I2NSF Network Security Function-Facing Interface YANG Data Model
draft-ietf-i2nsf-nsf-facing-interface-dm-29

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.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

This document defines a YANG [RFC6020][RFC7950] data model for security policy rule configuration of Network Security Functions
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, layer 3, 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 Dataplane 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.


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

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]
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 source-mac-address     yang:mac-address
---rw destination-mac-address-mask?     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
   |               |       +--:(ipv6)
   |               |       ---rw ipv6
   |               |           ---rw description?     string
   |               |           ---rw dscp?     inet:dscp


Internet-Draft    NSF-Facing Interface YANG Data Model         June 2022
++-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
    |   ++-(source-network)?
    |     |   +++-(source-ipv6-network)
    |     |     ++--rw source-ipv6-network? inet:ipv6-prefix
    |     |     ++--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)?
    |     |     |   +++-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)?
---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 context
  | ---rw description?     string
  | ---rw content*         binary
  +---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
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.
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].
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


<CODE BEGINS> file "ietf-i2nsf-nsf-facing-interface@2022-06-01.yang"
module ietf-i2nsf-nsf-facing-interface {
    yang-version 1.1;
    prefix i2nsfnfi;

    import ietf-inet-types {
        prefix inet;

        reference "Section 4 of RFC 6991";
    }
    import ietf-yang-types {
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
     Monitoring Interface YANG Data Model - Event";
}

identity system-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
Monitoring Interface YANG Data Model - System event"
}

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
  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
}

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
}

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
}

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
}
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
         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
         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";
}


Internet-Draft    NSF-Facing Interface YANG Data Model         June 2022
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
}

identity fmr {
  base resolution-strategy;
  description
  "Conflict resolution with First Matching Rule (FMR)";
  reference
}

identity lmr {
  base resolution-strategy;
  description
  "Conflict resolution with Last Matching Rule (LMR)";
  reference
}

identity pmre {
  base resolution-strategy;
  description
"Conflict resolution with Prioritized Matching Rule with Errors (PMRE)";
reference
}

identity pmrn {
  base resolution-strategy;
  description
  "Conflict resolution with Prioritized Matching Rule with No Errors (PMRN)";
  reference
}

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
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;
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.";
        }
    }
}
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-z)\{0-9\}\{5,8\}' + '|([0-9]\{A-Za-z\}0-9\{3\}\})*(-A-9A-WYzA-wyz)' + '((-A-Za-z\{0-9\}\{2,8\})+)*(-Xx)(-A-Za-z\{0-9\}' + '{1,8})?Xx)\{0-9\}\{1,8\}\{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]' + '([Dd][Ee]\{Ff\}[Aa]\{Uu]\{Ll]}|Tt][Ii]";

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;
    }
}

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

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

    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";
}
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

  leaf description {
    type string;
    description
      "Description for an event clause";
    }
  leaf-list system-event {
    type identityref {
      base system-event;
    }
  }

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
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.

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.

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.

"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.";
      }
    }
  }
"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.";
}

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
    "The security policy rule according to
tcp source port number.";
reference
    "draft-ietf-tcpm-rfc793bis-25: Transmission Control
    Protocol (TCP) Specification - Port Number";
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 {

        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 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.";
    }
    description
    "The security policy rule according to sctp destination port number.";
  }
  description
  "The security policy rule according to sctp destination port number.";
  reference
  "draft-ietf-tsvwg-rfc4960-bis-18: Stream Control Transmission Protocol - Port number";
}
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.";
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.";
    }
  }
}
The choice of source port definition using range/operator or a choice to use list of port numbers.

The security policy rule according to dccp source port number.

Destination port definition from range or operator. Can be used when a single port range to be specified.

List of destination port numbers. In the case of multiple port ranges needed to be specified.

The choice of destination port definition using range/operator or a choice to use list of port numbers.
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
}

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
affect 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";
  }
  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.

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.";

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";
    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";
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
}

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

Kim, et al. Expires 3 December 2022 [Page 60]

Internet-Draft NSF-Facing Interface YANG Data Model June 2022

"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

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.

```xml
  <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>
        </context>
      </time>
    </condition>
  </rules>
</i2nsf-security-policy>
```
  <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>
          </period>
        </time>
      </context>
    </condition>
  </rules>
</i2nsf-security-policy>

Figure 6: Configuration XML for Time-based Firewall to Block SNS Access during Business Hours in IPv4 Networks
Figure 7: Configuration XML for Time-based Firewall to Block SNS Access during Business Hours in IPv6 Networks
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
  <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:i2nf-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
  <name>flood_attack_mitigation</name>
  <rules>
    <name>mitigate_http_andHttps_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>
          <start>443</start>
          <end>443</end>
        </destination-port-number>
      </tcp>
    </condition>
    <action>
      <advanced-action>
        <attack-mitigation-control>
Figure 11: Configuration XML for General Firewall to Mitigate HTTP and HTTPS Flood Attacks on a Company Web Server

```xml
<i2nsf-security-policy
  <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 cannot 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 cannot 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 HTTP 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]:

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
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


[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]
8.2. Informative References


their subdivisions", ISO 3166, September 2018, 

[IEEE-802.3] Institute of Electrical and Electronics Engineers, "IEEE Standard for Ethernet", 2018, 

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


Internet-Draft NSF-Facing Interface YANG Data Model June 2022

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

Daeyeong Hyun - Department of Computer Science and Engineering,
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 "|\((A-Za-z0-9\{3\})\)*\(-[0-9A-WY-Za-wy-z]\)" 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