--:(operator)
|  |  |  |     |     |        +--rw operator?     operator
|  |  |  |     |     |        +--rw port        inet:port-number
|  |  |  |     |     +--:(port-list)
|  |  |  |     |        +--rw port-numbers* [start end]
|  |  |  |     |           +--rw start    inet:port-number
|  |  |  |     |           +--rw end      inet:port-number
|  |  |  |     +--rw destination-port-number
|  |  |  |     |  +--rw (destination-port)?
|  |  |  |     |     +--:(range-or-operator)
|  |  |  |     |     |  +--rw (port-range-or-operator)?
|  |  |  |     |     |     +--:(range)
```

```
|  |  |  |     |     |     |  +--rw lower-port  inet:port-number
|  |  |  |     |     |     |  +--rw upper-port  inet:port-number
|  |  |  |     |     |     +--:(operator)
|  |  |  |     |     |        +--rw operator?     operator
|  |  |  |     |     |        +--rw port        inet:port-number
|  |  |  |     |     +--:(port-list)
|  |  |  |     |        +--rw port-numbers* [start end]
|  |  |  |     |           +--rw start    inet:port-number
|  |  |  |     |           +--rw end      inet:port-number
|  |  |  |     +--rw service-code*              uint32
|  |  |  |     +--rw type*                      uint8
|  |  |  |     +--rw data-offset?               uint8
|  |  |  +--:(icmp)
|  |  |     +--rw icmp
|  |  |        +--rw description?      string
|  |  |        +--rw version?          enumeration
|  |  |        +--rw type?             uint8
|  |  |        +--rw code?             uint8
|  |  |        +--rw rest-of-header?   binary
|  |  +--rw url-category
|  |  |  +--rw description?    string
|  |  |  +--rw pre-defined*    string
|  |  |  +--rw user-defined*   string
|  |  +--rw voice
|  |  |  +--rw description?            string
|  |  |  +--rw source-voice-id*        string
|  |  |  +--rw destination-voice-id*   string
|  |  |  +--rw user-agent*             string
|  |  +--rw ddos
|  |  |  +--rw description?         string
|  |  |  +--rw alert-packet-rate?   uint32
|  |  |  +--rw alert-flow-rate?     uint32
|  |  |  +--rw alert-byte-rate?     uint32
|  |  +--rw anti-virus
|  |  |  +--rw profile*           string
|  |  |  +--rw exception-files*   string
|  |  +--rw payload
|  |  |  +--rw description?   string
|  |  |  +--rw content*       binary
|  |  +--rw context
|  |     +--rw description?           string
|  |     +--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*        string
|  |     |  +--rw frequency?         enumeration
|  |     +--rw application
|  |     |  +--rw description?   string
|  |     |  +--rw protocol*      identityref
|  |     +--rw device-type
|  |     |  +--rw description?   string
|  |     |  +--rw device*        identityref
|  |     +--rw users
|  |     |  +--rw description?   string
|  |     |  +--rw user* [id]
|  |     |  |  +--rw id      uint32
|  |     |  |  +--rw name?   string
|  |     |  +--rw group* [id]
|  |     |     +--rw id      uint32
|  |     |     +--rw name?   string
|  |     +--rw geographic-location
|  |        +--rw description?   string
|  |        +--rw source*        string
|  |        +--rw destination*   string
|  +--rw action
|     ...
+--rw rule-group
   ...
```

Figure 3: YANG Tree Diagram for a Condition Clause

A condition clause is defined as a set of attributes, features, and/or values that are to be compared with a set of known attributes, features, and/or values in order to determine whether the set of actions in that (imperative) I2NSF policy rule can be executed or not. A condition clause works with 'AND' logic, where all fields set in the condition MUST match the packet or flow for the condition to be evaluated as 'TRUE'. A condition clause is classified as a condition of generic network security functions, advanced network security functions, or context. A condition clause of generic network security functions is defined as IPv4 condition, IPv6 condition, TCP condition, UDP condition, SCTP condition, DCCP condition, or ICMP (ICMPv4 and ICMPv6) condition.

Note that the data model in this document does not focus on only IP addresses, but focuses on all the fields of IPv4 and IPv6 headers. The IPv4 and IPv6 headers have similarity with some different fields. In this case, it is better to handle separately the IPv4 and IPv6 headers such that the different fields can be used to handle IPv4 and IPv6 packets. Also, note that the YANG data model in this document is based on the YANG Data Model for Network Access Control Lists (ACLs) [RFC8519] that does not support IPv6 extension headers including various options, the support of IPv6 extension headers is left as future work.

The data model provides transport layer condition for TCP, UDP, SCTP, and DCCP. With ICMPv4 and ICMPv6 are included as a choice for layer 4 as the header fields in ICMP are above the network layer. 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 and will be considered in the future I2NSF documents.

A condition clause of advanced network security functions is defined as url category condition, voice condition, DDoS condition, or payload condition. A condition clause of context is defined as application condition, target condition, users condition, and geography condition.

Note that this document deals only with conditions of several advanced network security functions such as url filter (i.e., web filter), VoIP/VoCN security, and DDoS-attack mitigator. A condition clause of other advanced network security functions such as Intrusion Prevention System (IPS) and Data Loss Prevention (DLP) can be defined as an extension in future. A condition clause can be extended according to specific vendor condition features. A condition clause is described in detail in [I-D.ietf-i2nsf-capability-data-model].

3.4. Action Clause

This section shows a YANG tree diagram for an action clause for a general I2NSF security policy rule for generic network security functions.

```
module: ietf-i2nsf-nsf-facing-interface
  +--rw i2nsf-security-policy* [name]
     ...
     +--rw rules* [name]
     |  ...
     |  +--rw event
     |  ...
     |  +--rw condition
     |  ...
     |  +--rw action
     |     +--rw description?       string
     |     +--rw packet-action
     |     |  +--rw ingress-action?   identityref
     |     |  +--rw egress-action?    identityref
     |     |  +--rw log-action?       identityref
     |     +--rw flow-action
     |     |  +--rw ingress-action?   identityref
     |     |  +--rw egress-action?    identityref
     |     |  +--rw log-action?       identityref
     |     +--rw advanced-action
     |        +--rw content-security-control*    identityref
     |        +--rw attack-mitigation-control*   identityref
     +--rw rule-group
        ...
```

Figure 4: YANG Tree Diagram for an Action Clause

An action is used to control and monitor aspects of flow-based NSFs when the policy rule event and condition clauses are satisfied. NSFs provide security services by executing various actions. The action clause is defined as ingress action, egress action, or log action for packet action, flow action, and advanced action for additional inspection. The packet action is an action for an individual packet such as an IP datagram as a stateless process that uses the packet's header and payload. The flow action is an action of a traffic flow such as the packets of a TCP session (e.g., an HTTP/HTTPS session) as a stateful process that uses the traffic flow information such as 5-tuple information, packet counts, and byte counts. The advanced action is an action for an advanced security service (e.g., url filter, DDoS-attack mitigator, and VoIP/VoCN filter) for either a packet or a traffic flow according to the intention of such an advanced security service. The action clause can be extended

according to specific vendor action features. The action clause is described in detail in [I-D.ietf-i2nsf-capability-data-model].

Note that an empty event clause means that the event boolean will always evaluate to true and starts the evaluation of the condition clause, while an empty condition clause means that the condition boolean will always evaluate to false.

4. YANG Data Model of NSF-Facing Interface

The main objective of this document is to provide the YANG data model of the I2NSF NSF-Facing Interface. This interface can be used to deliver control and management messages between a Security Controller and NSFs for the I2NSF low-level security policies.

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 NSF-Facing Interface, this document suggests use cases for security policy rules such as time-based firewall, web filter, VoIP/VoCN security service, and DDoS-attack mitigation in Section 5.

4.1. YANG Module of NSF-Facing Interface

This section describes a YANG module of NSF-Facing Interface. This document provides identities in the data model for the configuration of an NSF. The identity has the same concept with the corresponding identity in [I-D.ietf-i2nsf-consumer-facing-interface-dm]. This YANG module imports from [RFC6991] and [RFC8519]. It makes references to [RFC0768] [RFC0791] [RFC0792] [RFC0854] [RFC0959] [RFC1939] [RFC2132] [RFC2595] [RFC3261] [RFC3986] [RFC4250] [RFC4340] [RFC4443] [RFC4732] [RFC4987] [RFC5321] [RFC5595] [RFC5646] [RFC6335] [RFC8075] [RFC8200] [RFC8329] [RFC8335] [RFC9051] [RFC9179] [GLOB] [IEEE-802.3] [ISO-3166] [I-D.ietf-httpbis-http2bis] [I-D.ietf-httpbis-messaging] [I-D.ietf-httpbis-semantics] [I-D.ietf-i2nsf-capability-data-model] [I-D.ietf-i2nsf-nsf-monitoring-data-model] [I-D.ietf-tcpm-rfc793bis] [I-D.ietf-tsvwg-rfc4960-bis]

```
<CODE BEGINS> file "ietf-i2nsf-nsf-facing-interface@2022-06-01.yang"

module ietf-i2nsf-nsf-facing-interface {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface";
  prefix
    i2nsfnfi;

  import ietf-inet-types {
    prefix inet;
    reference
      "Section 4 of RFC 6991";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "Section 3 of RFC 6991";
  }
  import ietf-packet-fields {
    prefix packet-fields;
    reference
      "Section 4.2 of RFC 8519";
  }

  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: Jinyong Tim Kim
     <mailto:timkim@skku.edu>

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

  description
    "This module is a YANG module for Network Security Functions
     (NSF)-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) 2022 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.";

revision "2022-06-01"{
  description "The latest revision.";
  reference
    "RFC XXXX: I2NSF Network Security Function-Facing Interface
     YANG Data Model";
}

/*
 * Identities
 */

identity priority-usage {
  description
    "Base identity for priority usage type to define the type of
     priority to be implemented in a security policy rule, such
     as priority by order and priority by number.";
}

identity priority-by-order {
  base priority-usage;
  description
    "This indicates that the priority of a security policy rule
     follows the order of the configuration. The earlier the
     configuration is, the higher the priority is.";
}

identity priority-by-number {
  base priority-usage;
  description
    "This indicates the priority of a security policy rule follows
     the number or value of the configuration. The higher the value
     is, the higher the priority is.";
}

identity event {
  description
```

```
    "Base identity for policy events.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
     Monitoring Interface YANG Data Model - Event";
}

identity system-event {
  base event;
  description
    "Base Identity for system events. 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.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
     Monitoring Interface YANG Data Model - System event";
}

identity system-alarm {
  base event;
  description
    "Base identity for system alarms. System alarm is defined as a
     warning related to service degradation in system hardware.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
     Monitoring Interface YANG Data Model - System alarm";
}

identity access-violation {
  base system-event;
  description
    "Access-violation system event is an event when a user tries
     to access (read, write, create, or delete) any information or
     execute commands above their privilege (i.e., not-conformant
     with the access profile).";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
     Monitoring Interface YANG Data Model - System event for access
     violation";
}

identity configuration-change {
  base system-event;
  description
    "The configuration-change system event is an event when a user
     adds a new configuration or modify an existing configuration
     (write configuration).";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
```

```
     Monitoring Interface YANG Data Model - System event for
     configuration change";
}

identity memory-alarm {
  base system-alarm;
  description
    "Memory is the hardware to store information temporarily or for
     a short period, i.e., Random Access Memory (RAM). A
     memory-alarm is emitted when the memory usage is exceeding
     the threshold.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
     Monitoring Interface YANG Data Model - System alarm for
     memory";
}

identity cpu-alarm {
  base system-alarm;
  description
    "CPU is the Central Processing Unit that executes basic
     operations of the system. A cpu-alarm is emitted when the CPU
     usage is exceeding a threshold.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
     Monitoring Interface YANG Data Model - System alarm for CPU";
}

identity disk-alarm {
  base system-alarm;
  description
    "Disk or storage is the hardware to store information for a
     long period, i.e., Hard Disk and Solid-State Drive. A
     disk-alarm is emitted when the disk usage is exceeding a
     threshold.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
     Monitoring Interface YANG Data Model - System alarm for disk";
}

identity hardware-alarm {
  base system-alarm;
  description
    "A hardware alarm is emitted when a hardware failure (e.g.,
     CPU, memory, disk, or interface) is detected. A hardware
     failure is a malfunction within the electronic circuits or
     electromechanical components of the hardware that makes it
     unusable.";
  reference
```

```
    "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
     Monitoring Interface YANG Data Model - System alarm for
     hardware";
}

identity interface-alarm {
  base system-alarm;
  description
    "Interface is the network interface for connecting a device
     with the network. The interface-alarm is emitted when the
     state of the interface is changed.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
     Monitoring Interface YANG Data Model - System alarm for
     interface";
}

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. Note that the device type of either a source
     or destination can be known with the help of DHCP
     Fingerprinting and the interaction between an NSF and a DHCP
     server.";
  reference
    "draft-ietf-i2nsf-capability-data-model-32: I2NSF Capability
     YANG Data Model
     RFC 2132: DHCP Options and BOOTP Vendor Extensions - Vendor
     Specific Information including device type, manufacturer,
     and operating system as DHCP fingerprinting information";
}


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.";
}

identity advanced-nsf {
```

```
  description
    "Base identity for advanced Network Security Function (NSF)
     capability.  This can be used for advanced NSFs such as
     Anti-DDoS Attack, IPS, URL-Filtering, Antivirus,
     and VoIP/VoCN Filter.";
  reference
    "draft-ietf-i2nsf-capability-data-model-32:
     I2NSF Capability YANG Data Model";
}

identity content-security-control {
  base advanced-nsf;
  description
    "Base identity for content security control. Content security
     control is an NSF that evaluates the payload of a packet,
     such as Intrusion Prevention System (IPS), URL Filter,
     Antivirus, and VoIP/VoCN Filter.";
  reference
    "draft-ietf-i2nsf-capability-data-model-32:
     I2NSF Capability YANG Data Model";
}

identity ips {
  base content-security-control;
  description
    "IPS (Intrusion Prevention System) prevents malicious activity
     within a network";
}

identity url-filtering {
  base content-security-control;
  description
    "URL filtering limits access by comparing the web traffic's
     URL with the URLs for web filtering in a database";
}

identity anti-virus {
  base content-security-control;
  description
    "Antivirus to protect the network by detecting and
     removing viruses or malwares.";
}

identity voip-vocn-filtering {
  base content-security-control;
  description
    "VoIP (Voice over Internet Protocol) and VoCN (Voice over
     Cellular  Network, such as Voice over LTE or 5G) security
     service that filters out the packets or flows of malicious
```

```
     users with a deny-list of malicious users in a database";
}

identity attack-mitigation-control {
  base advanced-nsf;
  description
    "Base identity for attack mitigation control. Attack mitigation
     control is an NSF that mitigates an attack such as
     anti-DDoS (i.e., DDoS-mitigator).";
  reference
    "draft-ietf-i2nsf-capability-data-model-32:
     I2NSF Capability YANG Data Model";
}

identity anti-ddos {
  base attack-mitigation-control;
  description
    "Anti-DDoS or DDoS Mitigator to protect a server or network
     from a DDoS attack. The mitigation approach is up to the
     implementation.";
  reference
    "RFC 4732: Internet Denial-of-Service Considerations - DoS
     Mitigation Strategies
     RFC 4987: TCP SYN Flooding Attacks and Common Mitigations -
     Common Defenses";
}

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

identity ingress-action {
  base action;
  description
    "Base identity for ingress action. The action 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 action;
  description
    "Base identity for egress action. The action 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 default-action {
  base action;
  description
    "Base identity for default action. The default action of the
     NSF when no rule matches the packet or flow.";
  reference
    "draft-ietf-i2nsf-capability-data-model-32:
     I2NSF Capability YANG Data Model - Default Action";
}

identity pass {
  base ingress-action;
  base egress-action;
  base default-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;
  base default-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;
  base default-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 TCP packet is rejected with
     TCP RST response or a UDP packet may be rejected with an
     ICMPv4 response message with Type 3 Code 3 or ICMPv6 response
     message Type 1 Code 4 (i.e., Destination Unreachable:
     Destination port unreachable).";
}

identity mirror {
  base ingress-action;
  base egress-action;
  base default-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;
  base default-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 log-action {
  base 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 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 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 application-protocol {
  description
    "Base identity for Application protocol. Note that a subset of
     application protocols (e.g., HTTP, HTTPS, FTP, POP3, and
```

```
     IMAP) are handled in this YANG module, rather than all
     the existing application protocols.";
}

identity http {
  base application-protocol;
  description
    "The identity for Hypertext Transfer Protocol version 1.1
     (HTTP/1.1).";
  reference
    "draft-ietf-httpbis-semantics-19: HTTP Semantics
     draft-ietf-httpbis-messaging-19: HTTP/1.1";
}

identity https {
  base application-protocol;
  description
    "The identity for Hypertext Transfer Protocol version 1.1
     (HTTP/1.1) over TLS.";
  reference
    "draft-ietf-httpbis-semantics-19: HTTP Semantics
     draft-ietf-httpbis-messaging-19: HTTP/1.1";
}

identity http2 {
  base application-protocol;
  description
    "The identity for Hypertext Transfer Protocol version 2
     (HTTP/2).";
  reference
    "draft-ietf-httpbis-http2bis-07: HTTP/2";
}

identity https2 {
  base application-protocol;
  description
    "The identity for Hypertext Transfer Protocol version 2
     (HTTP/2) over TLS.";
  reference
    "draft-ietf-httpbis-http2bis-07: HTTP/2";
}

identity ftp {
  base application-protocol;
  description
    "The identity for File Transfer Protocol.";
  reference
    "RFC 959: File Transfer Protocol (FTP)";
}
```

```
identity ssh {
  base application-protocol;
  description
    "The identity for Secure Shell (SSH) protocol.";
  reference
    "RFC 4250: The Secure Shell (SSH) Protocol";
}

identity telnet {
  base application-protocol;
  description
    "The identity for telnet.";
  reference
    "RFC 854: Telnet Protocol";
}

identity smtp {
  base application-protocol;
  description
    "The identity for Simple Mail Transfer Protocol.";
  reference
    "RFC 5321: Simple Mail Transfer Protocol (SMTP)";
}

identity pop3 {
  base application-protocol;
  description
    "The identity for Post Office Protocol 3 (POP3).";
  reference
    "RFC 1939: Post Office Protocol - Version 3 (POP3)";
}

identity pop3s {
  base application-protocol;
  description
    "The identity for Post Office Protocol 3 (POP3) over TLS";
  reference
    "RFC 1939: Post Office Protocol - Version 3 (POP3)
     RFC 2595: Using TLS with IMAP, POP3 and ACAP";
}

identity imap {
  base application-protocol;
  description
    "The identity for Internet Message Access Protocol (IMAP).";
  reference
    "RFC 9051: Internet Message Access Protocol (IMAP) - Version
     4rev2";
```

```
}

identity imaps {
  base application-protocol;
  description
    "The identity for Internet Message Access Protocol (IMAP) over
     TLS";
  reference
    "RFC 9051: Internet Message Access Protocol (IMAP) - Version
     4rev2";
}

/*
 * Typedefs
 */

typedef time {
  type string {
    pattern '(0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9](\.\d+)?'
      + '(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 port-range {
  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 port numbers. If only one value is needed,
     then set both start and end to the same value.";
  reference
    "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
     (TCP) Specification - Port Number
     RFC 768: User Datagram Protocol - Port Number
     draft-ietf-tsvwg-rfc4960-bis-18: Stream Control Transmission
     Protocol - Port Number
     RFC 4340: Datagram Congestion Control Protocol (DCCP)
     - Port Number";
}

grouping ipv4-range {
  description
    "A range match for IPv4 addresses.  If only one value is
     needed, then set both start and end to the same value.
```

```
     The end IPv4 address MUST be equal to or greater than the
     start IPv4 address.";
  leaf start {
    type inet:ipv4-address-no-zone;
    description
      "A start IPv4 address for a range match.";
  }
  leaf end {
    type inet:ipv4-address-no-zone;
    description
      "An end IPv4 address for a range match.";
  }
  reference
    "RFC 791: Internet Protocol - IPv4 address";
}

grouping ipv6-range {
  description
    "A range match for IPv6 addresses.  If only one value is
     needed, then set both start and end to the same value.
     The end IPv6 address MUST be equal to or greater than the
     start IPv6 address.";
  leaf start {
    type inet:ipv6-address-no-zone;
    description
      "A start IPv6 address for a range match.";
  }

  leaf end {
    type inet:ipv6-address-no-zone;
    description
      "An end IPv6 address for a range match.";
  }
  reference
    "RFC 8200: Internet Protocol, Version 6 (IPv6)
     Specification - IPv6 address";
}

/*
 * Data nodes
 */

list i2nsf-security-policy {

  key "name";

  description
    "Container for security policy
     including a set of security rules according to certain logic,
```

```
   i.e., their similarity or mutual relations, etc. The network
   security policy can be applied to both the unidirectional
   and bidirectional traffic across the NSF.
   The I2NSF security policies use the Event-Condition-Action
   (ECA) policy model ";

reference
  "RFC 8329: Framework for Interface to Network Security
   Functions - I2NSF Flow Security Policy Structure
   draft-ietf-i2nsf-capability-data-model-32:
   I2NSF Capability YANG Data Model - Design Principles and
   ECA Policy Model Overview";

leaf name {
  type string;
  description
    "The name of the security policy.
     This must be unique.";
}

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-security-policy'. This field is mandatory only when
     one or more of the 'leaf description' is used.

     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 priority-usage {
  type identityref {
    base priority-usage;
  }
  default priority-by-order;
  description
    "Priority usage type for security policy rule:
     priority by order and priority by number";
}

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";
}

leaf default-action {
  type identityref {
    base default-action;
  }
  default mirror;
  description
    "This default action can be used to specify a predefined
     action when no other alternative action was matched
     by the currently executing I2NSF Policy Rule. An analogy
     is the use of a default statement in a C switch statement.";
  reference
```

```
    "draft-ietf-i2nsf-capability-data-model-32:
     I2NSF Capability YANG Data Model - Default Action";
}

list rules {
  key "name";
  description
    "This is a rule for network security functions.";

  leaf name {
    type string;
    description
      "The name of the rule.";
  }

  leaf description {
    type string;
    description
      "This description gives more information about
       rules.";
  }

  leaf priority {
    type uint8 {
      range "1..255";
    }
    description
      "The priority for the rule comes with a mandatory
       numeric value which can range from 1 up to 255.
       Note that a higher number means a higher priority";
  }

  leaf enable {
    type boolean;
    description
      "If true, the rule is enabled and enforced.
       If false, the rule is configured but disabled and not
       enforced.";
  }

  container long-connection {
    description
      "A container for long connection. A long connection is a
       connection that is maintained after the socket connection
       is established, regardless of whether it is used for data
       traffic or not.";

    leaf enable {
      type boolean;
```

```
    description
      "If true, the rule is enabled and enforced.
       If false, the rule is configured but disabled
       and not enforced.";
  }

  leaf duration {
    when "../enable = 'true'";
    type uint32;
    units "second";
    description
      "This is the maximum inactive connection duration of a
       long connection before a connection is declared as
       expired.";
  }
}

container event {
  description
    "An event is defined as any important
     occurrence in time of a change in the system being
     managed, and/or in the environment of the system being
     managed. When used in the context of policy rules for
     a flow-based NSF, it is used to determine whether the
     Condition clause of the Policy Rule can be evaluated
     or not. Examples of an I2NSF event include time and
     user actions (e.g., logon, logoff, and actions that
     violate any ACL.).";

  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - I2NSF Flow Security Policy Structure
     draft-ietf-i2nsf-capability-data-model-32:
     I2NSF Capability YANG Data Model - Design Principles and
     ECA Policy Model Overview
     draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF
     NSF Monitoring Interface YANG Data Model - Alarms,
     Events, Logs, and Counters";

  leaf description {
    type string;
    description
      "Description for an event clause";
  }

  leaf-list system-event {
    type identityref {
      base system-event;
    }
```

```
    description
      "The security policy rule according to
       system events.";
  }

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

container condition {
  description
    "A condition is defined as a set
     of attributes, features, and/or values that are to be
     compared with a set of known attributes, features,
     and/or values in order to determine whether the
     set of Actions in that (imperative) I2NSF Policy Rule
     can be executed or not. Examples of I2NSF Conditions
     include matching attributes of a packet or flow, and
     comparing the internal state of an NSF to a desired
     state.
     The condition works with 'AND' logic, where all
     fields set in a condition MUST match the packet or flow
     for the condition to be evaluated as 'TRUE'";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - I2NSF Flow Security Policy Structure
     draft-ietf-i2nsf-capability-data-model-32:
     I2NSF Capability YANG Data Model - Design Principles and
     ECA Policy Model Overview";

  leaf description {
    type string;
    description
      "Description for a condition clause.";
  }

  list layer-2 {
    key "destination-mac-address source-mac-address ethertype";
    description
      "The purpose of this container is to represent layer 2
       packet header information to determine the set of policy
       actions in this ECA policy rule should be executed or
       not.";
```

```
  reference
    "IEEE 802.3: IEEE Standard for Ethernet";

  leaf description {
    type string;
    description
      "The ethernet condition description";
  }

  uses packet-fields:acl-eth-header-fields;
}

choice layer-3 {
  case ipv4 {
    container ipv4 {
      description
        "The purpose of this container is to represent
         IPv4 packet header information to determine if
         the set of policy actions in this ECA policy rule
         should be executed or not.";
      reference
        "RFC 791: Internet Protocol";

      leaf description {
        type string;
        description
          "This is description for IPv4 condition.";
      }

      uses packet-fields:acl-ip-header-fields;
      uses packet-fields:acl-ipv4-header-fields {
        augment destination-network {
          case destination-ipv4-range {
            list destination-ipv4-range {
              key "start end";
              uses ipv4-range;
              description
                "The list of IPv4 addresses specified with
                 a start IPv4 address and an end IPv4
                 address. If only one value is needed, then
                 set both start and end to the same value.
                 Note that the 'end' IPv4 address MUST be
                 equal to or greater than the 'start' IPv4
                 address.";
            }
          }
          description
            "IPv4 destination network denoted as IPv4
             addresses";
```

```
      }
      augment source-network {
        case source-ipv4-range {
          list source-ipv4-range {
            key "start end";
            uses ipv4-range;
            description
              "The list of IPv4 addresses specified with
               a start IPv4 address and an end IPv4
               address. If only one value is needed, then
               set both start and end to the same value.
               Note that the 'end' IPv4 address MUST be
               equal or greater than the 'start' IPv4
               address.";
          }
        }
        description
          "IPv4 source network denoted as IPv4
           addresses";
      }
    }
  }
}
case ipv6 {
  container ipv6 {
    description
      "The purpose of this container is to represent IPv6
       packet header information to determine if the set
       of policy actions in this ECA policy rule should
       be executed or not.";

    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
       Specification";

    leaf description {
      type string;
      description
        "This is description for IPv6 condition.";
    }

    uses packet-fields:acl-ip-header-fields;
    uses packet-fields:acl-ipv6-header-fields {
      augment destination-network {
        case destination-ipv6-range {
          list destination-ipv6-range {
            key "start end";
            uses ipv6-range;
            description
```

```
                "The list of IPv6 addresses specified with
                 a start IPv6 address and an end IPv6
                 address. If only one value is needed, then
                 set both start and end to the same value.
                 Note that the 'end' IPv6 address MUST be
                 equal to or greater than the 'start' IPv6
                 address.";
              }
            }
            description
              "IPv6 destination network denoted as IPv6
               addresses";
          }
          augment source-network {
            case source-ipv6-range {
              list source-ipv6-range {
                key "start end";
                uses ipv6-range;
                description
                  "The list of IPv6 addresses specified with
                   a start IPv6 address and an end IPv6
                   address. If only one value is needed, then
                   set both start and end to the same value.
                   Note that the 'end' IPv6 address MUST be
                   equal to or greater than the 'start' IPv6
                   address.";
              }
            }
            description
              "IPv6 source network denoted as IPv6
               addresses";
          }
        }
      }
    }
    description
      "Choice of either IPv4 or IPv6 as layer-3 protocol";
  }

  choice layer-4 {
    case tcp {
      container tcp {
        description
          "The purpose of this container is to represent
           TCP packet header information to determine
           if the set of policy actions in this ECA policy
           rule should be executed or not.";
        reference
          "draft-ietf-tcpm-rfc793bis-25: Transmission Control
```

```
   Protocol (TCP) Specification";


leaf description {
  type string;
  description
   "This is description for tcp condition.";
}

container source-port-number {
  choice source-port {
    case range-or-operator {
      uses packet-fields:port-range-or-operator;
      description
        "Source port definition from range or operator.
         Can be used when a single port range to be
         specified.";
    }
    case port-list {
      list port-numbers {
        key "start end";
        uses port-range;
        description
          "List of source port numbers.";
      }
      description
        "Source port definition from list of port
         numbers. In the case of multiple port ranges
         needed to be specified.";
    }
    description
      "The choice of source port definition using
       range/operator or a choice to use list of port
       numbers.";
  }
  description
    "The security policy rule according to
     tcp source port number.";
  reference
    "draft-ietf-tcpm-rfc793bis-25: Transmission Control
     Protocol (TCP) Specification - Port Number";
}

container destination-port-number {
  choice destination-port {
    case range-or-operator {
      uses packet-fields:port-range-or-operator;
      description
        "Destination port definition from range or
```

```
            operator.
            Can be used when a single port range to be
            specified.";
        }
        case port-list {
          list port-numbers {
            key "start end";
            uses port-range;
            description
              "List of destination port numbers.";
          }
          description
            "Destination port definition from list of port
             numbers.
             In the case of multiple port ranges needed to
             be specified.";
        }
        description
          "The choice of destination port definition using
           range/operator or a choice to use list of port
           numbers.";
      }
      description
        "The security policy rule according to
         tcp destination port number.";
      reference
        "draft-ietf-tcpm-rfc793bis-25: Transmission Control
         Protocol (TCP) Specification - Port Number";
    }

    uses packet-fields:acl-tcp-header-fields;
  }
}

case udp {
  container udp {
    description
      "The purpose of this container is to represent
       UDP packet header information to determine
       if the set of policy actions in this ECA policy
       rule should be executed or not.";
    reference
      "RFC 768: User Datagram Protocol";

    leaf description {
      type string;
      description
       "This is description for udp condition.";
    }
```

```
container source-port-number {
  choice source-port {
    case range-or-operator {
      uses packet-fields:port-range-or-operator;
      description
        "Source port definition from range or operator.
         Can be used when a single port range to be
         specified.";
    }
    case port-list {
      list port-numbers {
        key "start end";
        uses port-range;
        description
          "List of source port numbers.";
      }
      description
        "Source port definition from list of port
         numbers. In the case of multiple port ranges
         needed to be specified.";
    }
    description
      "The choice of source port definition using
       range/operator or a choice to use list of port
       numbers.";
  }
  description
    "The security policy rule according to
     udp source port number.";
  reference
    "RFC 768: User Datagram Protocol - Port Number";
}

container destination-port-number {
  choice destination-port {
    case range-or-operator {
      uses packet-fields:port-range-or-operator;
      description
        "Destination port definition from range or
         operator.
         Can be used when a single port range to be
         specified.";
    }
    case port-list {
      list port-numbers {
        key "start end";
        uses port-range;
        description
```

```
              "List of destination port numbers.";
          }
          description
            "Destination port definition from list of port
             numbers.
             In the case of multiple port ranges needed to
             be specified.";
        }
        description
          "The choice of destination port definition using
           range/operator or a choice to use list of port
           numbers.";
      }
      description
        "The security policy rule according to
         udp destination port number.";
      reference
        "RFC 768: User Datagram Protocol - Port Number";
    }

    uses packet-fields:acl-udp-header-fields;
  }
}

case sctp {
  container sctp {
    description
      "The purpose of this container is to represent
       SCTP packet header information to determine
       if the set of policy actions in this ECA policy
       rule should be executed or not.";

    leaf description {
      type string;
      description
        "This is description for sctp condition.";
    }

    container source-port-number {
      choice source-port {
        case range-or-operator {
          uses packet-fields:port-range-or-operator;
          description
            "Source port definition from range or operator.
             Can be used when a single port range to be
             specified.";
        }
        case port-list {
          list port-numbers {
```

```
        key "start end";
        uses port-range;
        description
          "List of source port numbers.";
      }
      description
        "Source port definition from list of port
         numbers. In the case of multiple port ranges
         needed to be specified.";
    }
    description
      "The choice of source port definition using
       range/operator or a choice to use list of port
       numbers.";
  }
  description
    "The security policy rule according to
     sctp source port number.";
  reference
    "draft-ietf-tsvwg-rfc4960-bis-18: Stream Control
     Transmission Protocol - Port number";
}

container destination-port-number {
  choice destination-port {
    case range-or-operator {
      uses packet-fields:port-range-or-operator;
      description
        "Destination port definition from range or
         operator.
         Can be used when a single port range to be
         specified.";
    }
    case port-list {
      list port-numbers {
        key "start end";
        uses port-range;
        description
          "List of destination port numbers.";
      }
      description
        "Destination port definition from list of port
         numbers.
         In the case of multiple port ranges needed to
         be specified.";
    }
    description
      "The choice of destination port definition using
       range/operator or a choice to use list of port
```

```
            numbers.";
        }
        description
          "The security policy rule according to
           sctp destination port number.";
        reference
          "draft-ietf-tsvwg-rfc4960-bis-18: Stream Control
           Transmission Protocol - Port Number";
      }

      leaf-list chunk-type {
        type uint8;
        description
          "The security policy rule according to
           sctp chunk type ID Value.";
        reference
          "draft-ietf-tsvwg-rfc4960-bis-18: Stream Control
           Transmission Protocol - Chunk Type";
      }

      leaf chunk-length {
        type uint16 {
          range "4..max";
        }
        description
          "The security policy rule according to the length
           of  the chunk in sctp. This value represents the
           size of the chunk in bytes, including the Chunk
           Type, Chunk Flags, Chunk Length, and Chunk Value
           fields.";
        reference
          "draft-ietf-tsvwg-rfc4960-bis-18: Stream Control
           Transmission Protocol - Chunk Length";
      }
    }
  }

  case dccp {
    container dccp {
      description
        "The purpose of this container is to represent
         DCCP packet header information to determine
         if the set of policy actions in this ECA policy
         rule should be executed or not.";
      leaf description {
        type string;
        description
          "This is description for dccp condition.";
      }
```

```
container source-port-number {
  choice source-port {
    case range-or-operator {
      uses packet-fields:port-range-or-operator;
      description
        "Source port definition from range or operator.
         Can be used when a single port range to be
         specified.";
    }
    case port-list {
      list port-numbers {
        key "start end";
        uses port-range;
        description
          "List of source port numbers.";
      }
      description
        "Source port definition from list of port
         numbers. In the case of multiple port ranges
         needed to be specified.";
    }
    description
      "The choice of source port definition using
       range/operator or a choice to use list of port
       numbers.";
  }
  description
    "The security policy rule according to
     dccp source port number.";
  reference
    "RFC 4340: Datagram Congestion Control Protocol
               (DCCP) - Port number";
}

container destination-port-number {
  choice destination-port {
    case range-or-operator {
      uses packet-fields:port-range-or-operator;
      description
        "Destination port definition from range or
         operator.
         Can be used when a single port range to be
         specified.";
    }
    case port-list {
      list port-numbers {
        key "start end";
        uses port-range;
```

```
        description
          "List of destination port numbers.";
      }
      description
        "Destination port definition from list of port
         numbers. In the case of multiple port ranges
         needed to be specified.";
    }
    description
      "The choice of destination port definition using
       range/operator or a choice to use list of port
       numbers.";
  }
  description
    "The security policy rule according to
     dccp destination port number.";
  reference
    "RFC 4340: Datagram Congestion Control Protocol
               (DCCP) - Port number";
}

leaf-list service-code {
  type uint32;
  description
    "The security policy rule according to
     dccp service code.";
  reference
    "RFC 4340: Datagram Congestion Control Protocol
               (DCCP) - Service Codes
     RFC 5595: The Datagram Congestion Control Protocol
               (DCCP) Service Codes
     RFC 6335: Internet Assigned Numbers Authority
               (IANA) Procedures for the Management of
               the Service Name and Transport Protocol
               Port Number Registry - Service Code";
}

leaf-list type {
  type uint8 {
    range "0..15";
  }
  description
    "The security policy rule according to the 4 bits
     of dccp type header field for dccp packet types
     such as DCCP-Request, DCCP-Response, DCCP-Data,
     DCCP-Ack, and DCCP-DataAck.";
  reference
    "RFC 4340: Datagram Congestion Control Protocol
               (DCCP) - Packet Types";
```

```
    }

    leaf data-offset {
      type uint8;
      description
        "The security policy rule according to the offset
         from
         the start of the packet's DCCP header to the start
         of its application data area, in 32-bit word.";
      reference
        "RFC 4340: Datagram Congestion Control Protocol
                   (DCCP) - Data Offset";
    }
  }
}
case icmp {
  container icmp {
    description
      "The purpose of this container is to represent
       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 description {
      type string;
      description
       "This is description for icmp condition.";
    }

    leaf version {
      type enumeration {
        enum icmpv4 {
          value "1";
          description
            "The ICMPv4 Protocol as defined in RFC 792";
        }
        enum icmpv6 {
          value "2";
          description
            "The ICMPv6 Protocol as defined in RFC 4443";
        }
      }
      description
        "The ICMP version to be matched. This value
         affected the type and code values.";
      reference
```

```
          "RFC  792: Internet Control Message Protocol
           RFC 4443: Internet Control Message Protocol
                     (ICMPv6) for the Internet Protocol
                     Version 6 (IPv6) Specification";
      }

      uses packet-fields:acl-icmp-header-fields;
    }
  }
  description
    "Choice of TCP, UDP, SCTP, DCCP, and ICMP as a layer-4
     protocol.";
}

container url-category {
  description
    "Condition for url category";
  leaf description {
    type string;
    description
      "This is description for the condition of a URL's
       category such as SNS sites, game sites, ecommerce
       sites, company sites, and university sites.";
  }

  leaf-list pre-defined {
    type string;
    description
      "This is pre-defined-category. To specify the name of
       URL database.";
  }
  leaf-list user-defined {
    type string;
    description
      "This user-defined-category. To allow a user's manual
       addition of URLs for URL filtering.";
    reference
      "RFC 3986: Uniform Resource Identifier (URI): Generic
       Syntax";
  }
}

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.";
    reference
      "RFC 3261: SIP: Session Initiation Protocol";

    leaf description {
      type string;
      description
       "This is description for voice condition.";
    }

    leaf-list source-voice-id {
      type string;
      description
        "The security policy rule according to
         a source voice ID for VoIP and VoCN.";
    }

    leaf-list destination-voice-id {
      type string;
      description
        "The security policy rule according to
         a destination voice ID for VoIP and VoCN.";
    }

    leaf-list user-agent {
      type string;
      description
        "The security policy rule according to
         a user agent for VoIP and VoCN.";
    }
  }

  container ddos {
    description
      "Condition for DDoS attack.";

    leaf description {
      type string;
      description
       "This is description for ddos condition.";
    }

    leaf alert-packet-rate {
      type uint32;
      units "pps";
```

```
    description
      "The alert rate of flood detection for
       packets per second (PPS) of an IP address.
       If the PPS of an IP address exceeds
       the alert rate threshold, an alert
       will be generated.";
  }

  leaf alert-flow-rate {
    type uint32;
    description
      "The alert rate of flood detection for the
       flow creating requests (e.g., new TCP connection
       establishment) per second of an IP address as
       either a source node or a destination node. If
       the flows per second of an IP address exceeds
       the alert rate threshold, an alert will be
       generated.";
  }

  leaf alert-byte-rate {
    type uint32;
    units "Bps";
    description
      "The alert rate of flood detection for
       bytes per second (Bps) of an IP address.
       If the bytes per second of an IP address
       exceeds the alert rate threshold, an alert
       will be generated.";
  }
}

container anti-virus {
  description
    "Condition for antivirus";

  leaf-list profile {
    type string;
    description
      "The security profile for antivirus. This is used to
       update the security profile for improving the
       security. The security profile is used to scan
       the viruses.";
  }

  leaf-list exception-files {
    type string;
    description
      "The type or name of the files to be excluded by the
```

```
        antivirus. 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: Linux Programmer's Manual - GLOB";
    }
  }

  container payload {
    description
      "Condition for packet payload";
    leaf description {
      type string;
      description
       "This is description for payload condition.";
    }
    leaf-list content {
      type binary;
      description
        "This is a condition for packet payload content.
         The payload content is the binary stream contained
         by a security attack such as backdoor attack. It is
         usually used for Deep Packet Inspection (DPI).";
    }
  }

  container context {
    description
      "Condition for context";
    leaf description {
      type string;
      description
        "This is description for context condition.";
    }

    container time {
      description
        "Time to determine when the policy 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 policy rule.
     The policy rule will stop working after the
     specified end-date-time.";
}

container period {
  when
    "../frequency!='only-once'";
  description
    "This represents the repetition time.  In the case
     where the frequency is weekly, the days can be
     set.";
  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 date of every month.
       More than one date can be specified.";
  }
  leaf-list month {
    when
      "../../frequency='yearly'";
    type string{
      pattern '\d{2}-\d{2}';
    }
```

```
      min-elements 1;
      description
        "This represents the repeated date and month of
         every year.  More than one can be specified.
         A pattern used here is Month and Date (MM-DD).";
    }
  }

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

```
container application {
  description
    "Condition for application";
  leaf description {
    type string;
    description
     "This is description for application condition.";
  }
  leaf-list protocol {
    type identityref {
      base application-protocol;
    }
    description
      "The condition based on the application layer
       protocol";
  }
}

container device-type {
  description
    "Condition for type of the destination device";
  leaf description {
    type string;
    description
      "This is description for destination device type
       condition. Vendors can write instructions for the
       condition that vendor made";
  }

  leaf-list device {
    type identityref {
      base device-type;
    }
    description
      "The device attribute that can identify a device,
       including the device type (i.e., router, switch,
       pc, ios, or android) and the device's owner as
       well.";
  }
}

container users {
  description
    "Condition for users";
  leaf description {
    type string;
    description
      "This is the description for users' condition.";
```

```
  }
  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
    "The location which network traffic flow is associated
     with. The region can be the geographic location such
     as country, province, and city, as well as the logical
     network location such as IP address, network section,
     and network domain.";
  reference
```

```
        "RFC 9179: A YANG Grouping for Geographic Locations";

      leaf description {
        type string;
        description
          "This is the description for the geographic location
           condition. It is used to describe the conditions and
           instructions that should be implemented.";
      }

      leaf-list source {
        type string;
        description
          "The source is a geographic location mapped into an
           IP address. It matches the mapped IP address to the
           source IP address of the traffic flow.";
        reference
          "ISO 3166: Codes for the representation of
           names of countries and their subdivisions
           RFC 9179: A YANG Grouping for Geographic Locations";
      }

      leaf-list destination {
        type string;
        description
          "The destination is a geographic location mapped into
           an IP address. It matches the mapped IP address to
           the destination IP address of the traffic flow.";
        reference
          "ISO 3166: Codes for the representation of
           names of countries and their subdivisions
           RFC 9179: A YANG Grouping for Geographic Locations";
      }
    }
  }
}

container action {
  description
    "An action is used to control and monitor aspects of
     flow-based NSFs when the event and condition clauses
     are satisfied. NSFs provide security functions by
     executing various Actions. Examples of I2NSF Actions
     include providing intrusion detection and/or protection,
     web and flow filtering, and deep packet inspection
     for packets and flows.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - I2NSF Flow Security Policy Structure
```

```
   draft-ietf-i2nsf-capability-data-model-32:
   I2NSF Capability YANG Data Model - Design Principles and
   ECA Policy Model Overview";

leaf description {
  type string;
  description
    "Description for an action clause.";
}

container packet-action {
  description
    "Action for packets";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - I2NSF Flow Security Policy Structure
     draft-ietf-i2nsf-capability-data-model-32:
     I2NSF Capability YANG Data Model - Design Principles and
     ECA Policy Model Overview";

  leaf ingress-action {
    type identityref {
      base ingress-action;
    }
    description
      "Ingress Action: pass, drop, reject, rate-limit, and
       mirror.";
  }

  leaf egress-action {
    type identityref {
      base egress-action;
    }
    description
      "Egress action: pass, drop, reject, rate-limit, mirror,
       invoke-signaling, tunnel-encapsulation, forwarding,
       redirection, and transformation.";
  }

  leaf log-action {
    type identityref {
      base log-action;
    }
    description
      "Log action: rule log and session log";
  }

}
```

```
container flow-action {
  description
    "Action for flows";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - I2NSF Flow Security Policy Structure
     draft-ietf-i2nsf-capability-data-model-32:
     I2NSF Capability YANG Data Model - Design Principles and
     ECA Policy Model Overview";

  leaf ingress-action {
    type identityref {
      base ingress-action;
    }
    description
      "Action: pass, drop, reject, rate-limit, and mirror.";
  }

  leaf egress-action {
    type identityref {
      base egress-action;
    }
    description
      "Egress action: pass, drop, reject, rate-limit, mirror,
       invoke-signaling, tunnel-encapsulation, forwarding,
       redirection, and transformation.";
  }

  leaf log-action {
    type identityref {
      base log-action;
    }
    description
      "Log action: rule log and session log";
  }
}

container advanced-action {
  description
    "If the packet needs to be additionally inspected,
     the packet is passed to advanced network
     security functions according to the profile.
     The profile means the types of NSFs where the packet
     will be forwarded in order to additionally
     inspect the packet.
     The advanced action activates Service Function
     Chaining (SFC) for further inspection of a packet.";
  reference
    "draft-ietf-i2nsf-capability-data-model-32:
```

```
          I2NSF Capability YANG Data Model - YANG Tree
          Diagram";

      leaf-list content-security-control {
        type identityref {
          base content-security-control;
        }
        description
          "Content-security-control is the NSFs that
           inspect the payload of the packet.
           The profile for the types of NSFs for mitigation is
           divided into content security control and
           attack-mitigation-control.
           Content security control: ips, url filtering,
           antivirus, and voip-vocn-filter. This can be
           extended according to the provided NSFs.";
        reference
          "draft-ietf-i2nsf-capability-data-model-32:
           I2NSF Capability YANG Data Model - YANG Tree Diagram";
      }

      leaf-list attack-mitigation-control {
        type identityref {
          base attack-mitigation-control;
        }
        description
          "Attack-mitigation-control is the NSFs that weaken
           the attacks related to a denial-of-service (DoS)
           and reconnaissance.
           The profile for the types of NSFs for mitigation is
           divided into content security control and
           attack-mitigation-control.
           Attack mitigation control: Anti-DDoS or DDoS
           mitigator. This can be extended according to the
           provided NSFs such as mitigators for ip sweep,
           port scanning, ping of death, teardrop, oversized
           icmp, and tracert.";
        reference
          "draft-ietf-i2nsf-capability-data-model-32:
           I2NSF Capability YANG Data Model - YANG Tree Diagram";
      }
    }
  }
}
container rule-group {
  description
    "This is rule group";

  list groups {
```

```
        key "group-name";
        description
          "This is a group for rules";

        leaf group-name {
          type string;
          description
            "This is the name of the group for rules";
        }

        leaf-list rule-name {
          type leafref {
            path
              "../../../rules/name";
          }
          description
            "The names of the rules to be grouped.";
        }

        leaf enable {
          type boolean;
          description
            "If true, the rule is enabled and enforced.
             If false, the rule is configured but disabled
             and not enforced.";
        }

        leaf description {
          type string;
          description
            "This is a description for rule-group";
        }
      }
    }
  }
}

<CODE ENDS>
```

Figure 5: YANG Data Module of I2NSF NSF-Facing-Interface

5. XML Configuration Examples of Low-Level Security Policy Rules

This section shows XML configuration examples of low-level security policy rules that are delivered from the Security Controller to NSFs over the NSF-Facing Interface. For security requirements, we assume that the NSFs (i.e., General firewall, Time-based firewall, URL filter, VoIP/VoCN filter, and HTTP and HTTPS flood mitigation) described in Appendix A of [I-D.ietf-i2nsf-capability-data-model] are registered with the I2NSF framework. With the registered NSFs,

we show configuration examples for security policy rules of network security functions according to the following three security requirements: (i) Block Social Networking Service (SNS) access during business hours, (ii) Block malicious VoIP/VoCN packets coming to the company, and (iii) Mitigate HTTP and HTTPS flood attacks on company web server.

5.1. Example Security Requirement 1: Block Social Networking Service (SNS) Access during Business Hours

This section shows a configuration example for blocking SNS access during business hours in IPv4 networks or IPv6 networks.

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
 <name>sns_access</name>
 <rules>
  <name>block_sns_access_during_operation_time_for_ipv4</name>
  <condition>
   <ipv4>
    <source-ipv4-network>192.0.2.0/24</source-ipv4-network>
   </ipv4>
   <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>
   <advanced-action>
    <content-security-control>
      url-filtering
    </content-security-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

Figure 6: Configuration XML for Time-based Firewall to Block SNS Access during Business Hours in IPv4 Networks

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
 <name>sns_access</name>
 <rules>
  <name>block_sns_access_during_operation_time_for_ipv6</name>
  <condition>
   <ipv6>
    <source-ipv6-network>2001:db8:1::/60</source-ipv6-network>
   </ipv6>
   <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>
   <advanced-action>
    <content-security-control>
      url-filtering
    </content-security-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

Figure 7: Configuration XML for Time-based Firewall to Block SNS Access during Business Hours in IPv6 Networks

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
 <name>sns_access</name>
 <rules>
  <name>block_sns_access_during_operation_time</name>
  <condition>
   <url-category>
    <user-defined>SNS_1</user-defined>
    <user-defined>SNS_2</user-defined>
   </url-category>
  </condition>
  <action>
   <packet-action>
    <egress-action>drop</egress-action>
   </packet-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

Figure 8: Configuration XML for Web Filter to Block SNS Access during Business Hours

Figure 6 and Figure 7 show the configuration XML documents for a time-based firewall for IPv4 and IPv6, respectively. Figure 8 shows the configuration XML document for a web filter. The two NSFs combined to block SNS access during business hours in IPv4 networks (or IPv6 networks). For the security requirement, two NSFs (i.e., a time-based firewall and a web filter) were used because one NSF cannot meet the security requirement. The instances of XML documents for the time-based firewall and the web filter are as follows: Note that a detailed data model for the configuration of the advanced network security function (i.e., web filter) can be defined as an extension in future.

Time-based Firewall is as follows:

1. The name of the security policy is sns_access.

2. The name of the rule is block_sns_access_during_operation_time_for_ipv4 and block_sns_access_during_operation_time_for_ipv6.

3. The rule is started from 2021-03-11 at 9 a.m. to 2021-12-31 at 6 p.m.

4. The rule is operated weekly every weekday (i.e., Monday, Tuesday, Wednesday, Thursday, and Friday) during the business hours (i.e., from 9 a.m. to 6 p.m.).

5. The rule inspects a source IPv4 address (i.e., 192.0.2.0/24). For the case of IPv6 networks, the rule inspects a source IPv6 address (i.e., from 2001:db8:1::/60).

6. If the outgoing packets match the rules above, the time-based firewall sends the packets to url filtering for additional inspection because the time-based firewall can not inspect contents of the packets for the SNS URL.

Web Filter is as follows:

1. The name of the security policy is sns_access.

2. The name of the rule is block_SNS_1_and_SNS_2.

3. The rule inspects URL address to block the access packets to the SNS_1 or the SNS_2.

4. If the outgoing packets match the rules above, the packets are blocked.

5.2. Example Security Requirement 2: Block Malicious VoIP/VoCN Packets Coming to a Company

This section shows a configuration example for blocking malicious VoIP/VoCN packets coming to a company.

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
 <name>voip_vocn_inspection</name>
 <rules>
  <name>block_malicious_voice_id</name>
  <condition>
   <ipv4>
    <destination-ipv4-network>192.0.2.0/24</destination-ipv4-network>
   </ipv4>
   <tcp>
    <destination-port-number>
     <lower-port>5060</lower-port>
     <upper-port>5061</upper-port>
    </destination-port-number>
   </tcp>
  </condition>
  <action>
   <advanced-action>
    <content-security-control>
      voip-vocn-filtering
    </content-security-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

Figure 9: Configuration XML for General Firewall to Block Malicious VoIP/VoCN Packets Coming to a Company

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
 <name>voip_vocn_inspection</name>
 <rules>
  <name>block_malicious_voice_id</name>
  <condition>
   <voice>
    <source-voice-id>
      user1@voip.malicious.example.com
    </source-voice-id>
    <source-voice-id>
      user2@voip.malicious.example.com
    </source-voice-id>
   </voice>
  </condition>
  <action>
   <flow-action>
    <ingress-action>drop</ingress-action>
   </flow-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

Figure 10: Configuration XML for VoIP/VoCN Filter to Block Malicious VoIP/VoCN Packets Coming to a Company

Figure 9 and Figure 10 show the configuration XML documents for general firewall and VoIP/VoCN filter to block malicious VoIP/VoCN packets coming to a company. For the security requirement, two NSFs (i.e., a general firewall and a VoIP/VoCN filter) were used because one NSF can not meet the security requirement. The instances of XML documents for the general firewall and the VoIP/VoCN filter are as follows: Note that a detailed data model for the configuration of the advanced network security function (i.e., VoIP/VoCN filter) can be described as an extension in future.

General Firewall is as follows:

1. The name of the security policy is voip_vocn_inspection.

2. The name of the rule is block_malicious_voice_id.

3. The rule inspects a destination IPv4 address (i.e., from 192.0.2.0/24).

4. The rule inspects a port number (i.e., 5060 and 5061) to inspect VoIP/VoCN packet.

5. If the incoming packets match the rules above, the general firewall sends the packets to VoIP/VoCN filter for additional

inspection because the general firewall can not inspect contents of the VoIP/VoCN packets.

VoIP/VoCN Filter is as follows:

1. The name of the security policy is malicious_voice_id.

2. The name of the rule is block_malicious_voice_id.

3. The rule inspects the voice ID of the VoIP/VoCN packets to block the malicious VoIP/VoCN packets (i.e., user1@voip.malicious.example.com and user2@voip.malicious.example.com).

4. If the incoming packets match the rules above, the packets are blocked.

5.3. Example Security Requirement 3: Mitigate HTTP and HTTPS Flood Attacks on a Company Web Server

This section shows a configuration example for mitigating HTTP and HTTPS flood attacks on a company web server.

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
 <name>flood_attack_mitigation</name>
 <rules>
  <name>mitigate_http_and_https_flood_attack</name>
  <condition>
   <ipv4>
    <destination-ipv4-network>192.0.2.0/24</destination-ipv4-network>
   </ipv4>
   <tcp>
    <destination-port-number>
     <port-numbers>
      <start>80</start>
      <end>80</end>
     </port-numbers>
     <port-numbers>
      <start>443</start>
      <end>443</end>
     </port-numbers>
    </destination-port-number>
   </tcp>
  </condition>
  <action>
   <advanced-action>
    <attack-mitigation-control>
      anti-ddos
    </attack-mitigation-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

Figure 11: Configuration XML for General Firewall to Mitigate HTTP and HTTPS Flood Attacks on a Company Web Server

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
 <name>flood_attack_mitigation</name>
 <rules>
  <name>mitigate_http_and_https_flood_attack</name>
  <condition>
   <ddos>
    <alert-packet-rate>1000</alert-packet-rate>
   </ddos>
  </condition>
  <action>
   <flow-action>
    <ingress-action>drop</ingress-action>
   </flow-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

Figure 12: Configuration XML for Anti-DDoS to Mitigate HTTP and HTTPS Flood Attacks on a Company Web Server

Figure 11 and Figure 12 show the configuration XML documents for general firewall and HTTP and HTTPS flood attack mitigation to mitigate HTTP and HTTPS flood attacks on a company web server. For the security requirement, two NSFs (i.e., a general firewall and a HTTP and HTTPS flood attack mitigation) were used because one NSF can not meet the security requirement. The instances of XML documents for the general firewall and HTTP and HTTPS flood attack mitigation are as follows: Note that a detailed data model for the configuration of the advanced network security function (i.e., HTTP and HTTPS flood attack mitigation) can be defined as an extension in future.

General Firewall is as follows:

1. The name of the security policy is flood_attack_mitigation.

2. The name of the rule is mitigate_http_and_https_flood_attack.

3. The rule inspects a destination IPv4 address (i.e., 192.0.2.0/24) to inspect the access packets coming into the company web server.

4. The rule inspects a port number (i.e., 80 and 443) to inspect HTTP and HTTPS packet.

5. If the packets match the rules above, the general firewall sends the packets to anti-DDoS for additional inspection because the general firewall can not control the amount of packets for HTTP and HTTPS packets.

Anti DDoS for HTTP and HTTPS Flood Attack Mitigation is as follows:

1. The name of the security policy is flood_attack_mitigation.

2. The name of the rule is mitigate_http_and_https_flood_attack.

3. The rule controls the HTTTP and HTTPS packets according to the amount of incoming packets (1000 packets per second).

4. If the incoming packets match the rules above, the packets are blocked.

6. 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-nsf-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-nsf-facing-interface
namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface
prefix:    i2nsfnfi
reference: RFC XXXX
```

7. 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 NETCONF access control model [RFC8341] provides a means of restricting access to specific NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative

effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

* ietf-i2nsf-nsf-facing-interface: Writing to almost any element of this YANG module would directly impact on the configuration of NSFs, 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 rendering useless trained statistics or artificial intelligence models.

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

* ietf-i2nsf-nsf-facing-interface: The attacker may gather the security policy information of any target NSFs and misuse the security policy information for subsequent attacks.

Policy rules identifying the specified users and user groups can be specified with "rules/condition/context/users". As with other data in this YANG module, this user information is provided by the Security Controller to the NSFs and is protected via the transport and access control mechanisms described above.

8. References

8.1. 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>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[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>.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <https://www.rfc-editor.org/info/rfc2132>.

[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>.

[RFC5595] Fairhurst, G., "The Datagram Congestion Control Protocol (DCCP) Service Codes", RFC 5595, DOI 10.17487/RFC5595, September 2009, <https://www.rfc-editor.org/info/rfc5595>.

[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>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[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>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

[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>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[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>.

[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>.

[RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <https://www.rfc-editor.org/info/rfc8519>.

[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>.

[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>.

[I-D.ietf-httpbis-http2bis] Thomson, M. and C. Benfield, "HTTP/2", Work in Progress, Internet-Draft, draft-ietf-httpbis-http2bis-07, 24 January 2022, <https://www.ietf.org/archive/id/draft-ietf-httpbis-http2bis-07.txt>.

[I-D.ietf-httpbis-messaging] Fielding, R. T., Nottingham, M., and J. Reschke, "HTTP/1.1", Work in Progress, Internet-Draft, draft-ietf-httpbis-messaging-19, 12 September 2021, <https://www.ietf.org/archive/id/draft-ietf-httpbis-messaging-19.txt>.

[I-D.ietf-httpbis-semantics]
Fielding, R. T., Nottingham, M., and J. Reschke, "HTTP Semantics", Work in Progress, Internet-Draft, draft-ietf-httpbis-semantics-19, 12 September 2021, <https://www.ietf.org/archive/id/draft-ietf-httpbis-semantics-19.txt>.

[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://www.ietf.org/archive/id/draft-ietf-i2nsf-capability-data-model-32.txt>.

[I-D.ietf-i2nsf-nsf-monitoring-data-model]
Jeong, J. P., Lingga, P., Hares, S., Xia, L. F., and H. Birkholz, "I2NSF NSF Monitoring Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-nsf-monitoring-data-model-19, 23 May 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-nsf-monitoring-data-model-19.txt>.

[I-D.ietf-tcpm-rfc793bis]
Eddy, W. M., "Transmission Control Protocol (TCP) Specification", Work in Progress, Internet-Draft, draft-ietf-tcpm-rfc793bis-28, 7 March 2022, <https://www.ietf.org/archive/id/draft-ietf-tcpm-rfc793bis-28.txt>.

[I-D.ietf-tsvwg-rfc4960-bis] Stewart, R. R., Tüxen, M., and K. E. E. Nielsen, "Stream Control Transmission Protocol", Work in Progress, Internet-Draft, draft-ietf-tsvwg-rfc4960-bis-19, 5 February 2022, <https://www.ietf.org/archive/id/draft-ietf-tsvwg-rfc4960-bis-19.txt>.

8.2. Informative References

[RFC4732] Handley, M., Ed., Rescorla, E., Ed., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, DOI 10.17487/RFC4732, December 2006, <https://www.rfc-editor.org/info/rfc4732>.

[RFC4987] Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations", RFC 4987, DOI 10.17487/RFC4987, August 2007, <https://www.rfc-editor.org/info/rfc4987>.

[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>.

[RFC9179] Hopps, C., "A YANG Grouping for Geographic Locations", RFC 9179, DOI 10.17487/RFC9179, February 2022, <https://www.rfc-editor.org/info/rfc9179>.

[I-D.ietf-i2nsf-consumer-facing-interface-dm] Jeong, J. P., Chung, C., Ahn, T., Kumar, R., and S. Hares, "I2NSF Consumer-Facing Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-consumer-facing-interface-dm-20, 23 May 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-consumer-facing-interface-dm-20.txt>.

[GLOB] "Linux Programmer's Manual - GLOB", 13 August 2020, <https://man7.org/linux/man-pages/man7/glob.7.html>.

[ISO-3166] "Codes for the representation of names of countries and their subdivisions", ISO 3166, September 2018, <https://www.iso.org/iso-3166-country-codes.html>.

[IEEE-802.3] Institute of Electrical and Electronics Engineers, "IEEE Standard for Ethernet", 2018, <https://ieeexplore.ieee.org/document/8457469/>.

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, Acee Lindem, Dan Romascanu (GenART), Yoshifumi Nishida (TSVART), Kyle Rose (SecDir), Joe Clarke (OpsDir), and Tom Petch. 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, 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 Seobu-ro Jangan-gu, Suwon, Gyeonggi-do 16419, Republic of Korea, EMail: patricklink@skku.edu

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

Daeyoung Hyun - Department of Computer Science and Engineering, Sungkyunkwan University, 2066 Seobu-ro Jangan-gu, Suwon, Gyeonggi-do 16419, Republic of Korea, EMail: dyhyun@skku.edu

Dongjin Hong - Department of Electronic, Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro Jangan-gu, Suwon, Gyeonggi-do 16419, Republic of Korea, EMail: dong.jin@skku.edu

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

Tae-Jin Ahn - Korea Telecom, 70 Yuseong-Ro, Yuseong-Gu, Daejeon, 305-811, Republic of Korea, EMail: taejin.ahn@kt.com

Se-Hui Lee - Korea Telecom, 70 Yuseong-Ro, Yuseong-Gu, Daejeon, 305-811, Republic of Korea, EMail: sehuilee@kt.com

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

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

* This version updated a 'leaf language' pattern by adding extra parentheses around "[A-Za-z]{2,3}(-[A-Za-z]{3}(-[A-Za-z]{3}){0,2})?" and removing a range character '-' between characters 'Y' and 'Z' in "|([0-9][A-Za-z0-9]{3})))*(-[0-9A-WY-Za-wy-z]" as 'Y' is alphabetically adjacent to 'Z'.

Authors' Addresses

Jinyong Tim Kim (editor)
Department of Electronic, Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea

Phone: +82 10 8273 0930
Email: timkim@skku.edu

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

Jung-Soo Park
Electronics and Telecommunications Research Institute
218 Gajeong-Ro, Yuseong-Gu
Daejeon
34129
Republic of Korea

Phone: +82 42 860 6514
Email: pjs@etri.re.kr

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
United States of America

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

Qiushi Lin
Huawei
Huawei Industrial Base
Shenzhen
Guangdong 518129,
China

Email: linqiushi@huawei.com