Operations and Management Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: 17 August 2025

P. Lingga, Ed.
J. Jeong, Ed.
Sungkyunkwan University
Y. Choi
ETRI

13 February 2025

I2NSF Analytics Interface YANG Data Model for Closed-Loop Security Control in the I2NSF Framework
draft-lingga-opsawg-analytics-interface-dm-01

Abstract

This document describes an information model and a YANG data model for the Analytics Interface between an Interface to Network Security Functions (I2NSF) Analyzer and a Security Controller in an I2NSF framework. I2NSF Analyzer collects the monitoring data from Network Security Functions (NSF), and analyzes them with Machine Learning (ML) algorithms. This Analytics Interface is used for I2NSF Analyzer to deliver analysis results (e.g., policy reconfiguration and feedback message) to Security Controller for Closed-Loop Security Control in the I2NSF Framework in [I-D.jeong-i2nsf-security-management-automation]. The YANG data model described in this document is based on the YANG data models of the I2NSF NSF-Facing Interface [I-D.ietf-i2nsf-nsf-facing-interface-dm] and the I2NSF Monitoring Interface [I-D.ietf-i2nsf-nsf-monitoring-data-model].

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 17 August 2025.

Table of Contents

1. Introduction

In a framework for Interface to Network Security Functions (I2NSF) [RFC8329], the Monitoring Interface [I-D.ietf-i2nsf-nsf-monitoring-data-model] is defined as an interface to collect monitoring data (e.g., network statistics and resources) from Network Security Functions (NSF). This data can be received by either a query or a report. In a query-based approach, the data is obtained by a request from a client (e.g., I2NSF Analyzer). But in a report-based approach, the data is provided to I2NSF Analyzer by a server (e.g., NSF) when either a notification or an alarm is triggered by an event. In this model, the report-based approach is

used in the I2NSF framework for realizing the Security Management Automation (SMA) for cloud-based security services [I-D.jeong-i2nsf-security-management-automation]. Thus, monitoring data is sent automatically by NSFs to an I2NSF Analyzer. Figure 1 shows the I2NSF Framework for Security Management Automation.

In the I2NSF Framework, Security Controller, NSFs, and I2NSF Analyzer can construct Closed-Loop Security Control to adjust the security policy system with NSF monitoring data, data analytics, feedback, and policy reconfiguration through the Analytics Interface in the document. The security policy enforcement of a user's perspective can also be done by a Security Policy Translator (SPT) [SPT]. It translates a high-level security policy into the corresponding low-level security policy for the security policy enforcement. Along with this SPT, Intent-Based Closed-Loop Secutity Control (ICSC) [ICSC], which supports Closed-Loop Security Control according to a user's intent, can be designed and implemented in the I2NSF Framework.

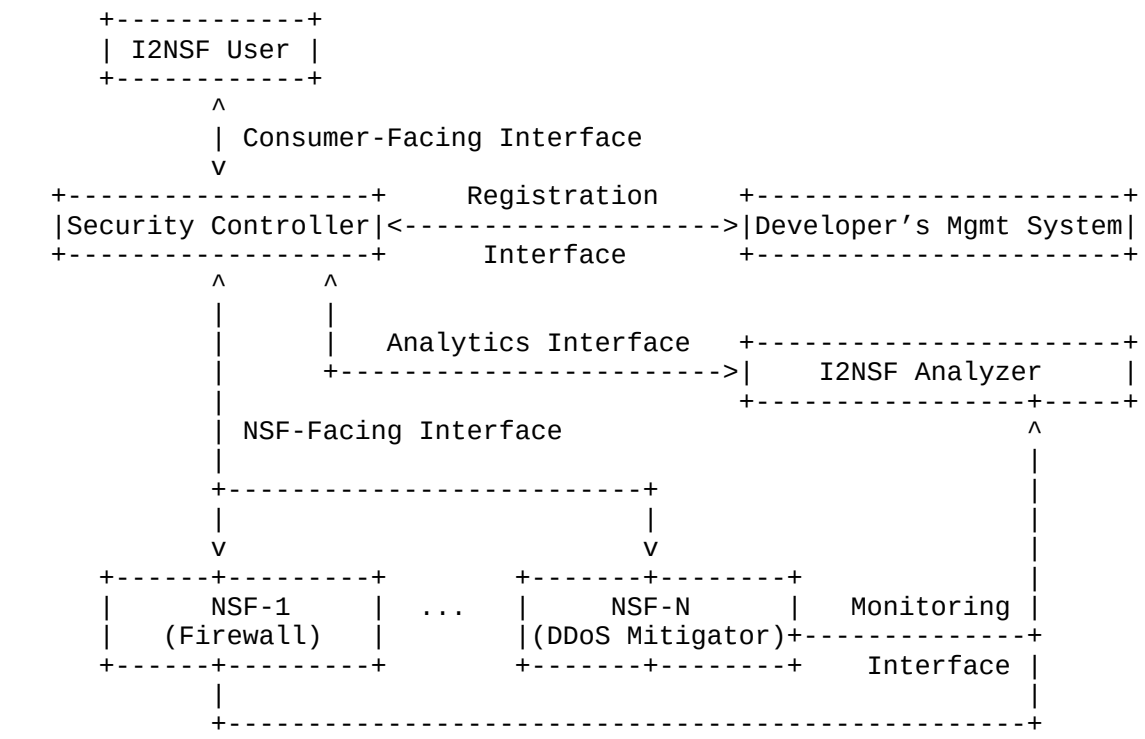

Figure 1: I2NSF Framework for Security Management Automation

The automatic reports of monitoring data by the NSFs are collected in a single instance (i.e., I2NSF Analyzer) to analyze them. By analyzing the monitoring data, a new security policy can be produced to further enhance the security of the network. To create the automated system, the analysis of the monitoring data should be performed automatically with the help of Machine Learning (ML) algorithms. The automated analysis is out of the scope of this document.

A new security policy needs to be delivered from the I2NSF Analyzer to the Security Controller so the new policy can be listed and monitored properly. For that purpose, this document introduces the Analytics Interface as an intermediary interface between the I2NSF Analyzer and the Security Controller. Then the new policy should be delivered directly to appropriate NSFs by the Security Controller via the NSF-Facing Interface [I-D.ietf-i2nsf-nsf-facing-interface-dm].

Therefore, the purpose of this document is to provide an Analytics Interface to a Security Controller in an I2NSF Framework. With the provided Analytics Interface, the realization of Security Management Automation (SMA) is possible through Closed-Loop Security Control in the I2NSF framework. This SMA can facilitate Intent-Based Security Management with Intent-Based Networking (IBN) in [RFC9315].

Note that the scope of this document is to propose a YANG data model for a new external interface called Analytics Interface between Security Controller in the I2NSF framework and an I2NSF Analyzer. The I2NSF Analyzer performs the analysis of NSF monitoring data and the generation of analysis results (e.g., policy reconfiguration and feedback message). With this Analytics Interface, the I2NSF framework can perform Security Management Automation (SMC) in term of Closed-Loop Security Control, which is specified in [I-D.jeong-i2nsf-security-management-automation].

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] and adopts the Network Management Datastore Architecture (NMDA). The meaning of the symbols in tree diagrams is defined in [RFC8340].

3. Information Model for Analytics Interface

This document introduces an Analytics Interface as an interface to deliver an analytics report for augmentation or generation of a security policy rule created by I2NSF Analyzer to Security Controller [I-D.jeong-i2nsf-security-management-automation]. This allows Security Controller to actively reinforce a target network with its security policy management. Figure 2 shows the high-level concept of Analytics Interface to deliver Analytics Information (i.e., Policy Reconfiguration and Feedback Information) to Security Controller.

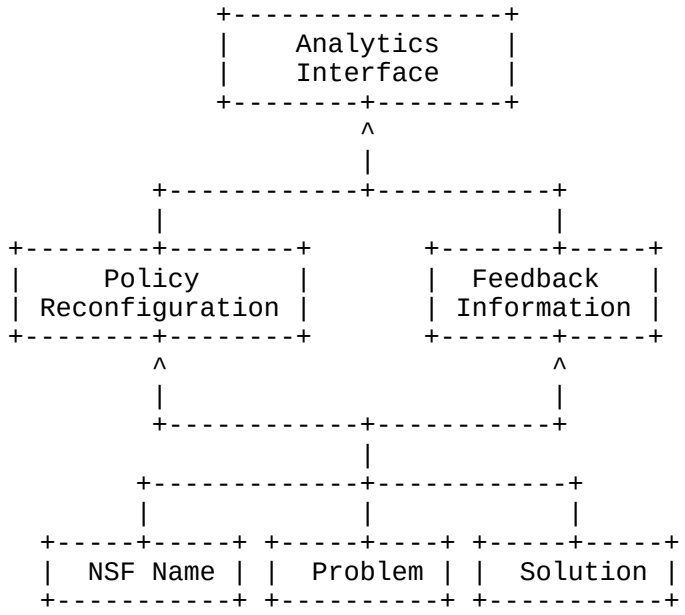

Figure 2: Diagram for Analytics Interface

Both policy reconfiguration and feedback information provide the following high-level abstraction:

* NSF Name: It is the name or IP address of the NSF for identifying the NSF with problem. The name is a unique string to identify an NSF, including a Fully Qualified Domain Name (FQDN).

* Problem: It describes the issue(s) in the NSF that needs to be handled.

* Solution: It specifies the possible solution(s) for the problem.

3.1. Information Model for Policy Reconfiguration

Policy reconfiguration is the rearrangement of a security policy in a different form or combination of the existing security policy to enhance the security service in the network. A policy reconfiguration is generated by the I2NSF Analyzer after receiving and analyzing monitoring information of NSF Events from an NSF [I-D.ietf-i2nsf-nsf-monitoring-data-model].

Policy reconfiguration works together with the three I2NSF interfaces defined for the I2NSF Framework, i.e., NSF-Facing Interface [I-D.ietf-i2nsf-nsf-facing-interface-dm], NSF Monitoring Interface [I-D.ietf-i2nsf-nsf-monitoring-data-model], and Analytics Interface, to create a closed-loop securty system for reinforcing the network security. Figure 3 shows an illustration of the closed-loop system for the I2NSF Framework.

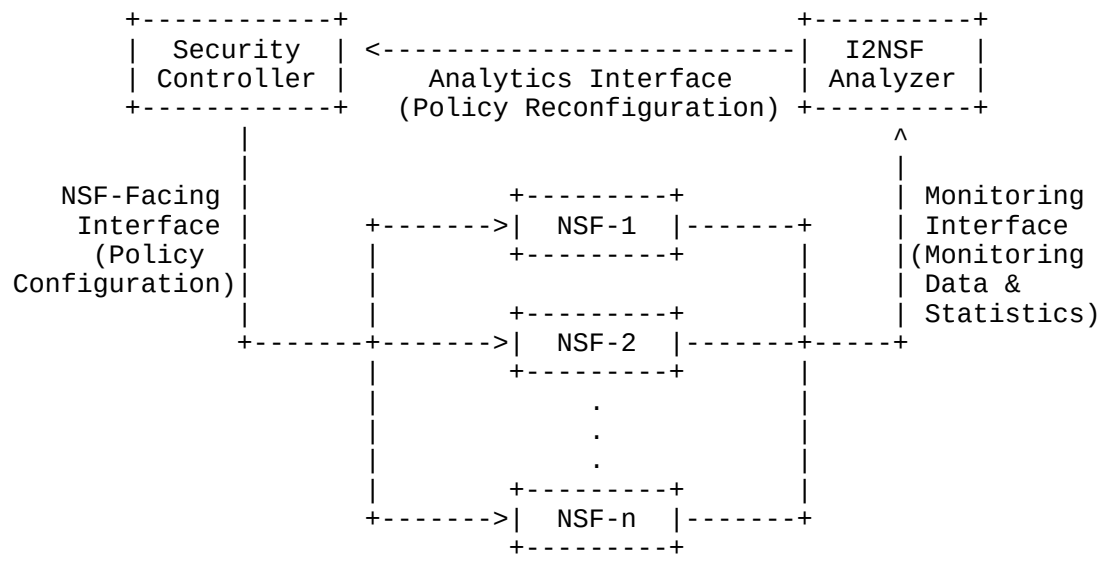

Figure 3: A Closed-Loop Security System for Security Management Automation (SMA)

Figure 3 shows a closed-loop security system between Security Controller, NSF, and I2NSF Analyzer. The Security Controller delivers a security policy to an appropriate NSF via the NSF-Facing Interface [I-D.ietf-i2nsf-nsf-facing-interface-dm]. The NSF will

prepare for a security service according to the given configuration and provide a security service for the network. The NSF SHOULD also provide monitoring data (e.g., NSF Events and System Alarms) to be analyzed. This monitoring data can be delivered by the NSF to an I2NSF Analyzer via the Monitoring Interface [I-D.ietf-i2nsf-nsf-monitoring-data-model]. Then the I2NSF Analyzer analyzes the monitoring data for the reconfiguration of an existing security policy, the generation of a new security policy, and the feedback for security system management (e.g., the scaling-up or scaling-down of resources related to NSFs). To fully automate the closed-loop system, the I2NSF Analyzer should analyze the monitoring data automatically using machine learning techniques (e.g., Deep Learning [Deep-Learning]). The results of the analysis may trigger the reconfiguration of an existing security policy or the generation of a new security policy to strengthen the network security. The reconfiguration or configuration request will be delivered by the I2NSF Analyzer to the Security Controller via the Analytics Interface.

To realize the closed-loop security system, the Analytics Interface needs to properly follow the similar guidelines for the I2NSF Framework [RFC8329]. The Analytics Interface follows [I-D.ietf-i2nsf-nsf-facing-interface-dm] to create a security policy to reconfigure an existing security policy of NSF(s) or to generate a new security policy.

Analytics Interface holds a list of security policies so that the (re)configuration of a security policy and the feedback information can be provided to the Security Controller. Each policy consists of a list of rule(s) to be enhanced on the NSF. Note that the synchronization of the list of security policies should be done between the Security Controller and the I2NSF Analyzer and the specific mechanism is out of the scope of this document. A (re)configured security policy rule should be able to cope with attacks or failures that can happen to the network in near future. Such a rule is reconfigured or generated by the I2NSF Analyzer to tackle a detected problem in the network. It uses the Event-Condition-Action (ECA) model as the basis for the design of I2NSF Policy (Re)configuration as described in [RFC8329] and [I-D.ietf-i2nsf-capability-data-model].

An example of Policy (Re)configuration is a DDoS Attack that is detected by a DDoS Mitigator. The DDoS Mitigator creates monitoring data and delivers it to the I2NSF Analyzer. The I2NSF Analyzer analyzes the monitoring data and generates a new policy to handle the DDoS Attack, such as a firewall rule to drop all packets from the source of the DDoS Attack.

3.2. YANG Tree Structure for Policy Reconfiguration

The YANG tree structure for policy reconfiguration is provided through the augmentation of the NSF-Facing Interface YANG Module [I-D.ietf-i2nsf-nsf-facing-interface-dm] as follows:

```
  augment /i2nsfnfi:i2nsf-security-policy:
    +--rw nsf-name?   union
    +--rw problem
       +--rw (attack-detection)?
          +--:(ddos-detected)
          |  +--rw ddos-detected
          |     +--rw attack-src-ip*     inet:ip-address-no-zone
          |     +--rw attack-dst-ip*     inet:ip-address-no-zone
          |     +--rw attack-src-port*   inet:port-number
          |     +--rw attack-dst-port*   inet:port-number
          +--:(virus-detected)
          |  +--rw virus-detected
          |     +--rw virus-name?   string
          |     +--rw virus-type?   identityref
          |     +--rw host?         union
          |     +--rw file-type?    string
          |     +--rw file-name?    string
          |     +--rw os?           string
          +--:(intrusion-detected)
          |  +--rw intrusion-detected
          |     +--rw protocol?      identityref
          |     +--rw app?           identityref
          |     +--rw attack-type?   identityref
          +--:(web-attack-detected)
          |  +--rw web-attack-detected
          |     +--rw attack-type?      identityref
          |     +--rw req-method?       identityref
          |     +--rw req-uri?          string
          |     +--rw req-user-agent?   string
          |     +--rw cookies?          string
          |     +--rw req-host?         string
          |     +--rw response-code?    string
          +--:(voip-vocn-detected)
             +--rw voip-vocn-detected
                +--rw source-voice-id*        string
                +--rw destination-voice-id*   string
                +--rw user-agent*             string
```

Figure 4: YANG Tree Structure of Policy Reconfiguration

The policy reconfiguration must include the following information:

NSF Name: The name or IP address (IPv4 or IPv6) of the NSF to be configured. If the given nsf-name is not IP address, the name can be an arbitrary string including a Fully Qualified Domain Name (FQDN).

Problem: The issue that is emitted by an NSF via the I2NSF Monitoring Interface. The problem for policy configuration includes the NSF Events described in NSF Monitoring Interface YANG Data Model [I-D.ietf-i2nsf-nsf-monitoring-data-model], such as DDoS detection, Virus detection, Intrusion detection, Web-attack detection, and Voice over Internet Protocol (VoIP) or Voice over Cellular Network (VoCN) violation detection.

Solution: The solution for policy (re)configuration is the security policy that is reconfigured or generated to solve a detected attack. The security policy can be configured using the NSF-Facing Interface YANG data model [I-D.ietf-i2nsf-nsf-facing-interface-dm].

3.3. Information Model for Feedback Information

Feedback information is information about problem(s) of an NSF for a security service such as either over-usage or malfunction of a system resource. This problem cannot be handled by creating a new policy. In the similar way with the policy reconfiguration in Section 3.1, the feedback information should be delivered by the I2NSF Analyzer to the Security Controller that will be able to handle the reported problem(s).

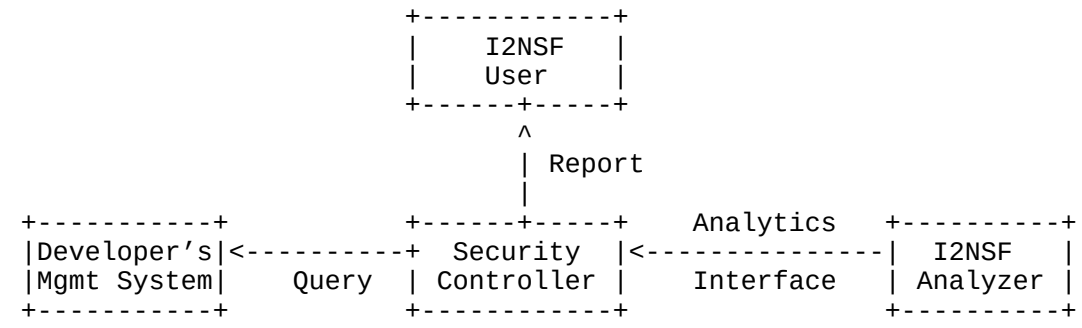

Figure 5: Handling of Feedback Information

Figure 5 shows the handling of feedback information. For the feedback information, the given feedback is not a security policy, hence the Security Controller needs to take an action to handle the reported problem(s). The action includes the report to the I2NSF

User and the query of the system resource management of the relevant NSF(s) to the Developer’s Management System (DMS). DMS will communicate with the Management and Orchestration (MANO) Unit in the Network Functions Virtualization (NFV) Framework to deal with the system management issue(s) of the relevant NSFs [I-D.ietf-i2nsf-applicability]. The details of the handling process are out of the scope of this document.

3.4. YANG Tree Structure for Feedback Information

The YANG tree structure for feedback information is provided with the use of the NSF Monitoring Interface YANG Module [I-D.ietf-i2nsf-nsf-monitoring-data-model] as follows:

```
module: ietf-i2nsf-analytics-interface
  +--rw i2nsf-feedback-information* [nsf-name time]
     +--rw nsf-name    union
     +--rw time        yang:date-and-time
     +--rw language?   string
     +--rw problem
     |  +--rw (alarm-type)?
     |     +--:(memory-alarm)
     |     |  +--rw memory-alarm
     |     |     +--rw usage?      uint8
     |     |     +--rw message?    string
     |     |     +--rw duration?   uint32
     |     +--:(cpu-alarm)
     |     |  +--rw cpu-alarm
     |     |     +--rw usage?      uint8
     |     |     +--rw message?    string
     |     |     +--rw duration?   uint32
     |     +--:(disk-alarm)
     |     |  +--rw disk-alarm
     |     |     +--rw disk-id?    string
     |     |     +--rw usage?      uint8
     |     |     +--rw message?    string
     |     |     +--rw duration?   uint32
     |     +--:(hardware-alarm)
     |     |  +--rw hardware-alarm
     |     |     +--rw component-name?   string
     |     |     +--rw message?          string
     |     |     +--rw duration?         uint32
     |     +--:(interface-alarm)
     |        +--rw interface-alarm
     |           +--rw interface-id?      string
     |           +--rw interface-state?   enumeration
     |           +--rw message?           string
     |           +--rw duration?          uint32
     +--rw solution*   string
```

Figure 6: YANG Tree Structure of Feedback Information

Figure 6 shows the high-level abstraction of Feedback Information. The feedback information should include:

* NSF Name: The name or IP address (IPv4 or IPv6) of the NSF that detected the problem. If the given nsf-name is not IP address, the name can be an arbitrary string including an FQDN.

* Time: The time of the delivery of the feedback information.

* Language: The language tag that is used for the natural language text that is included in the "message" and "solution" attributes. The language field is encoded following the rules in Section 2.1 of [RFC5646]. The default language tag is "en-US".

* Problem: The issue that is emitted by an NSF via the I2NSF Monitoring Interface. The problem for feedback information includes system alarms described in NSF Monitoring Interface YANG Data Model [I-D.ietf-i2nsf-nsf-monitoring-data-model], such as Memory alarm, CPU alarm, Disk alarm, Hardware alarm, and Interface alarm.

* Solution: A possible solution given as feedback is in the form of a free-form string (e.g., a high-level instruction). This value can be interpreted using a Natural Language Processing (NLP) or manually processed by a network operator.

4. YANG Data Model of Analytics Interface

This section shows the YANG module of Analytics Interface. The YANG module in this document is referencing to [RFC6991] [I-D.ietf-i2nsf-nsf-facing-interface-dm] [I-D.ietf-i2nsf-nsf-monitoring-data-model].

The YANG module makes references to [RFC5646] [RFC6265] [RFC8343] [RFC9110]

<CODE BEGINS> file "ietf-i2nsf-analytics-interface@2025-02-13.yang"

```
module ietf-i2nsf-analytics-interface {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-analytics-interface";
  prefix
    i2nsfai;

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

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

 import ietf-i2nsf-nsf-facing-interface {
   prefix i2nsfnfi;
   reference
```

```
        "Section 4 of draft-ietf-i2nsf-nsf-facing-interface-dm-29";
    }

    import ietf-i2nsf-monitoring-interface {
      prefix i2nsfmi;
      reference
        "Section 8 of draft-ietf-i2nsf-nsf-monitoring-data-model-20";
    }

    organization
      "IETF OPSAWG (Operations and Management Area Working Group)";

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

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

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

    description
      "This module is a YANG module for Analytics 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) 2025 IETF Trust and the persons identified as
       authors of the code.  All rights reserved.

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

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

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

```
    revision "2025-02-13" {
      description "Initial revision.";
      reference
        "RFC XXXX: I2NSF Analytics Interface YANG Data Model";

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


    augment "/i2nsfnfi:i2nsf-security-policy" {
      description
      "Augment the NSF-Facing Interface Data Model for the policy
       reconfiguration";
      leaf nsf-name {
        type union {
          type string;
          type inet:ip-address;
        }
        description
          "The name or IP address (IPv4 or IPv6) of the NSF to be
           configured. If the given nsf-name is not IP address, the
           name can be an arbitrary string including FQDN (Fully
           Qualified Domain Name).";
      }

      container problem {
        description
         "Problem: The issue that is emitted by an NSF via the
          I2NSF Monitoring Interface such as DDoS detection, Virus
          detection, Intrusion detection, Web-attack detection, and
          VoIP/VoCN violation detection.";
        choice attack-detection {
          description
            "The detected attack type";
          case ddos-detected {
            container ddos-detected {
              leaf-list attack-src-ip {
                type inet:ip-address-no-zone;
                description
                  "The source IPv4 or IPv6 addresses of attack
                   traffic. It can hold multiple IPv4 or IPv6
                   addresses. Note that all IP addresses should not be
                   included, but only limited IP addresses are included
                   to conserve the server resources. The listed
                   attacking IP addresses can be an arbitrary sampling
                   of the 'top talkers', i.e., the attackers that send
                   the highest amount of traffic.";
```

```
            }
            leaf-list attack-dst-ip {
              type inet:ip-address-no-zone;
              description
                "The destination IPv4 or IPv6 addresses of attack
                 traffic. It can hold multiple IPv4 or IPv6
                 addresses.";
            }
            leaf-list attack-src-port {
              type inet:port-number;
              description
                "The transport-layer source ports of the DDoS attack.
                 Note that not all ports will have been seen on all
                 the corresponding source IP addresses.";
            }
            leaf-list attack-dst-port {
              type inet:port-number;
              description
                "The transport-layer destination ports of the DDoS
                 attack. Note that not all ports will have been seen
                 on all the corresponding destination IP addresses.";
            }
            description
              "A container for DDoS Attack";
          }
          description
            "A DDoS Attack is detected";
        }
        case virus-detected {
          container virus-detected {
            leaf virus-name {
              type string;
              description
                "The name of the detected virus";
            }
            leaf virus-type {
              type identityref {
                base i2nsfmi:virus-type;
              }
              description
                "The virus type of the detected virus";
            }
            leaf host {
              type union {
                type string;
                type inet:ip-address-no-zone;
              }
              description
```

```
                "The name or IP address of the host/device. This is
                 used to identify the host/device that is infected by
                 the virus. If the given name is not an IP address,
                 the name can be an arbitrary string including a FQDN
                 (Fully Qualified Domain Name). The name MUST be
                 unique in the scope of management domain for
                 identifying the device that has been infected with
                 a virus.";
            }
            leaf file-type {
              type string;
              description
                "The type of a file (indicated by the file's suffix,
                 e.g., .exe) where virus code is found (if
                 applicable).";
            }
            leaf file-name {
              type string;
              description
                "The name of file virus code is found in (if
                 applicable).";
            }
            leaf os {
              type string;
              description
                "The operating system of the device.";
            }
            description
              "A Virus Attack is detected";
          }
          description
            "A virus is detected";
        }
        case intrusion-detected {
          container intrusion-detected {
            leaf protocol {
              type identityref {
                base i2nsfmi:transport-protocol;
              }
              description
                "The transport protocol type for
                 nsf-detection-intrusion notification";
            }
            leaf app {
              type identityref {
                base i2nsfmi:application-protocol;
              }
              description
```

```
                "The employed application layer protocol";
              }
              leaf attack-type {
                type identityref {
                  base i2nsfmi:intrusion-attack-type;
                }
                description
                  "The sub attack type for intrusion attack";
              }
              description
                "An intrusion is detected";
            }
          }
          case web-attack-detected {
            container web-attack-detected {
              leaf attack-type {
                type identityref {
                  base i2nsfmi:web-attack-type;
                }
                description
                  "Concrete web attack type, e.g., SQL injection,
                   command injection, XSS, and CSRF.";
              }
              leaf req-method {
                type identityref {
                  base i2nsfmi:req-method;
                }
                description
                  "The HTTP request method, e.g., PUT or GET.";
                reference
                  "RFC 9110: HTTP Semantics - Request Methods";
              }
              leaf req-uri {
                type string;
                description
                  "The Requested URI";
              }
              leaf req-user-agent {
                type string;
                description
                  "The request user agent";
              }
              leaf cookies {
                type string;
                description
                  "The HTTP Cookies header field of the request from
                   the user agent. The cookie information needs to be
                   kept confidential and is NOT RECOMMENDED to be
```

```
                    included in the monitoring data unless the
                    information is absolutely necessary to help to
                    enhance the security of the network.";
                  reference
                    "RFC 6265: HTTP State Management Mechanism - Cookie";
                }
                leaf req-host {
                  type string;
                  description
                    "The domain name of the requested host";
                }
                leaf response-code {
                  type string;
                  description
                    "The HTTP Response code";
                  reference
                    "IANA Website: Hypertext Transfer Protocol (HTTP)
                     Status Code Registry";
                }
                description
                  "A web attack is detected";
              }
              description
                "A web attack is detected";
            }
            case voip-vocn-detected {
              container voip-vocn-detected {
                leaf-list source-voice-id {
                  type string;
                  description
                    "The detected source voice ID for Voice over Internet
                     Protocol (VoIP) and Voice over Cellular Network
                     (VoCN) that violates the security policy.";
                }
                leaf-list destination-voice-id {
                  type string;
                  description
                    "The detected destination voice ID for Voice over
                     Internet Protocol (VoIP) and Voice over Cellular
                     Network (VoCN) that violates the security policy.";
                }
                leaf-list user-agent {
                  type string;
                  description
                    "The detected user-agent for VoIP and VoCN that
                     violates the security policy.";
                }
                description
```

```
                "A violation of VoIP/VoCN is detected";
            }
            description
              "A violation of VoIP/VoCN is detected";
          }
        }
      }
    }

    list i2nsf-feedback-information {
      key "nsf-name time";

      description
        "Feedback information is information about problem(s) of an
         NSF for a security service such as system resource over-usage
         or malfunction. ";

      leaf nsf-name {
        type union {
          type string;
          type inet:ip-address;
        }
        description
          "The name or IP address (IPv4 or IPv6) of the NSF to be
           configured. If the given nsf-name is not IP address, the
           name can be an arbitrary string including FQDN (Fully
           Qualified Domain Name).";
      }

      leaf time {
        type yang:date-and-time;
        description
          "The time of the feedback information delivered";
      }

      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-WY-Za-wy-z]'
                + '(-([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 text in the module
             (i.e., 'leaf message' and 'leaf-list solution').

             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";
        }

        container problem {
          description
            "The issue that is emitted by an NSF via the I2NSF Monitoring
             Interface. The problem for feedback information includes the
             system alarms, such as Memory alarm, CPU alarm, Disk alarm,
             Hardware alarm, and Interface alarm.";
          choice alarm-type {
            description
              "The detected alarm type";
            case memory-alarm {
              container memory-alarm {
                leaf usage {
                  type uint8 {
                    range "0..100";
                  }
                  units "percent";
                  description
                    "The average usage for the duration of the alarm.";
                }
                leaf message {
                  type string;
```

```
            description
              "A message explaining the problem.";
          }
          leaf duration {
            type uint32;
            description
              "Specify the duration of the first alarm triggered
               until the feedback information is created.";
          }
          description
            "The container for memory-alarm";
        }
        description
          "The detected alarm type is memory-alarm";
      }
      case cpu-alarm {
        container cpu-alarm {
          leaf usage {
            type uint8 {
              range "0..100";
            }
            units "percent";
            description
              "The average usage for the duration of the alarm.";
          }
          leaf message {
            type string;
            description
              "A message explaining the problem.";
          }
          leaf duration {
            type uint32;
            description
              "Specify the duration of the first alarm triggered
               until the feedback information is created.";
          }
          description
            "The container for cpu-alarm";
        }
        description
          "The detected alarm type is cpu-alarm";
      }
      case disk-alarm {
        container disk-alarm {
          leaf disk-id {
            type string;
            description
              "The ID of the storage disk. It is a free form
```

```
                identifier to identify the storage disk.";
            }
            leaf usage {
              type uint8 {
                range "0..100";
              }
              units "percent";
              description
                "The average usage for the duration of the alarm.";
            }
            leaf message {
              type string;
              description
                "A message explaining the problem.";
            }
            leaf duration {
              type uint32;
              description
                "Specify the duration of the first alarm triggered
                 until the feedback information is created.";
            }
            description
              "The container for disk-alarm";
          }
          description
            "The detected alarm type is disk-alarm";
        }
        case hardware-alarm {
          container hardware-alarm {
            leaf component-name {
              type string;
              description
                "The hardware component responsible for generating
                 the message. Applicable for Hardware Failure
                 Alarm.";
            }
            leaf message {
              type string;
              description
                "A message explaining the problem.";
            }
            leaf duration {
              type uint32;
              description
                "Specify the duration of the first alarm triggered
                 until the feedback information is created.";
            }
            description
```

```
              "The container for hardware-alarm";
          }
          description
            "The detected alarm type is hardware-alarm";
        }
        case interface-alarm {
          container interface-alarm {
            leaf interface-id {
              type string;
              description
                "The interface ID responsible for generating
                 the message.";
            }
            leaf interface-state {
              type enumeration {
                enum up {
                  value 1;
                  description
                    "The interface state is up and not congested.
                     The interface is ready to pass packets.";
                }
                enum down {
                  value 2;
                  description
                    "The interface state is down, i.e., does not pass
                     any packets.";
                }
                enum congested {
                  value 3;
                  description
                    "The interface state is up but congested.";
                }
                enum testing {
                  value 4;
                  description
                    "In some test mode.  No operational packets can
                     be passed.";
                }
                enum unknown {
                  value 5;
                  description
                    "Status cannot be determined for some reason.";
                }
                enum dormant {
                  value 6;
                  description
                    "Waiting for some external event.";
                }
```

```
                    enum not-present {
                      value 7;
                      description
                        "Some component (typically hardware) is
                         missing.";
                    }
                    enum lower-layer-down {
                      value 8;
                      description
                        "Down due to state of lower-layer interface(s).";
                    }
                  }
                  description
                    "The state of the interface. Applicable for Network
                     Interface Failure Alarm.";
                  reference
                    "RFC 8343: A YANG Data Model for Interface Management
                     - Operational States";
                }
                leaf message {
                  type string;
                  description
                    "A message explaining the problem.";
                }
                leaf duration {
                  type uint32;
                  description
                    "Specify the duration of the first alarm triggered
                     until the feedback information is created.";
                }
                description
                  "The container for interface-alarm";
              }
              description
                "The detected alarm type is interface-alarm";
            }
          }
        }

        leaf-list solution {
          type string;
          description
             "A possible solution given as feedback is in the form of
              a free-form string (as a high-level instruction).";
        }
      }
    }
    <CODE ENDS>
```

Figure 7: YANG for Analytics Interface

5. XML Configuration Examples of Analytics Information

This section shows XML configuration examples of analytics information including feedback policy rules, which is delivered by the I2NSF Analyzer to the Security Controller over the Analytics Interface after the I2NSF Analyzer analyzes monitoring data.

5.1. Policy Reconfiguration for a DDoS Detection

In this example, the scenario can be seen in Figure 8.

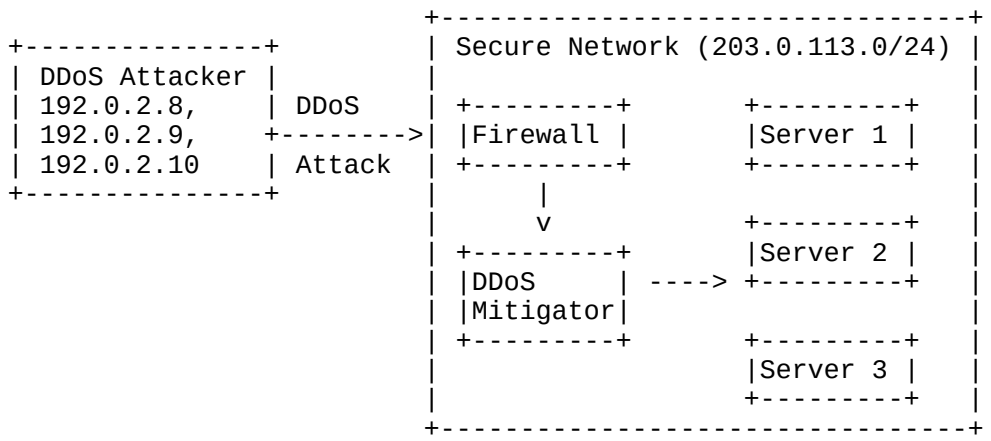

Figure 8: A Scenario Example of a DDoS Attack

In this scenario, a DDoS Mitigator detects a DDoS Attack and sends a notification to the I2NSF Analyzer as shown in Figure 9.

```
<?xml version="1.0" encoding="UTF-8"?>
<notification
 xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2021-08-27T09:00:01.00Z</eventTime>
  <i2nsf-nsf-event
    xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitoring-interface">
    <acquisition-method>subscription</acquisition-method>
    <emission-type>on-change</emission-type>
    <dampening-type>on-repetition</dampening-type>
    <i2nsf-nsf-detection-ddos>
      <attack-type>i2nsfmi:syn-flood</attack-type>
      <start-time>2021-08-27T09:00:00.00Z</start-time>
      <attack-src-ip>192.0.2.8</attack-src-ip>
      <attack-src-ip>192.0.2.9</attack-src-ip>
      <attack-src-ip>192.0.2.10</attack-src-ip>
      <attack-dst-ip>203.0.113.0/24</attack-dst-ip>
      <attack-rate>100</attack-rate>
      <message>A DDoS Attack is detected</message>
      <nsf-name>DDoS_mitigator</nsf-name>
    </i2nsf-nsf-detection-ddos>
  </i2nsf-nsf-event>
</notification>
```

Figure 9: A Detected DDoS Attack by DDoS Mitigator

In the scenario shown in Figure 9, the description of the XML example is as follows:

1. The DDoS attack is detected at 9 am on August 27 in 2021.

2. The sources of the attack are 192.0.2.8, 192.0.2.9, and 192.0.2.10.

3. The destination of the attack is 203.0.113.0/24.

After receiving the monitoring data, the I2NSF Analyzer analyzes it and creates a new feedback policy to enforce the security of the network. The I2NSF Analyzer delivers the feedback policy to the Security Controller as shown in Figure 10.

```
<?xml version="1.0" encoding="UTF-8" ?>
<i2nsf-security-policy
 xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-analytics-interface">
  <system-policy-name>
    feedback_policy_for_ddos_attack
  </system-policy-name>
  <rules>
    <rule-name>deny_ddos_attack</rule-name>
    <condition>
      <ipv4>
        <source-ipv4-range>
          <start>192.0.2.8</start>
          <end>192.0.2.10</end>
        <source-ipv4-range>
      </ipv4>
      <context>
        <time>
          <start-date-time>2021-08-27T09:00:00.00Z</start-date-time>
        </time>
      </context>
    </condition>
    <actions>
      <packet-action>
        <ingress-action>drop</ingress-action>
      </packet-action>
    </actions>
  </rules>
  <nsf-name>Firewall</nsf-name>
  <problem>
    <ddos-detected>
      <attack-src-ip>192.0.2.8</attack-src-ip>
      <attack-src-ip>192.0.2.9</attack-src-ip>
      <attack-src-ip>192.0.2.10</attack-src-ip>
      <attack-dst-ip>203.0.113.0/24</attack-dst-ip>
    </ddos-detected>
  </problem>
</i2nsf-security-policy>
```

Figure 10: Policy Reconfiguration for a Detected DDoS Attack

The policy reconfiguration in Figure 10 means the following:

1. The feedback policy is named as "feedback_policy_for_ddos_attack".

2. The rule is named as "deny_ddos_attack".

3. The rule starts from 09:00 am on August 24 in 2021. The condition of the rule is from the sources of the IP addresses of 192.0.2.8, 192.0.2.9, and 192.0.2.10.

4. The action required is to "drop" any access from the IP addresses that have been identified as malicious.

5. The NSF to be configured is named "Firewall".

6. The problem that triggered the generation of the feedback is a DDoS attack from the sources of the IP addresses of 192.0.2.8, 192.0.2.9, and 192.0.2.10 to the protected network of 203.0.113.0/24.

5.2. Feedback Information for an Overloaded NSF

In this scenario, an NSF is overloaded and sends a notification to the I2NSF Analyzer as shown in Figure 11.

```
<?xml version="1.0" encoding="UTF-8"?>
<notification
 xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2021-08-27T07:43:52.181088+00:00</eventTime>
  <i2nsf-event
    xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitoring-interface">
    <acquisition-method>subscription</acquisition-method>
    <emission-type>on-change</emission-type>
    <dampening-type>on-repetition</dampening-type>
    <language>en-US</language>
    <i2nsf-system-detection-alarm>
      <alarm-category>memory-alarm</alarm-category>
      <usage>91</usage>
      <threshold>90</threshold>
      <message>Memory Usage Exceeded the Threshold</message>
      <nsf-name>time_based_firewall</nsf-name>
      <severity>high</severity>
    </i2nsf-system-detection-alarm>
  </i2nsf-event>
</notification>
```

Figure 11: The Monitoring of an Overloaded NSF

In the scenario shown in Figure 11, the description of the XML example is as follows:

1. The NSF that sends the monitoring data is named "firewall".

2. The memory usage of the NSF triggered the alarm.

3. The memory usage of the NSF is 98 percent.

4. The memory threshold to trigger the alarm is 80 percent.

5. The event is delivered at 2021-08-27T07:43:52.181088+00:00.

After receiving the monitoring data, the I2NSF Analyzer analyzes it and creates a new feedback policy to solve the problem that is detected by the NSF. The I2NSF Analyzer delivers the feedback information to the Security Controller as shown in Figure 12.

```
<?xml version="1.0" encoding="UTF-8" ?>
<i2nsf-feedback-information
 xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-analytics-interface">
  <time>2021-08-27T08:43:52.000000+00:00</time>
  <nsf-name>Firewall</nsf-name>
  <language>en-US</language>
  <problem>
    <memory-alarm>
      <usage>95</usage>
      <message>Memory Usage Exceeded the Threshold</message>
      <duration>3600</duration>
    </memory-alarm>
  </problem>
  <solution>
    Add more memory capacity to the NSF
  </solution>
  <solution>
    Create a new NSF with the same security service
  </solution>
</i2nsf-feedback-information>
```

Figure 12: Feedback Information for an Overloaded NSF

The feedback information in Figure 12 means the following:

1. The name of the NSF that needs to be handled is called "Firewall".

2. The feedback information is delivered at 2021-08-27T08:43:52.000000+00:00.

3. The problem is that the Memory Usage Exceeded the Threshold with the average usage of memory as 95.

4. The problem persists for 3,600 seconds (1 hour) without any fix.

5. The proposed solution to the problem is either to add more memory capacity in hardware to the NSF or to create a new NSF with the same security service.

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-analytics-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-analytics-interface
namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-analytics-interface
prefix: i2nsfai
reference: RFC XXXX

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

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 contents.

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. The data model in this document uses

the data model from NSF-Facing Interface data model, it MUST follow the Security Considerations mentioned in [I-D.ietf-i2nsf-nsf-facing-interface-dm].

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. Thus, it is important to control read access (e.g., via get, get-config, or notification) to these data nodes. This document MUST also follow the Security Considerations about the readable data nodes mentioned in [I-D.ietf-i2nsf-nsf-facing-interface-dm].

8. References

8.1. Normative References

[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>.

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

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

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

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

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <https://www.rfc-editor.org/info/rfc6265>.

[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>.

[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>.

[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>.

[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>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

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

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

[RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <https://www.rfc-editor.org/info/rfc8525>.

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

[RFC9315] Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", RFC 9315, DOI 10.17487/RFC9315, October 2022, <https://www.rfc-editor.org/info/rfc9315>.

[I-D.ietf-i2nsf-nsf-facing-interface-dm] Kim, J. T., Jeong, J. P., Jung-Soo, J., Hares, S., and Q. Lin, "I2NSF Network Security Function-Facing Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-nsf-facing-interface-dm-29, 1 June 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-nsf-facing-interface-dm-29>.

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

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

8.2. Informative References

[I-D.jeong-i2nsf-security-management-automation] Jeong, J. P., Lingga, P., Jung-Soo, J., Lopez, D., and S. Hares, "An I2NSF Framework for Security Management Automation in Cloud-Based Security Systems", Work in Progress, Internet-Draft, draft-jeong-i2nsf-security-management-automation-08, 26 July 2024, <https://datatracker.ietf.org/doc/html/draft-jeong-i2nsf-security-management-automation-08>.

[I-D.ietf-i2nsf-applicability] Jeong, J. P., Hyun, S., Ahn, T., Hares, S., and D. Lopez, "Applicability of Interfaces to Network Security Functions to Network-Based Security Services", Work in Progress, Internet-Draft, draft-ietf-i2nsf-applicability-18, 16 September 2019, <https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-applicability-18>.

[ICSC] Lingga, P., Jeong, J., and L. Dunbar, "ICSC: Intent-Based Closed-Loop Security Control System for Cloud-Based Security Services", IEEE Communications Magazine, DOI https://doi.org/10.1109/MCOM.001.2400022, December 2024, <https://doi.org/https://doi.org/10.1109/MCOM.001.2400022>.

[SPT] Lingga, P., Jeong, J., Yang, J., and J. Kim, "SPT: Security Policy Translator for Network Security Functions in Cloud-Based Security Services", IEEE Transactions on Dependable and Secure Computing, Volume 21, Issue 6, DOI https://doi.org/10.1109/TDSC.2024.3371788, November 2024, <https://doi.org/https://doi.org/10.1109/TDSC.2024.3371788>.

[Deep-Learning] Goodfellow, I., Bengio, Y., and A. Courville, "Deep Learning", Publisher: The MIT Press, URL: https://www.deeplearningbook.org/, November 2016.

Acknowledgments

This document benefited from discussions in the I2NSF Working Group, especially from Linda Dunbar, Yoav Nir, and Diego Lopez. This document took advantage of the review and comments from the following experts: Roman Danyliw 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 Ministry of Science and ICT (MSIT)(RS-2024-00398199).

This work was supported in part by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Ministry of Science and ICT (MSIT)(No. RS-2022-II221015, Development of Candidate Element Technology for Intelligent 6G Mobile Core Network).

Contributors

The following are coauthors of this document:

Jung-Soo Park
Standards & Open Source Research Division
Electronics and Telecommunications Research Institute
218 Gajeong-Ro, Yuseong-Gu,
Email: pjs@etri.re.kr

Younghan Kim
School of Electronic Engineering
Soongsil University
369, Sangdo-ro, Dongjak-gu
Email: younghak@ssu.ac.kr

Authors' Addresses

Patrick Lingga (editor)
Department of Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4957
Email: patricklink@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

Yunchul Choi
Standards & Open Source Research Division
Electronics and Telecommunications Research Institute
218 Gajeong-Ro, Yuseong-Gu
Daejeon
34129
Republic of Korea
Phone: +82 42 860 5978
Email: cyc79@etri.re.kr